

University of Tasmania Open Access Repository Cover sheet

Title

Distributed computing and communication in peer-to-peer networks

Author

Goldsmith, B

Bibliographic citation

Goldsmith, B (2010). Distributed computing and communication in peer-to-peer networks. University Of Tasmania. Thesis. https://doi.org/10.25959/23210420.v1

Is published in:

Copyright information

This version of work is made accessible in the repository with the permission of the copyright holder/s under the following,

Licence.

Rights statement: Copyright 2010 the author

If you believe that this work infringes copyright, please email details to: oa.repository@utas.edu.au

Downloaded from University of Tasmania Open Access Repository

Please do not remove this coversheet as it contains citation and copyright information.

University of Tasmania Open Access Repository

Library and Cultural Collections University of Tasmania Private Bag 3 Hobart, TAS 7005 Australia



abstractpeer.h

```
00001 /*
00002
          abstractpeer.h
00003
       * /
00004
00005
00006 #ifndef _simplep2p_h_
00007 #define _simplep2p_h_
00008
00009 #include "controller.h"
00010 #include "comptorrentpeer.h"
00011
00012 #include <iostream>
00013 #include <cc++/socket.h>
00014
00015
00016 #define SERVER
                                         0
00017 #define CONNECT
                                          2
00018 #define PEER
00019 #define PEER LISTENER
                                         4
00020 #define CLIENT
00021
00022 #define TRANSMIT
                                         0
00023 #define RECEIVE
00024
00025 using namespace std;
00026 using namespace ost;
00027
00028 enum duplex { d_listen, d_talk};
00029
00030 class AbstractPeer : public TCPSession
00031 {
00032 public:
00033
00034
               AbstractPeer(TCPSocket &server, Controller* p, short node_type_);
00035
               AbstractPeer(InetHostAddress i, tpport_t port, Controller* p, short
node_type_);
               ~AbstractPeer();
00036
00037
00038
               void kill_yourself() { die = true; };
00039
00040
               void set_my_ip(string ip, string port) { my_ip = ip; my_port = port; }
               void throttle_wait(int time_) { current_wait = time_; }
00041
00042
00043
               void push_msg(string msg) { out_box.push_back(msg); }
               string get_my_ip() {return my_ip; }
string get_my_port() { return my_port;
00044
00045
               Controller* get_parent() {return parent; }
00046
00047
00048
               MessageCollection out_box;
00049
               MessageCollection in_box;
00050
00051 protected:
00052
00053
               int current_wait;
00054
               void run();
00055
               void log(const string s) { cout << s << endl; }</pre>
00056
00057
               bool die;
               short sync_mode;
00058
00059
00060
               CompTorrentPeer* dt;
00061
               Controller* parent;
00062
               short node_type;
               int virtcount;
00063
00064
               unsigned int max_chunk_size;
00065
00066
               string my_ip, my_port;
00067
               unsigned short timeout;
00068
00069
               unsigned long num messages;
00070
00071
               int threadid;
```

```
00072 };
00073
00074 #endif
00075
```

Generated on Sat Mar 27 12:27:34 2010 for CompTorrent by



AbstractPeer

CompTorrentParser CompTorrentPeer Controller file_chunk file_chunk_request known_peer

Listener

MessageCollection

NetworkSocket

orig_data_chunk

peer Processor

Router

TrackerParser

WorkerThread

xml_to_map Controller



Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

nere are the classes, structs,
AbstractPeer
CompTorrentParser
CompTorrentPeer
Controller
file_chunk
file_chunk_request
known_peer
Listener
MessageCollection
NetworkSocket
orig_data_chunk
peer
Processor
Router
TrackerParser
WorkerThread



.

BLOBs for PHPLIB

Sascha Schumann, sascha@schumann.cx

\$Date: 2000/04/17 16:40:06 \$, \$Revision: 1.1.1.1 \$

RFD on a BLOB API for PHPLIB

1. BLOB (Binary Large OBjects)

- 1.1 Defining the problem
- 1.2 Proposed extension API
- 1.3 Design notes

Next Previous Contents

BLOBs for PHPLIB

Sascha Schumann, sascha@schumann.cx

\$Date: 2000/04/17 16:40:06 \$, \$Revision: 1.1.1.1 \$

RFD on a BLOB API for PHPLIB

11.. BBLLOOBB ((BBiinnaarryy LLaarrggee OOBBjjeeccttss))

This chapter is intended to give you a quick introduction to BLOBs, their interface as proposed for PHPLIB.

11..11.. DDeeffiinniinngg tthhee pprroobblleemm

Achieving the goal of creating portable applications is hindered by the fact that no portable way of storing large amount of data exists.

Every database vendor defines his own application programming interface (API) to access binary large objects, or uses other, non-standard ways to enable the use of BLOBs.

Enhancing the existing vendor neutral support for databases in PHPLIB to also include BLOB support could be a short-term solution to the problem.

11..22.. PPrrooppoosseedd eexxtteennssiioonn AAPPII

An overview over the function calls follows.

bblloobb ccrreeaattee(())

This will create a new BLOB and return its ID. If it is impossible to create a new BLOB, this function will halt() the execution and generate an error message.

bblloobb__ooppeenn((\$\$IIDD))

Opens the specified BLOB for reading and writing and returns true or false on success or failure, respectively.

bblloobb__cclloossee((\$\$IIDD))

Closes the referenced BLOB. Do not forget this - it might be necessary for some databases or you might face data loss.

bblloobb ddeelleettee((\$\$IIDD))

Deletes the specified BLOB. All associated resources are freed. The BLOB must not be referenced at a later time.

bblloobb__rreeaadd((\$\$IIDD))

Reads the entire data of BLOB and returns it.

bblloobb_wwrriittee((\$\$IIDD,, \$\$DDAATTAA))

Overwrites the whole BLOB with \$DATA.

11..33.. DDeessiiggnn nnootteess

The goal was to create a vendor neutral API which could be implemented easily with most databases. Eventually, it should provide a "fallback" mode enabling the user to utilize BLOBs even if the database does not support BLOBs natively. Keep in mind that it should be able to cover as many databases as possible and therefore only implements a subset of functionality provided by modern databases.



1. BLOB (Binary Large OBjects)

This chapter is intended to give you a quick introduction to BLOBs, their interface as proposed for PHPLIB.

1.1 Defining the problem

Achieving the goal of creating portable applications is hindered by the fact that no portable way of storing large amount of data exists.

Every database vendor defines his own application programming interface (API) to access binary large objects, or uses other, non-standard ways to enable the use of BLOBs.

Enhancing the existing vendor neutral support for databases in PHPLIB to also include BLOB support could be a short-term solution to the problem.

1.2 Proposed extension API

An overview over the function calls follows.

blob create()

This will create a new BLOB and return its ID. If it is impossible to create a new BLOB, this function will halt() the execution and generate an error message.

blob_open(\$ID)

Opens the specified BLOB for reading and writing and returns true or false on success or failure, respectively.

blob close(\$ID)

Closes the referenced BLOB. Do not forget this - it might be necessary for some databases or you might face data loss.

blob_delete(\$ID)

Deletes the specified BLOB. All associated resources are freed. The BLOB must not be referenced at a later time.

blob read(\$ID)

Reads the entire data of BLOB and returns it.

blob_write(\$ID, \$DATA)

Overwrites the whole BLOB with \$DATA.

1.3 Design notes

The goal was to create a vendor neutral API which could be implemented easily with most databases. Eventually, it should provide a "fallback" mode enabling the user to utilize BLOBs even if the database does not support BLOBs

natively. Keep in mind that it should be able to cover as many databases as possible and therefore only implements a subset of functionality provided by modern databases.

Next Previous Contents

.



Copyright (c) 2005, The EasyBMP Project (http://easybmp.sourceforge.net) All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- 3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Changes in version 1.0.1:

- Fixed comment tags which were outputing as '<?--' instead of the correct '<!--'.
- Implemented the Next and Prev methods of the TiXmlAttribute class.
- Renamed 'LastAttribtute' to 'LastAttribute'
- Fixed bad pointer to 'isspace' that could occur while parsing text.
- Errors finding beginning and end of tags no longer throw it into an infinite loop. (Hopefully.)

Changes in version 1.0.2

- Minor documentation fixes.

Changes in version 1.0.3

- After nodes are added to a document, they return a pointer to the new node instead of a bool for success.
- Elements can be constructed with a value, which is the element name. Every element must have a value or it will be invalid, but the code changes to enforce this are not fully in place.

Changes in version 1.1.0

- Added the TiXmlAttributeSet class to pull the attributes into a seperate container.
- Moved the doubly liked list out of XmlBase. Now XmlBase only requires the Print() function and defines some utility functions.
- Moved errors into a seperate file. (With the idea of internationalization to the other latin-1 languages.)
- Added the "NodeType"
- Fixed white space parsing in text to conform with the standard. Basically, all white space becomes just one space.
- Added the TiXmlDeclaration class to read xml declarations.

Changes in version 1.2.0

- Removed the factory. The factory was not really in the spirit of small and simple, confused the code, and was of limited value.
- Added FirstChildElement and NextSiblingElement, because they are such common functions.
- Re-wrote the example to test and demonstrate more functionality.

Changes in version 1.2.1

- Fixed a bug where comments couldn't be inside elements.
- Loading now clears out existing XML rather than appending.
- Added the "Clear" method on a node to delete all its children.

Changes in version 1.2.2

- Fixed TiXmlAttribute::Previous actually returning "next." Thanks to Rickard Troedsson for the bug fix.

Changes in version 1.2.3

- Added the TIXML prefix to the error strings to resolve conflicts with #defines in OS headers. Thanks to Steve Lhomme.
- Fixed a delete buf that should be a delete [] buf. Thanks to Ephi Sinowitz.

Changes in version 1.2.4

- ReplaceChild() was almost guarenteed to fail. Should be fixed, thanks to Joe Smith. Joe also pointed out that the Print() functions should take stream references: I agree, and would like to overload the Print() method to take either format, but I don't want to do this in a dot release.
- Some compilers seem to need an extra <ctype.h> include. Thanks to Steve Lhomme for that.

Changes in version 2.0.0 BETA

- Made the ToXXX() casts safe if 'this' is null.
- When "LoadFile" is called with a filename, the value will correctly get set.

Thanks to Brian Yoder.

- Fixed bug where isalpha() and isalnum() would get called with a negative value for high ascii numbers. Thanks to Alesky Aksenov.
- Fixed some errors codes that were not getting set.
- Made methods "const" that were not.
- Added a switch to enable or disable the ignoring of white space. (TiXmlDocument::SetIgnoreWhiteSpace())
- Greater standardization and code re-use in the parser.
- Added a stream out operator.
- Added a stream in operator.
- Entity support, of predefined entites. &#x entities are untouched by input or output.
- Improved text out formatting.
- Fixed ReplaceChild bug, thanks to Tao Chen.

Changes in version 2.0.1

- Fixed hanging on loading a 0 length file. Thanks to Jeff Scozzafava.
- Fixed crashing on InsertBeforeChild and InsertAfterChild. Also possibility of bad links being created by same function. Thanks to Frank De prins.
- Added missing licence text. Thanks to Lars Willemsens.
- Added <ctype.h> include, at the suggestion of Steve Walters.

Changes in version 2.1.0

- Yves Berquin brings us the STL switch. The forum on SourceForge, and various emails to me, have long debated all out STL vs. no STL at all. And now you can have it both ways. TinyXml will compile either way.

Changes in version 2.1.1

- Compilation warnings.

Changes in version 2.1.2

- Uneeded code is not compiled in the STL case.
- Changed headers so that STL can be turned on or off in tinyxml.h

Changes in version 2.1.3

- Fixed non-const reference in API; now uses a pointer.
- Copy constructor of TiXmlString not checking for assignment to self.
- Nimrod Cohen found a truly evil bug in the STL implementation that occurs when a string is converted to a c_str and then assigned to self. Search for STL_STRING_BUG for a full description. I'm asserting this is a Microsoft STL bug, since &string and string.c_str() should never be the same. Nevertheless, the code works around it.
- Urivan Saaib pointed out a compiler conflict, where the C headers define

the isblank macro, which was wiping out the TiXmlString::isblank() method. The method was unused and has been removed.

Changes in version 2.1.4

- Reworked the entity code. Entities were not correctly surving round trip input and output. Will now automatically create entities for high ascii in output.

Changes in version 2.1.5

- Bug fix by kylotan: infinite loop on some input (tinyxmlparser.cpp rev 1.27)
- Contributed by Ivica Aracic (bytelord): 1 new VC++ project to compile versions as static libraries (tinyxml_lib.dsp), and an example usage in xmltest.dsp (Patch request ID 678605)
- A suggestion by Ronald Fenner Jr (dormlock) to add #include <istream> and <ostream> for Apple's Project Builder (Patch request ID 697642)
- A patch from ohommes that allows to parse correctly dots in element names and attribute names (Patch request 602600 and kylotan 701728)
- A patch from hermitgeek (James) and wasteland for improper error reporting
- Reviewed by Lee, with the following changes:
 - Got sick of fighting the STL/non-STL thing in the windows build. Broke them out as seperate projects.
 - I have too long not included the dsw. Added.
 - TinyXmlText had a protected Print. Odd.
 - Made LinkEndChild public, with docs and appropriate warnings.
 - Updated the docs.

2.2.0

- Fixed an uninitialized pointer in the TiXmlAttributes
- Fixed STL compilation problem in MinGW (and gcc 3?) thanks Brian Yoder for finding this one
- Fixed a syntax error in TiXmlDeclaration thanks Brian Yoder
- Fletcher Dunn proposed and submitted new error handling that tracked the row and column. Lee modified it to not have performance impact.
- General cleanup suggestions from Fletcher Dunn.
- In error handling, general errors will no longer clear the error state of specific ones.
- Fix error in documentation : comments starting with "<?--" instead of "<!--" (thanks ion_pulse)
- Added the TiXmlHandle. An easy, safe way to browse XML DOMs with less code.
- Added QueryAttribute calls which have better error messaging. (Proposed by Fletcher Dunn)
- Nodes and attributes can now print themselves to strings. (Yves suggestion)
- Fixed bug where entities with one character would confuse parser. (Thanks Roman)

2.2.1

- Additional testing (no more bugs found to be fixed in this release)
- Significant performance improvement to the cursor code.

2.3.0

- User Data are now defined in TiXmlBase instead of TiXmlNode
- Character Entities are now UCS-2
- Character Entities can be decimal or hexadecimal
- UTF-8 conversion.
- Fixed many, many bugs.

2.3.1

- Fixed bug in handling nulls embedded in the input.
- Make UTF-8 parser tolerant of bad text encoding.
- Added encoding detection.

- Many fixes and input from John-Philip Leonard Johansson (JP) and Ellers, including UTF-8 feedback, bug reports, and patches. Thanks!
- Added version # constants a suggestion from JP and Ellers.
- [979180] Missing; in entity reference, fix from Rob Laveaux.
- Copy constructors and assignment have been a long time coming. Thanks to Fokke and JP.

2.3.2

- Made the IsAlpha and IsAlphaNum much more tolerant of non-UTF-8 encodings. Thanks Volker Boerchers for finding the issue.
- Ran the program though the magnificent Valgrind http://valgrind.kde.org to check for memory errors. Fixed some minor issues.

2.3.3

- Fixed crash when test program was run from incorrect directory.
- Fixed bug 1070717 empty document not returned correctly thanks Katsuhisa Yuasa.
- Bug 1079301 resolved deprecated stdlib calls. Thanks Adrian Boeing.
- Bug 1035218 fixed documentation errors. Xunji Luo
- Other bug fixes have accumulated and been fixed on the way as well; my apologies to authors not credited!
- Big fix / addition is to correctly return const values. TinyXml could basically remove const in a method like this: TiXmlElement* Foo() const, where the returned element was a pointer to internal data. That is now: const TiXmlElement* Foo() const and TiXmlElement* Foo().

2.3.4

- Fixed additional const errors, thanks Kent Gibson.
- Correctly re-enable warnings after tinyxml header. Thanks Cory Nelson.
- Variety of type cleanup and warning fixes. Thanks Warren Stevens.
- Cleaned up unneeded constructor calls in TinyString thanks to Geoff Carlton and the discussion group on sourceforge.

2.4.0

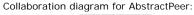
- Improved string class, thanks Tyge Lovset (whose name gets mangled in English sorry)
- Type cast compiler warning, thanks Rob van den Bogaard
- Added GetText() convenience function. Thanks Ilya Parniuk & Andrew Ellers for input.
- Many thanks to marlonism for finding an infinite loop in bad xml.
- A patch to cleanup warnings from Robert Gebis.
- Added ValueStr() to get the value of a node as a string.
- TiXmlText can now parse and output as CDATA
- Additional string improvement from James (z2895)
- Removed extraneous 'const', thanks David Aldrich
- First pass at switching to the "safe" stdlib functions. Many people have suggested and pushed on this, but Warren Stevens put together the first proposal.
- TinyXml now will do EOL normalization before parsing, consistent with the W3C XML spec.
- Documents loaded with the UTF-8 BOM will now save with the UTF-8 BOM. Good suggestion from 'instructor '
- Ellers submitted his very popular tutorials, which have been added to the distribution.

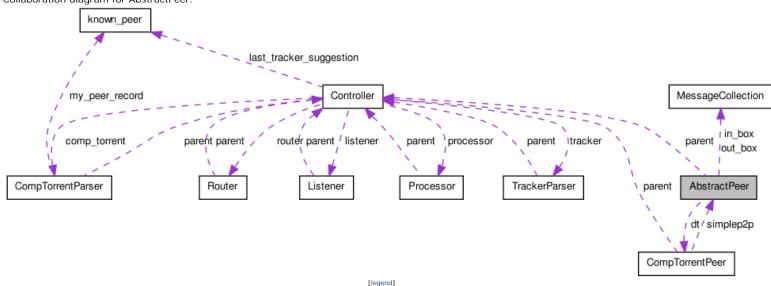
2.4.1

- Fixed CDATA output formatting
- Fixed memory allocators in TinyString to work with overloaded new/delete



AbstractPeer Class Reference





List of all members.

Public Member Functions

	AbstractPeer (TCPSocket &server, Controller *p, short node_type_)	
	AbstractPeer (InetHostAddress i, tpport_t port, Controller *p, short node_type_)	
void	kill_yourself ()	
void	set_my_ip (string ip, string port)	
void	throttle_wait (int time_)	
void	push_msg (string msg)	
string	get_my_ip ()	
string	get_my_port ()	
Controller *	get_parent ()	

Public Attributes

Protected Member Functions

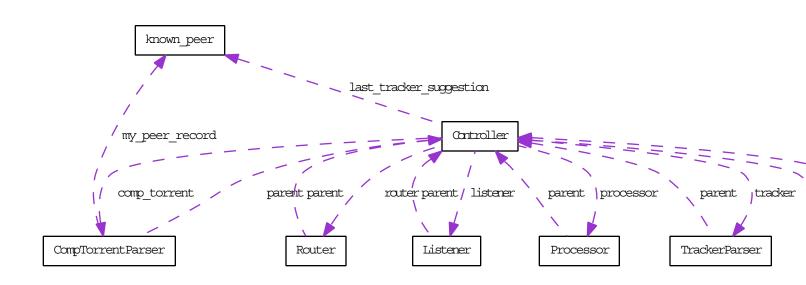
void run () void log (const string s)

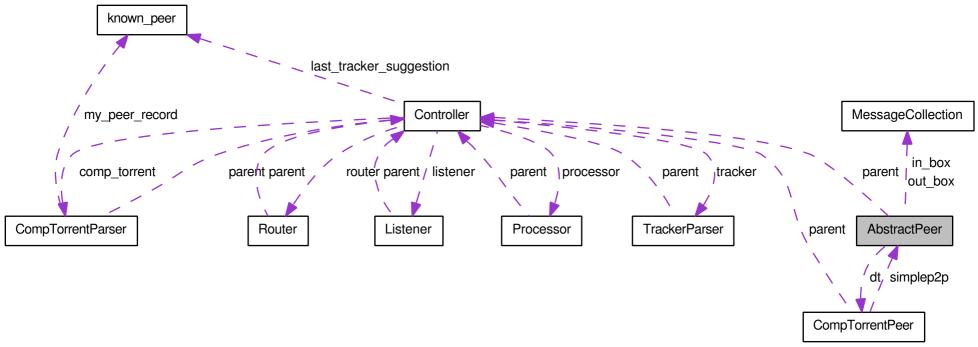
Protected Attributes

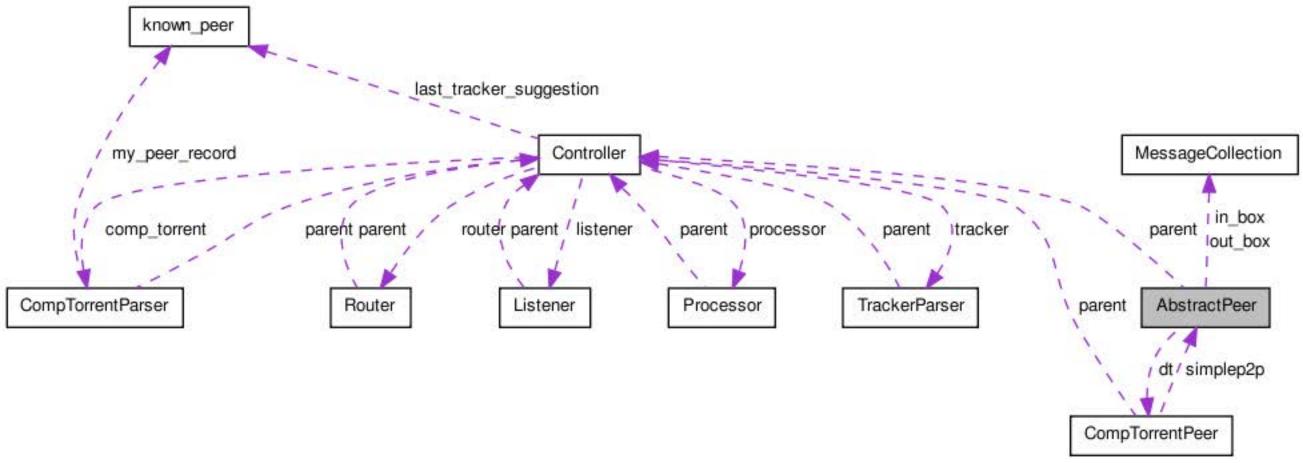
int	current_wait
bool	die
short	sync_mode
CompTorrentPeer *	dt
Controller *	parent
short	node_type
int	virtcount
unsigned int	max_chunk_size
string	my_ip
string	my_port
unsigned short	timeout
unsigned long	num_messages
int	threadid

The documentation for this class was generated from the following files:

- · abstractpeer.h
- abstractpeer.cpp









AbstractPeer Member List

This is the complete list of members for **AbstractPeer**, including all inherited members.

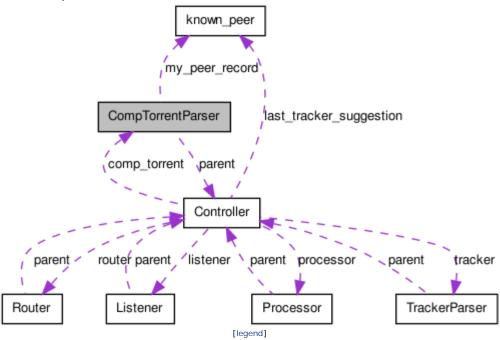
······································	
AbstractPeer (TCPSocket &server, Controller *p, short node_type_) (defined in AbstractPeer)	AbstractPeer
AbstractPeer (InetHostAddress i, tpport_t port, Controller *p, short node_type_) (defined in AbstractPeer)	AbstractPeer
current_wait (defined in AbstractPeer)	AbstractPeer [protected]
die (defined in AbstractPeer)	AbstractPeer [protected]
dt (defined in AbstractPeer)	AbstractPeer [protected]
get_my_ip() (defined in AbstractPeer)	AbstractPeer [inline]
get_my_port() (defined in AbstractPeer)	AbstractPeer [inline]
get_parent() (defined in AbstractPeer)	AbstractPeer [inline]
in_box (defined in AbstractPeer)	AbstractPeer
kill_yourself() (defined in AbstractPeer)	AbstractPeer [inline]
log(const string s) (defined in AbstractPeer)	AbstractPeer [inline, protected]
max_chunk_size (defined in AbstractPeer)	AbstractPeer [protected]
my_ip (defined in AbstractPeer)	AbstractPeer [protected]
my_port (defined in AbstractPeer)	AbstractPeer [protected]
node_type (defined in AbstractPeer)	AbstractPeer [protected]
num_messages (defined in AbstractPeer)	AbstractPeer [protected]
out_box (defined in AbstractPeer)	AbstractPeer
parent (defined in AbstractPeer)	AbstractPeer [protected]
<pre>push_msg(string msg) (defined in AbstractPeer)</pre>	AbstractPeer [inline]
run() (defined in AbstractPeer)	AbstractPeer [protected]
set_my_ip(string ip, string port) (defined in AbstractPeer)	AbstractPeer [inline]
sync_mode (defined in AbstractPeer)	AbstractPeer [protected]
threadid (defined in AbstractPeer)	AbstractPeer [protected]
throttle_wait (int time_) (defined in AbstractPeer)	AbstractPeer [inline]
timeout (defined in AbstractPeer)	AbstractPeer [protected]
virtcount (defined in AbstractPeer)	AbstractPeer [protected]
~AbstractPeer() (defined in AbstractPeer)	AbstractPeer





CompTorrentParser Class Reference

Collaboration diagram for CompTorrentParser:



List of all members.

Public Member Functions

	CompTorrentParser (Controller *parent_)
bool	load_xml (string)
bool	create_working ()
string	get_version ()
string	get_tracker_url ()
string	get_tracker_port ()
string	get_name ()
string	get_size ()
string	get_md5 ()
string	get_algorithm_base64 ()
string	get_algorithm_filename ()
string	get_execution ()
unsigned long	get_num_orig_file_chunks ()
bool	comp_data_exists (string filename)
bool	orig_data_exists (string filename)
bool	known_peer_exists (known_peer kp)
void	add_known_peer (known_peer kp)
void	remove_known_peer (known_peer kp)
unsigned long	get_num_known_peers ()
unsigned long	get_num_connected_peers ()
string	chunk_name (unsigned long chunk_num)
string	chunk_orig_hash (string chunk_name)
bool	get_orig_hash (string chunk_name, string &resulthash)
void	<pre>update_chunk (unsigned long chunk_num, computation_state state, bool orig_file_exists, bool comp_file_exists)</pre>
void	<pre>update_chunk (string chunk_name, computation_state state, bool orig_file_exists, bool comp_file_exists)</pre>

void	<pre>update_comp_chunk (string chunk_name, computation_state state, bool file_exists)</pre>
void	<pre>update_orig_chunk (string chunk_name, computation_state state, bool file_exists)</pre>
unsigned long	get_num_chunks ()
bool	has_comp_converged (computation_state state)
string	get_node_uuid ()
unsigned long	get_num_computed_chunks ()
unsigned long	get_num_original_chunks ()
void	increment_num_computed_chunks ()
void	<pre>add_directory_to_file (string filename, string comp_or_data, string &filepath)</pre>
void	<pre>strip_directory_from_file (string filepath, string &chunk_name)</pre>
void	calculate_file_hash (string filepath, string &resulthash)

Static Public Member Functions

static int **my_file_size** (const string filepath)

Public Attributes

```
vector< orig_data_chunk > data_chunk_list
filemap known_files
```

Protected Member Functions

string **get_UUID** (string device)

Protected Attributes

bool	loaded_ok
string	version
string	tracker_url
string	tracker_port
string	name
string	size
string	md5
string	algorithm_base64
string	algorithm_filename
string	execution
string	uuid
vector< known_peer >	known_peers
known_peer	my_peer_record
Mutex	status_mutex
unsigned long	num_computed_chunks
unsigned long	num_original_chunks
Controller *	parent

Friends

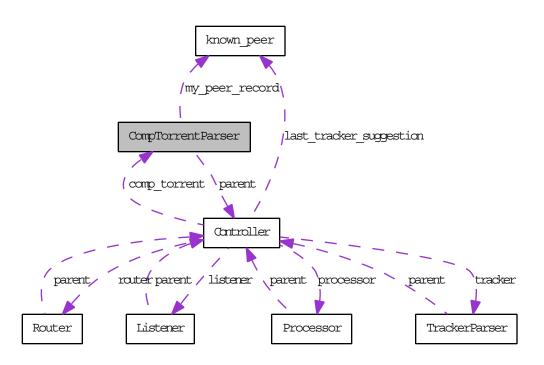
class	Processor
class	CompTorrent
class	DistTorrent

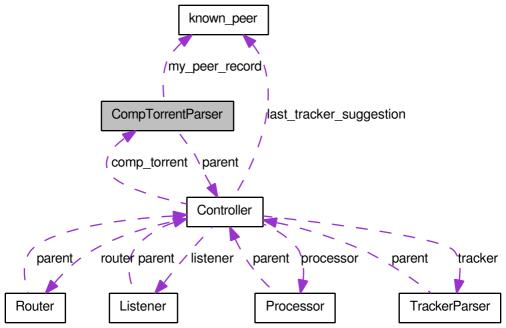
The documentation for this class was generated from the following files:

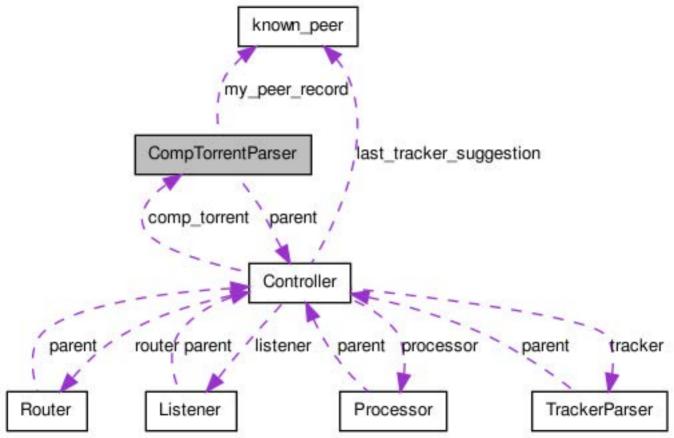
- comptorrentparser.h
- comptorrentparser.cpp

1.6.1

Generated on Sat Mar 27 12:27:42 2010 for CompTorrent by







Main Page Classes Files

Class List Class Members

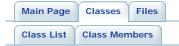
CompTorrentParser Member List

This is the complete list of members for **CompTorrentParser**, including all inherited members.

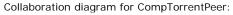
This is the complete list of members for comproment arser , including all inherited	TICTIDCI 3.	
<pre>add_directory_to_file(string filename, string comp_or_data, string &filepath) (defined in CompTorrentParser)</pre>	CompTorrentParser	
add_known_peer(known_peer kp) (defined in CompTorrentParser)	CompTorrentParser	
algorithm_base64 (defined in CompTorrentParser)	CompTorrentParser	[protected]
algorithm_filename (defined in CompTorrentParser)	CompTorrentParser	[protected]
<pre>calculate_file_hash(string filepath, string &resulthash) (defined in CompTorrentParser)</pre>	CompTorrentParser	
<pre>chunk_name(unsigned long chunk_num) (defined in CompTorrentParser)</pre>	CompTorrentParser	
<pre>chunk_orig_hash(string chunk_name) (defined in CompTorrentParser)</pre>	CompTorrentParser	
<pre>comp_data_exists(string filename) (defined in CompTorrentParser)</pre>	CompTorrentParser	
CompTorrent (defined in CompTorrentParser)	CompTorrentParser	[friend]
CompTorrentParser(Controller *parent_) (defined in CompTorrentParser)	CompTorrentParser	
<pre>create_working() (defined in CompTorrentParser)</pre>	CompTorrentParser	
data_chunk_list (defined in CompTorrentParser)	CompTorrentParser	
DistTorrent (defined in CompTorrentParser)	CompTorrentParser	[friend]
execution (defined in CompTorrentParser)	CompTorrentParser	[protected]
get_algorithm_base64() (defined in CompTorrentParser)	CompTorrentParser	
get_algorithm_filename() (defined in CompTorrentParser)	CompTorrentParser	
get_execution() (defined in CompTorrentParser)	CompTorrentParser	
get_md5() (defined in CompTorrentParser)	CompTorrentParser	
get_name() (defined in CompTorrentParser)	CompTorrentParser	
get_node_uuid() (defined in CompTorrentParser)	CompTorrentParser	
get_num_chunks() (defined in CompTorrentParser)	CompTorrentParser	
get_num_computed_chunks() (defined in CompTorrentParser)	CompTorrentParser	
get_num_connected_peers() (defined in CompTorrentParser)	CompTorrentParser	
get_num_known_peers() (defined in CompTorrentParser)	CompTorrentParser	
get_num_orig_file_chunks() (defined in CompTorrentParser)	CompTorrentParser	
get_num_original_chunks() (defined in CompTorrentParser)	CompTorrentParser	
<pre>get_orig_hash(string chunk_name, string &resulthash) (defined in CompTorrentParser)</pre>	CompTorrentParser	
get_size() (defined in CompTorrentParser)	CompTorrentParser	
get_tracker_port() (defined in CompTorrentParser)	CompTorrentParser	
get_tracker_url() (defined in CompTorrentParser)	CompTorrentParser	
get_UUID(string device) (defined in CompTorrentParser)	CompTorrentParser	[protected]
get_version() (defined in CompTorrentParser)	CompTorrentParser	
has_comp_converged(computation_state state) (defined in CompTorrentParser)	CompTorrentParser	
increment_num_computed_chunks() (defined in CompTorrentParser)	CompTorrentParser	
known_files (defined in CompTorrentParser)	CompTorrentParser	
known_peer_exists(known_peer kp) (defined in CompTorrentParser)	CompTorrentParser	
known_peers (defined in CompTorrentParser)	CompTorrentParser	[protected]
load_xml(string) (defined in CompTorrentParser)	CompTorrentParser	
loaded_ok (defined in CompTorrentParser)	CompTorrentParser	
md5 (defined in CompTorrentParser)	CompTorrentParser	
my_file_size(const string filepath) (defined in CompTorrentParser)	CompTorrentParser	
my_peer_record (defined in CompTorrentParser)	CompTorrentParser	
name (defined in CompTorrentParser)	CompTorrentParser	
num_computed_chunks (defined in CompTorrentParser)	CompTorrentParser	
num_original_chunks (defined in CompTorrentParser)	CompTorrentParser	[protected]

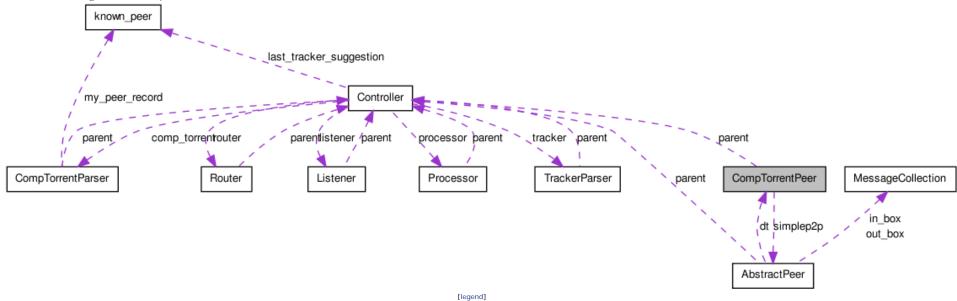
<pre>orig_data_exists(string filename) (defined in CompTorrentParser)</pre>	CompTorrentParser
parent (defined in CompTorrentParser)	CompTorrentParser [protected]
Processor (defined in CompTorrentParser)	CompTorrentParser [friend]
remove_known_peer(known_peer kp) (defined in CompTorrentParser)	CompTorrentParser
size (defined in CompTorrentParser)	CompTorrentParser [protected]
status_mutex (defined in CompTorrentParser)	CompTorrentParser [protected]
<pre>strip_directory_from_file(string filepath, string &chunk_name) (defined in CompTorrentParser)</pre>	CompTorrentParser
tracker_port (defined in CompTorrentParser)	CompTorrentParser [protected]
tracker_url (defined in CompTorrentParser)	CompTorrentParser [protected]
<pre>update_chunk(unsigned long chunk_num, computation_state state, bool orig_file_exists, bool comp_file_exists) (defined in CompTorrentParser)</pre>	CompTorrentParser
<pre>update_chunk(string chunk_name, computation_state state, bool orig_file_exists, bool comp_file_exists) (defined in CompTorrentParser)</pre>	CompTorrentParser
<pre>update_comp_chunk(string chunk_name, computation_state state, bool file_exists) (defined in CompTorrentParser)</pre>	CompTorrentParser
<pre>update_orig_chunk(string chunk_name, computation_state state, bool file_exists) (defined in CompTorrentParser)</pre>	CompTorrentParser
uuid (defined in CompTorrentParser)	CompTorrentParser [protected]
version (defined in CompTorrentParser)	CompTorrentParser [protected]





CompTorrentPeer Class Reference





List of all members.

Public Member Functions

	CompTorrentPeer (Controller *s, AbstractPeer *simplep2p_)
void	<pre>push_msg (string msg)</pre>
bool	has_file_available (string filename, string type)
void	debug_stats ()
string	get_buddy_route ()
bool	is_higher_route ()

Protected Member Functions

bool	add_available_file (string, string)
void	send_welcome_message ()
void	send_noop ()
void	run ()
bool	<pre>process_noop (string intmp)</pre>
bool	<pre>process_file_update (string intmp)</pre>
bool	<pre>process_welcome_message (string intmp)</pre>
bool	<pre>process_connect_message (string intmp)</pre>
bool	<pre>process_file_reply (string intmp)</pre>
bool	<pre>process_file_request (string intmp)</pre>
bool	<pre>process_overlay_message (string intmp)</pre>
bool	<pre>process_overlay_update_message (string intmp)</pre>
bool	ask_for_next_comp_file ()
bool	ask_for_next_orig_file ()

CompTorrent: CompTorrentPeer Class Reference

void	save_file_reply (xmlmap &xml_map)
void	parse_datachunks (string xml)

Static Protected Member Functions

static bool	string_find (const string needle, const string haystack)
static string	trim_spaces (std::string const &str)

Protected Attributes

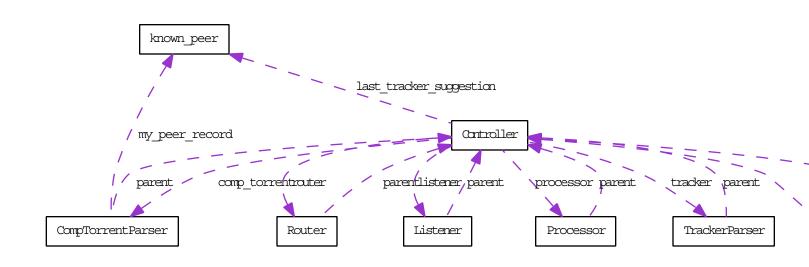
AbstractPeer *	simplep2p
Controller *	parent
int	num_messages
short	peer_type
distributed_state	state
string	buddy_route
file_chunk_type	files_available

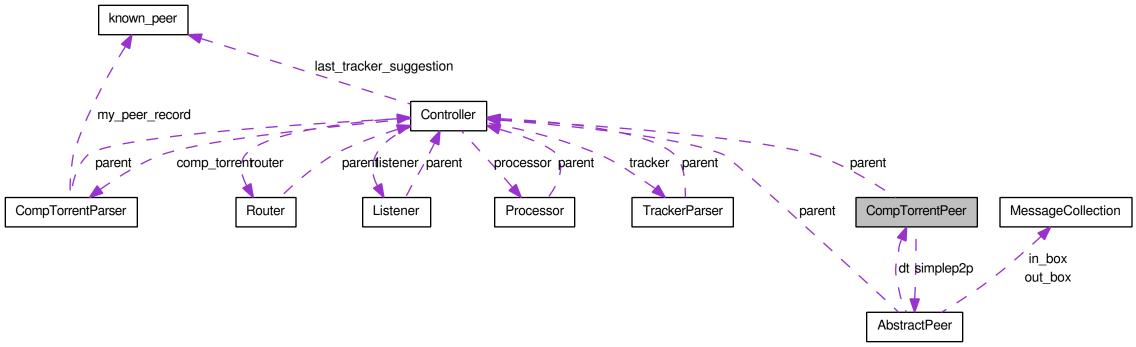
The documentation for this class was generated from the following files:

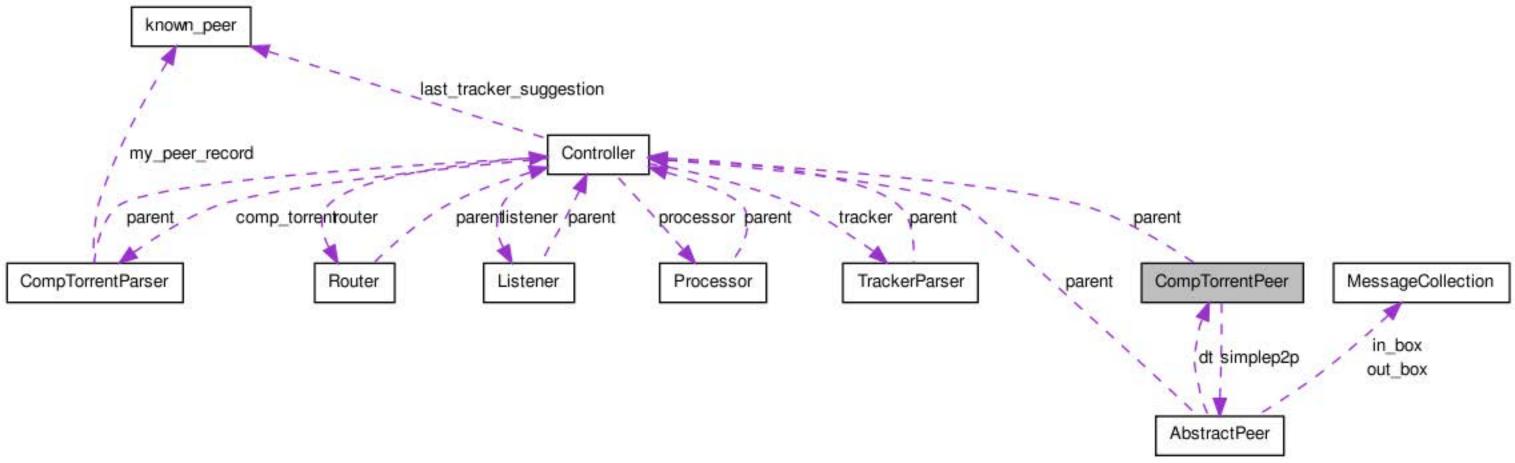
- comptorrentpeer.h
- comptorrentpeer.cpp

Generated on Sat Mar 27 12:27:43 2010 for CompTorrent by 1.6.1











CompTorrentPeer Member List

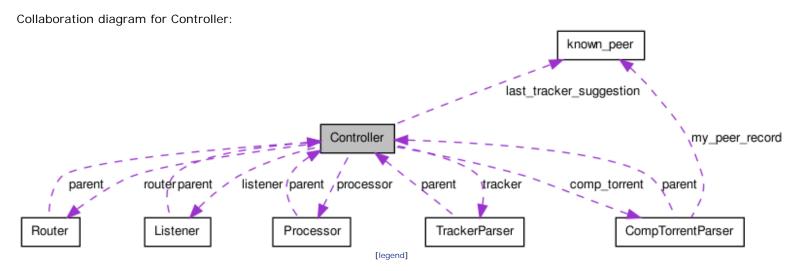
This is the complete list of members for **CompTorrentPeer**, including all inherited members.

<pre>add_available_file(string, string) (defined in CompTorrentPeer) ask_for_next_comp_file() (defined in CompTorrentPeer)</pre>	CompTorrentPeer	[protected]
ask_for_next_comp_file() (defined in CompTorrentPeer)		
	CompTorrentPeer	[protected]
ask_for_next_orig_file() (defined in CompTorrentPeer)	CompTorrentPeer	[protected]
buddy_route (defined in CompTorrentPeer)	CompTorrentPeer	[protected]
CompTorrentPeer (Controller *s, AbstractPeer *simplep2p_) (defined in CompTorrentPeer)	CompTorrentPeer	
debug_stats() (defined in CompTorrentPeer)	CompTorrentPeer	
files_available (defined in CompTorrentPeer)	CompTorrentPeer	[protected]
get_buddy_route() (defined in CompTorrentPeer)	CompTorrentPeer	[inline]
has_file_available(string filename, string type) (defined in CompTorrentPeer)	CompTorrentPeer	
is_higher_route() (defined in CompTorrentPeer)	CompTorrentPeer	
num_messages (defined in CompTorrentPeer)	CompTorrentPeer	[protected]
parent (defined in CompTorrentPeer)	CompTorrentPeer	[protected]
parse_datachunks(string xml) (defined in CompTorrentPeer)	CompTorrentPeer	[protected]
peer_type (defined in CompTorrentPeer)	CompTorrentPeer	[protected]
<pre>process_connect_message(string intmp) (defined in CompTorrentPeer)</pre>	CompTorrentPeer	[protected]
<pre>process_file_reply(string intmp) (defined in CompTorrentPeer)</pre>	CompTorrentPeer	[protected]
<pre>process_file_request(string intmp) (defined in CompTorrentPeer)</pre>	CompTorrentPeer	[protected]
<pre>process_file_update(string intmp) (defined in CompTorrentPeer)</pre>	CompTorrentPeer	[protected]
<pre>process_noop(string intmp) (defined in CompTorrentPeer)</pre>	CompTorrentPeer	[protected]
<pre>process_overlay_message(string intmp) (defined in CompTorrentPeer)</pre>	CompTorrentPeer	[protected]
<pre>process_overlay_update_message(string intmp) (defined in CompTorrentPeer)</pre>	CompTorrentPeer	[protected]
<pre>process_welcome_message(string intmp) (defined in CompTorrentPeer)</pre>	CompTorrentPeer	[protected]
<pre>push_msg(string msg) (defined in CompTorrentPeer)</pre>	CompTorrentPeer	
run() (defined in CompTorrentPeer)	CompTorrentPeer	[protected]
<pre>save_file_reply(xmlmap &xml_map) (defined in CompTorrentPeer)</pre>	CompTorrentPeer	[protected]
send_noop() (defined in CompTorrentPeer)	CompTorrentPeer	[protected]
send_welcome_message() (defined in CompTorrentPeer)	CompTorrentPeer	[protected]
simplep2p (defined in CompTorrentPeer)	CompTorrentPeer	[protected]
state (defined in CompTorrentPeer)	CompTorrentPeer	[protected]
<pre>string_find(const string needle, const string haystack) (defined in CompTorrentPeer)</pre>	CompTorrentPeer	
	CompTorrentPeer	[protected,
trim_spaces(std::string const &str) (defined in CompTorrentPeer)		bcacic]





Controller Class Reference



List of all members.

Public Member Functions

	Controller (bool index, bool is_processing)
void	debug_stats ()
void	<pre>start_distributed (string host_ip_, string host_port_)</pre>
void	<pre>start_listener (string host_ip_, string host_port_)</pre>
void	<pre>make_connection (string host_ip_, string host_port_)</pre>
void	<pre>attempt_outgoing_connection (string host_ip_, string host_port_)</pre>
void	<pre>attempt_incoming_connection (string host_ip_, string host_port_)</pre>
bool	<pre>parse_torrent (string file_path)</pre>
bool	create_working ()
void	log (const ostringstream s)
void	log (const string s)
int	get_num_connections ()
int	get_num_incoming_connections ()
int	get_num_outgoing_connections ()
void	increment_num_incoming_connections ()
void	increment_num_outgoing_connections ()
void	decrement_num_outgoing_connections ()
void	decrement_num_incoming_connections ()
void	set_tracker_url (string tracker_url_)
string	get_tracker_url ()
unsigned long	bootstrap_from_tracker ()
void	clear_tracker_data ()
void	set_routing_id (string routing_id_)
string	get_routing_id ()
string	get_next_routing_id ()
void	set_ip (string ip_)
void	set_port (string port_)
string	get_ip ()
string	get_port ()
void	set_network_device (string my_device)
string	get_network_device ()
void	report_change (bool change_)
bool	change_occured ()
bool	have_comp_chunk (string chunkname)
void	add_comp_chunk (string filename, string resulthash)

void	y 1 \ y /
int	3
bool	
void	add_required_comp_chunk (string filename)
bool	get_next_required_comp_chunk (file_chunk &f)
int	num_required_comp_chunks ()
bool	have_file (string chunkname, string filetype)
void	request_file (string chunkname, string filetype)
bool	have_orig_chunk (string chunkname)
void	add_orig_chunk (string filename, string resulthash)
void	get_known_orig_chunks_xml (ostringstream &xml)
int	get_num_orig_chunks ()
bool	remove_required_orig_chunk (string chunk_name)
void	add_required_orig_chunk (string filename)
void	add_required_orig_chunk_front (string filename)
bool	get_next_required_orig_chunk (file_chunk &f)
int	num_required_orig_chunks ()
void	<pre>add_connected_peer (string ip, string port, void *distorrent)</pre>
bool	<pre>connected_peer_exists (string ip, string port)</pre>
void	out_broadcast (string msg)
bool	remove_connected_peer (string ip, string port)
void	<pre>add_peer_under_consideration (string ip, string port)</pre>
bool	<pre>peer_under_consideration (string ip, string port)</pre>
bool	remove_considered_peer (string ip, string port)
bool	<pre>add_backlog_file (string filename, string filetype, CompTorrentPeer *who)</pre>
void	check_backlog (xmlmap &xml_map)
bool	is_index ()

Static Public Member Functions

static bool	xml_to_map (const string xml, xmlmap &xml_map)
static string	get_device_ip (string device)
static string	trim_spaces (std::string const &str)

Public Attributes

Mutex	connect_mutex
CompTorrentParser *	comp_torrent
TrackerParser *	tracker
Processor *	processor [NUM_PROCESSORS]
Mutex	parser_tracker_mutex
Mutex	get_next_comp_chunk_mutex
Mutex	get_next_orig_chunk_mutex
Mutex	comp_chunk_collection
Mutex	orig_chunk_collection

Protected Attributes

Router *	router
peer_type	connected_peers
peer_type	peers_being_considered
Mutex	peers_considered_mutex
Mutex	backlog_mutex
file_chunk_type	comp_chunks_we_want
file_chunk_type	comp_chunks_we_have
file_chunk_type	orig_chunks_we_have
file_chunk_type	orig_chunks_we_want
file_request_type	request_backlog
string	network_device
string	routing_id
bool	waiting
int	num_incoming_connections
int	num_outgoing_connections
bool	index_node

bool	is_processing
string	tracker_url
string	ip
string	port
known_peer	last_tracker_suggestion
int	wait_time
int	connection_wait
Listener *	listener
bool	routing_set
unsigned long	next_route_id
bool	change

Friends

class Router

Member Function Documentation

A static method which takes some comptorrent xml and converts it into an associative array to make life a little easier when using it (since most of the work required with the xml is just pulling tags and values out. Only works one level deep!

Parameters:

xml The input xmlxml_map The container to fill

Returns:

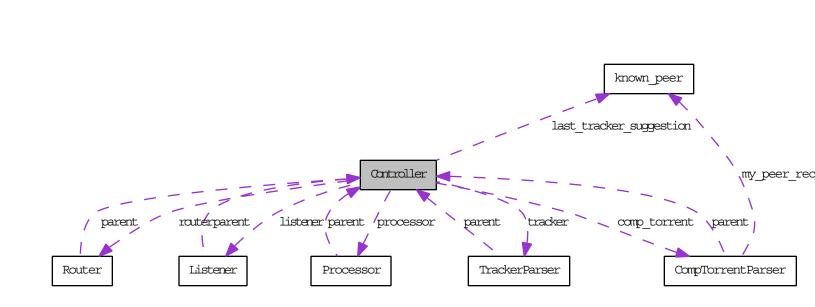
boolean success or failure

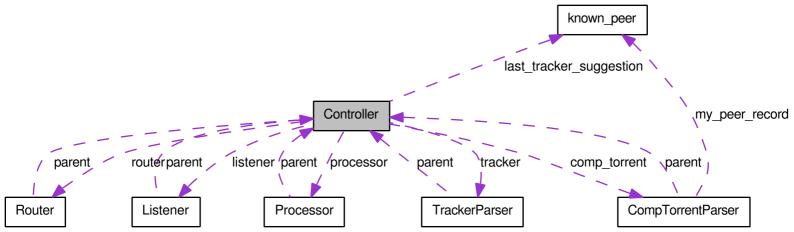
The documentation for this class was generated from the following files:

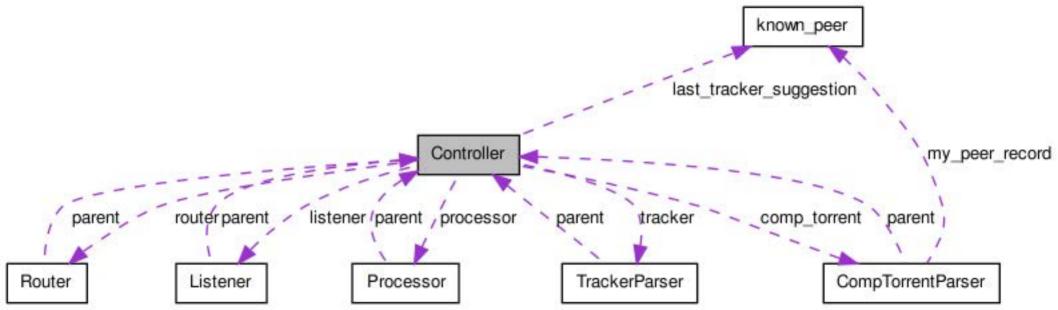
- · controller.h
- controller.cpp



Generated on Sat Mar 27 12:27:44 2010 for CompTorrent by







Main Page Classes Files

Class List Class Members

Controller Member List

This is the complete list of members for **Controller**, including all inherited members.

This is the complete list of members for controller , including all inherited members.	
<pre>add_backlog_file(string filename, string filetype, CompTorrentPeer *who) (defined in Controller)</pre>	Controller
add_comp_chunk (string filename, string resulthash) (defined in Controller)	Controller
add_connected_peer(string ip, string port, void *distorrent) (defined in Controller)	Controller
add_orig_chunk (string filename, string resulthash) (defined in Controller)	Controller
add_peer_under_consideration(string ip, string port) (defined in Controller)	Controller
add_required_comp_chunk(string filename) (defined in Controller)	Controller
add_required_orig_chunk(string filename) (defined in Controller)	Controller
add_required_orig_chunk_front(string filename) (defined in Controller)	Controller
attempt_incoming_connection (string host_ip_, string host_port_) (defined in Controller)	Controller
attempt_outgoing_connection(string host_ip_, string host_port_) (defined in Controller)	Controller
backlog_mutex (defined in Controller)	Controller [protected]
bootstrap_from_tracker() (defined in Controller)	Controller
change (defined in Controller)	Controller [protected]
change_occured() (defined in Controller)	Controller [inline]
check_backlog(xmlmap &xml_map) (defined in Controller)	Controller
clear_tracker_data() (defined in Controller)	Controller
comp_chunk_collection (defined in Controller)	Controller
comp_chunks_we_have (defined in Controller)	Controller [protected]
comp_chunks_we_want (defined in Controller)	<pre>Controller [protected]</pre>
comp_torrent (defined in Controller)	Controller
connect_mutex (defined in Controller)	Controller
connected_peer_exists(string ip, string port) (defined in Controller)	Controller
connected_peers (defined in Controller)	<pre>Controller [protected]</pre>
connection_wait (defined in Controller)	<pre>Controller [protected]</pre>
Controller (bool index, bool is_processing) (defined in Controller)	Controller
create_working() (defined in Controller)	Controller
debug_stats() (defined in Controller)	Controller
decrement_num_incoming_connections() (defined in Controller)	Controller [inline]
decrement_num_outgoing_connections() (defined in Controller)	Controller [inline]
get_device_ip(string device) (defined in Controller)	Controller [static]
get_ip() (defined in Controller)	Controller [inline]
get_known_comp_chunks_xml (ostringstream &xml) (defined in Controller)	Controller
<pre>get_known_orig_chunks_xml(ostringstream &xml) (defined in Controller)</pre>	Controller
get_network_device() (defined in Controller)	Controller [inline]
get_next_comp_chunk_mutex (defined in Controller)	Controller
get_next_orig_chunk_mutex (defined in Controller)	Controller
<pre>get_next_required_comp_chunk(file_chunk &f) (defined in Controller)</pre>	Controller
<pre>get_next_required_orig_chunk(file_chunk &f) (defined in Controller)</pre>	Controller
get_next_routing_id() (defined in Controller)	Controller
get_num_comp_chunks() (defined in Controller)	Controller [inline]
get_num_connections() (defined in Controller)	Controller [inline]
get_num_incoming_connections() (defined in Controller)	Controller [inline]
get_num_orig_chunks() (defined in Controller)	<pre>Controller [inline]</pre>
get_num_outgoing_connections() (defined in Controller)	<pre>Controller [inline]</pre>
get_port() (defined in Controller)	Controller [inline]
get_routing_id() (defined in Controller)	Controller

get_tracker_url() (defined in Controller)	Controller
have_comp_chunk(string chunkname) (defined in Controller)	Controller
have_file (string chunkname, string filetype) (defined in Controller)	Controller
have_orig_chunk(string chunkname) (defined in Controller)	Controller
increment_num_incoming_connections() (defined in Controller)	Controller [inline]
increment_num_outgoing_connections() (defined in Controller)	Controller [inline]
index_node (defined in Controller)	Controller [protected]
ip (defined in Controller)	Controller [protected]
is_index() (defined in Controller)	Controller [inline]
is_processing (defined in Controller)	Controller [protected]
last_tracker_suggestion (defined in Controller)	Controller [protected]
listener (defined in Controller)	Controller [protected]
log(const ostringstream s) (defined in Controller)	Controller [inline]
log(const string s) (defined in Controller)	Controller [inline]
make_connection(string host_ip_, string host_port_) (defined in Controller)	Controller
network_device (defined in Controller)	Controller [protected]
next_route_id (defined in Controller)	Controller [protected]
num_incoming_connections (defined in Controller)	Controller [protected]
num_outgoing_connections (defined in Controller)	Controller [protected]
num_required_comp_chunks() (defined in Controller)	Controller [inline]
num_required_orig_chunks() (defined in Controller)	Controller [inline]
orig_chunk_collection (defined in Controller)	Controller
orig_chunks_we_have (defined in Controller)	Controller [protected]
orig_chunks_we_want (defined in Controller)	Controller [protected]
out_broadcast(string msg) (defined in Controller)	Controller
<pre>parse_torrent(string file_path) (defined in Controller)</pre>	Controller
parser_tracker_mutex (defined in Controller)	Controller
<pre>peer_under_consideration(string ip, string port) (defined in Controller)</pre>	Controller
peers_being_considered (defined in Controller)	Controller [protected]
peers_considered_mutex (defined in Controller)	<pre>Controller [protected]</pre>
port (defined in Controller)	<pre>Controller [protected]</pre>
processor (defined in Controller)	Controller
remove_connected_peer(string ip, string port) (defined in Controller)	Controller
remove_considered_peer(string ip, string port) (defined in Controller)	Controller
remove_required_comp_chunk(string chunk_name) (defined in Controller)	Controller
remove_required_orig_chunk(string chunk_name) (defined in Controller)	Controller
report_change (bool change_) (defined in Controller)	Controller [inline]
request_backlog (defined in Controller)	Controller [protected]
request_file (string chunkname, string filetype) (defined in Controller)	Controller
router (defined in Controller)	Controller [protected]
Router (defined in Controller)	Controller [friend]
routing_id (defined in Controller)	Controller [protected]
routing_set (defined in Controller)	Controller [protected]
set_ip(string ip_) (defined in Controller)	Controller [inline]
set_network_device(string my_device) (defined in Controller)	Controller
set_port(string port_) (defined in Controller)	Controller [inline]
set_routing_id(string routing_id_) (defined in Controller)	Controller
set_tracker_url(string tracker_url_) (defined in Controller)	Controller
start_distributed(string host_ip_, string host_port_) (defined in Controller)	Controller
start_listener(string host_ip_, string host_port_) (defined in Controller)	Controller
tracker (defined in Controller)	Controller [protected]
tracker_url (defined in Controller) trim_spaces(std::string.const_8:str) (defined in Controller)	Controller [protected] Controller [static]
trim_spaces(std::string const &str) (defined in Controller) wait_time (defined in Controller)	Controller [protected]
wait_time (defined in Controller)	Controller [procected]

CompTorrent: Member List

waiting (defined in Controller)	Controller [protected]
xml_to_map(const string xml, xmlmap &xml_map)	<pre>Controller [static]</pre>
~Controller() (defined in Controller)	Controller

Generated on Sat Mar 27 12:27:44 2010 for CompTorrent by



CompTorrent: Alphabetical List



Class Index

AICIFIK		1 7 1 1 1 4 4		
1 - 1 1	A C F K L M N O P R T W			
F	Listener	orig_data_chunk	Router	
file_chunk	M	Р	Т	
file_chunk_request	MessageCollection	peer	TrackerParser	
K	N	Processor	W	
known_peer	NetworkSocket	R	WorkerThread	
L	0			
A C F K L M N O P R T W				
	file_chunk file_chunk_request K known_peer	file_chunk file_chunk_request MessageCollection N known_peer NetworkSocket O	file_chunk file_chunk_request MessageCollection P R N P Processor known_peer NetworkSocket R	



AbstractPeer

CompTorrentParser CompTorrentPeer Controller file_chunk file_chunk_request known_peer

Listener

MessageCollection

NetworkSocket

orig_data_chunk

peer Processor

Router

TrackerParser

WorkerThread



file_chunk Class Reference

List of all members.

Public Member Functions

```
file_chunk (string name_, string type_)
file_chunk & operator= (const file_chunk &rhs)
```

Public Attributes

string	name
string	type

The documentation for this class was generated from the following file:

known_peer.h

Generated on Sat Mar 27 12:27:44 2010 for CompTorrent by 1.6.1





file_chunk_request Class Reference

List of all members.

Public Member Functions

	<pre>file_chunk_request (string name_, string type_, void *who_)</pre>
file_chunk_request &	<pre>operator= (const file_chunk_request &rhs)</pre>

Public Attributes

string	name
string	type
void *	who

The documentation for this class was generated from the following file:

• known_peer.h



Main Page	Classes	Files	
Class List	Class Members		

file_chunk_request Member List

This is the complete list of members for **file_chunk_request**, including all inherited members.

<pre>file_chunk_request(string name_, string type_, void *who_) (defined in file_chunk_request)</pre>	file_chunk_request [inline]
file_chunk_request() (defined in file_chunk_request)	<pre>file_chunk_request [inline]</pre>
name (defined in file_chunk_request)	file_chunk_request
<pre>operator=(const file_chunk_request &rhs) (defined in file_chunk_request)</pre>	<pre>file_chunk_request [inline]</pre>
type (defined in file_chunk_request)	file_chunk_request
who (defined in file_chunk_request)	file_chunk_request



Files Main Page Classes **Class List Class Members**

file_chunk Member List

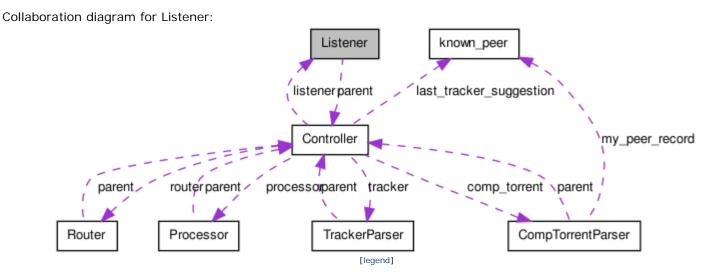
This is the complete list of members for file_chunk, including all inherited members.

<pre>file_chunk(string name_, string type_) (defined in file_chunk)</pre>	file_chunk	[inline]
file_chunk() (defined in file_chunk)	file_chunk	[inline]
name (defined in file_chunk)	file_chunk	
<pre>operator=(const file_chunk &rhs) (defined in file_chunk)</pre>	file_chunk	[inline]
type (defined in file_chunk)	file_chunk	





Listener Class Reference



List of all members.

Public Member Functions

```
Listener (Controller *parent_, string host_ip_, string host_port_)
void run ()
```

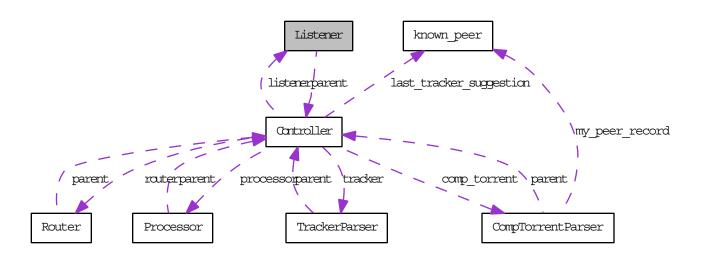
Protected Attributes

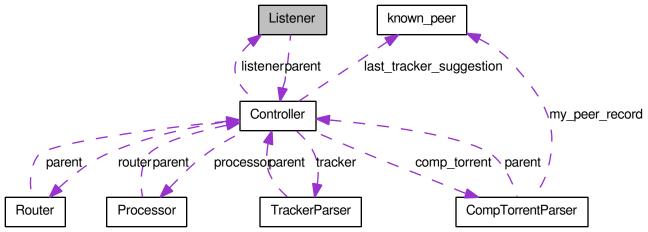
string	host_ip
string	host_port
Controller *	parent

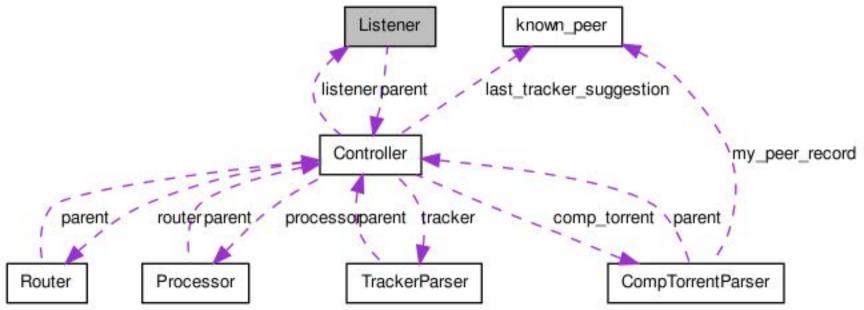
The documentation for this class was generated from the following files:

- · listener.h
- listener.cpp

Generated on Sat Mar 27 12:27:44 2010 for CompTorrent by







CompTorrent: Member List



Listener Member List

This is the complete list of members for Listener, including all inherited members.

host_ip (defined in Listener)	Listener	[protected]
host_port (defined in Listener)	Listener	[protected]
Listener(Controller *parent_, string host_ip_, string host_port_) (defined in Listener)	Listener	[inline]
parent (defined in Listener)	Listener	[protected]
run() (defined in Listener)	Listener	





MessageCollection Class Reference

List of all members.

Public Member Functions

void	<pre>push_back (const string)</pre>
void	<pre>push_front (const string)</pre>
bool	pop (string &)
collection_iter	begin ()
collection_iter	end ()
void	clear ()
unsigned long	size ()

Private Attributes

collection_type	collection
Mutex	collection_mutex
int	num_items

The documentation for this class was generated from the following files:

- messagecollection.h
- messagecollection.cpp

Generated on Sat Mar 27 12:27:44 2010 for CompTorrent by



MessageCollection Member List

This is the complete list of members for MessageCollection, including all inherited members.

begin() (defined in MessageCollection)	MessageCollection [[inline]
clear() (defined in MessageCollection)	MessageCollection	
collection (defined in MessageCollection)	MessageCollection [[private]
collection_mutex (defined in MessageCollection)	MessageCollection [[private]
end() (defined in MessageCollection)	MessageCollection [[inline]
MessageCollection() (defined in MessageCollection)	MessageCollection [[inline]
num_items (defined in MessageCollection)	MessageCollection [[private]
<pre>pop(string &) (defined in MessageCollection)</pre>	MessageCollection	
<pre>push_back(const string) (defined in MessageCollection)</pre>	MessageCollection	
<pre>push_front(const string) (defined in MessageCollection)</pre>	MessageCollection	
size() (defined in MessageCollection)	MessageCollection [inline]





NetworkSocket Class Reference

List of all members.

Public Member Functions

NetworkSocket (tpport_t port) NetworkSocket (const IPV4Address &ia, tpport_t port)

The documentation for this class was generated from the following files:

- network_socket.h
- network_socket.cpp



CompTorrent: Member List



NetworkSocket Member List

This is the complete list of members for **NetworkSocket**, including all inherited members.

NetworkSocket(tpport_t port) (defined in NetworkSocket) **NetworkSocket**

NetworkSocket (const IPV4Address &ia, tpport_t port) (defined in NetworkSocket) NetworkSocket

~NetworkSocket() (defined in NetworkSocket)

NetworkSocket

Generated on Sat Mar 27 12:27:44 2010 for CompTorrent by 1.6.1





peer Class Reference

List of all members.

Public Member Functions

```
peer (string ip_, string port_, void *distorrent_)
peer & operator= (const peer &rhs)
```

Public Attributes

string	ip
string	port
void *	distorrent

The documentation for this class was generated from the following file:

known_peer.h

Generated on Sat Mar 27 12:27:44 2010 for CompTorrent by 1.6.1





peer Member List

This is the complete list of members for **peer**, including all inherited members.

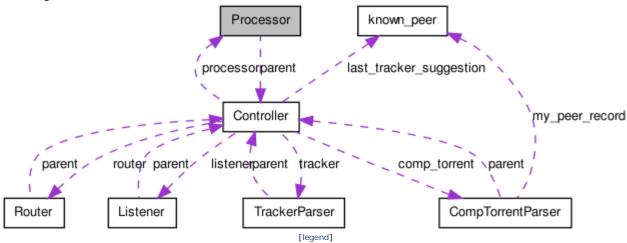
distorrent (defined in peer)	peer	
ip (defined in peer)	peer	
<pre>operator=(const peer &rhs) (defined in peer)</pre>	peer	[inline]
peer() (defined in peer)	peer	[inline]
<pre>peer(string ip_, string port_, void *distorrent_) (defined in peer)</pre>	peer	[inline]
port (defined in peer)	peer	





Processor Class Reference

Collaboration diagram for Processor:



List of all members.

Public Member Functions

	Processor (Controller *s)
void	run ()
bool	processing_finished ()

Protected Member Functions

void process_orig_chunk (string chunk_name)

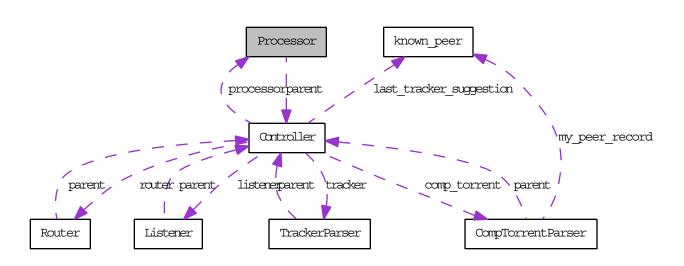
Protected Attributes

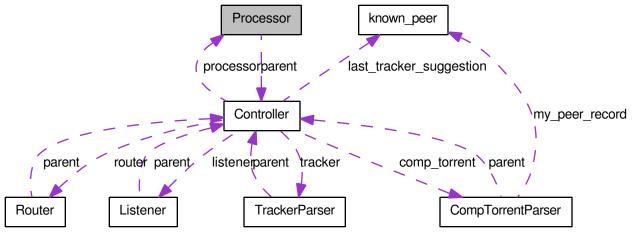
processor_state	state
Controller *	parent
string	current_chunk_name
int	wait_timeout
bool	finished

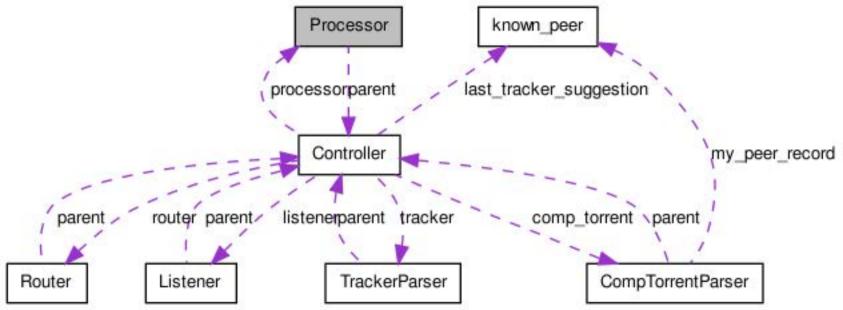
The documentation for this class was generated from the following files:

- · processor.h
- processor.cpp

Generated on Sat Mar 27 12:27:45 2010 for CompTorrent by









Processor Member List

This is the complete list of members for **Processor**, including all inherited members.

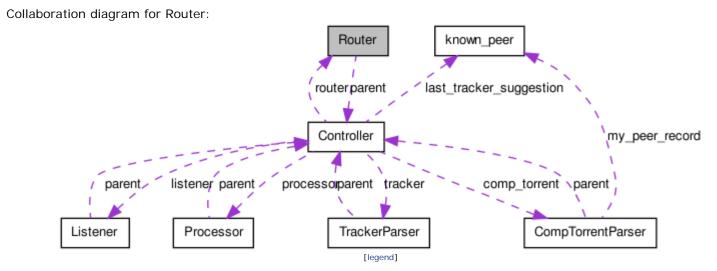
current_chunk_name (defined in Processor)	Processor	[protected]
finished (defined in Processor)	Processor	[protected]
parent (defined in Processor)	Processor	[protected]
<pre>process_orig_chunk(string chunk_name) (defined in Processor)</pre>	Processor	[protected]
processing_finished() (defined in Processor)	Processor	[inline]
Processor(Controller *s) (defined in Processor)	Processor	
run() (defined in Processor)	Processor	
state (defined in Processor)	Processor	[protected]
wait_timeout (defined in Processor)	Processor	[protected]

Generated on Sat Mar 27 12:27:45 2010 for CompTorrent by 1.6.1





Router Class Reference



List of all members.

Public Member Functions

	Router (Controller *parent_)	
void	run ()	

Protected Member Functions

bool	get_next_orig_chunk ()
bool	get_next_comp_chunk ()

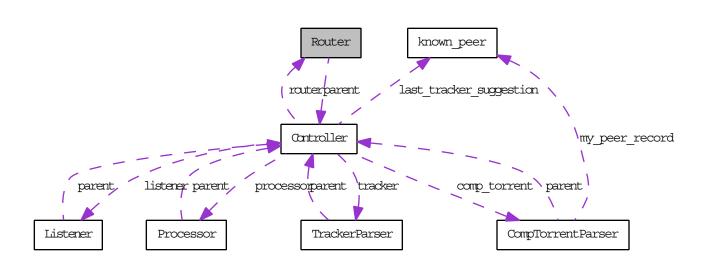
Protected Attributes

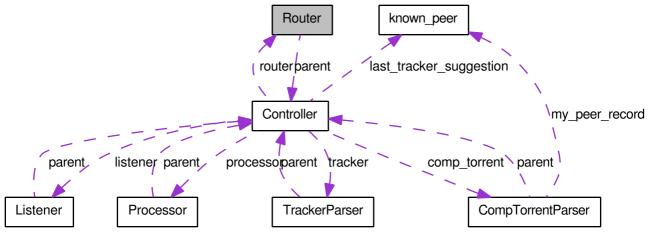
int	num_requested_orig_chunks
int	num_requested_comp_chunks
Controller *	parent

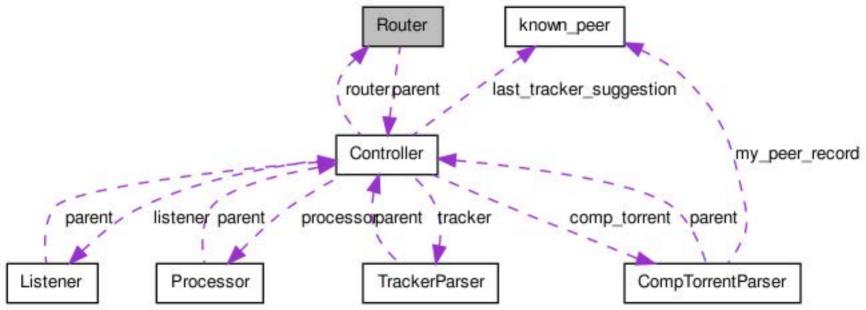
The documentation for this class was generated from the following files:

- router.h
- router.cpp

Generated on Sat Mar 27 12:27:45 2010 for CompTorrent by







CompTorrent: Member List



Router Member List

This is the complete list of members for **Router**, including all inherited members.

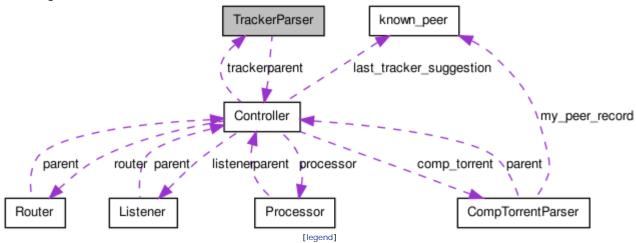
<pre>get_next_comp_chunk() (defined in Router)</pre>	Router	[protected]
<pre>get_next_orig_chunk() (defined in Router)</pre>	Router	[protected]
<pre>num_requested_comp_chunks (defined in Router)</pre>	Router	[protected]
num_requested_orig_chunks (defined in Router)	Router	[protected]
parent (defined in Router)	Router	[protected]
Router(Controller *parent_) (defined in Router)	Router	[inline]
run() (defined in Router)	Router	





TrackerParser Class Reference

Collaboration diagram for TrackerParser:



List of all members.

Public Member Functions

	TrackerParser (Controller *s)	
void	init ()	
bool	clear_xml ()	
bool	load_xml ()	
bool	send_work_xml ()	
bool	get_data_from_tracker (string url, string &data)	
bool	report_stats_to_tracker ()	
void	set_tracker_url (string tracker_url_)	
string	get_tracker_url ()	
unsigned long	y = _ =, _ ::	
bool	pop_next_known_peer (known_peer &kp)	
void	copy_chunks_from_comptorrentparser ()	
bool	prime_chunk_records ()	
bool	get_next_chunk_name (string &chunk_name)	
bool	<pre>set_chunk_done (string chunk_name, string resulthash, string orig_chunk_hash, bool report)</pre>	
bool	report_file_exists (string resulthash, string filename)	
bool		
bool		
bool		
bool	get_comp_hash (string chunk_name, string &resulthash)	
bool	G = 1 = 1	
bool	<pre>get_next_missing_comp_chunk_name (string &chunk_name)</pre>	
void	register_node (string ip, string port, string uuid, string comptorrentname, int routeid)	
bool	stats (string type, string statistic)	
bool	report_connection (string server, string client)	
bool	suggest_peer_to_try (known_peer &kp)	
bool	report_ipconnection (string client, string server)	
bool	ipconnection_exists (string client, string server)	
bool	get_route_to_file (const string chunk_hash, string &routeid)	
void	debug ()	

Protected Member Functions

bool	get_xml_from_tracker ()
bool	send_data_to_tracker (string data)
bool	post_data_to_tracker (string data)
void	add_known_peer (known_peer kp)
bool	known_peer_exists (known_peer kp)
void	tokenize (string &str, vector< string > &tokens, string delimiters)

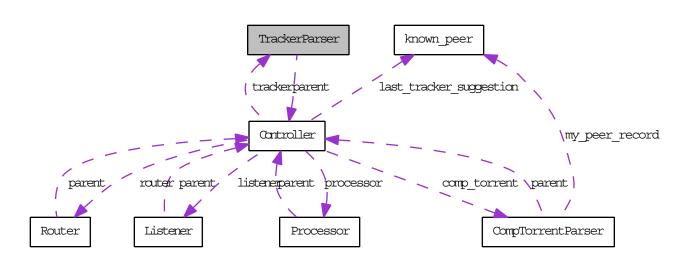
Protected Attributes

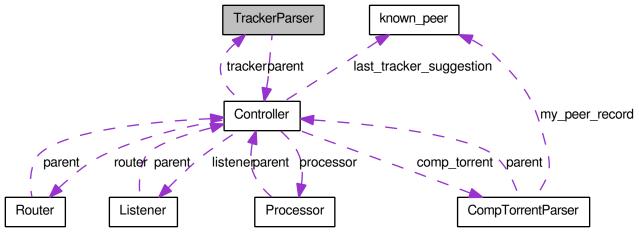
TiXmlDocument doc Controller * parent string string tracker_url string tracker_xml vector< known_peer > known_peers vector< orig_data_chunk > data_chunks unsigned long unsigned original_chunks unsigned long original_chunks unsigned long original_hits int stat_get_num_known_peers int stat_is_suitable_peer int stat_copy_chunks_from_comptorrentparser int stat_copy_chunks_from_comptorrentparser int stat_set_chunk_records int stat_set_chunk_done int stat_set_chunk_done int stat_set_chunk_done int stat_get_next_missing_corig_chunk_name int stat_get_comp_hash int stat_get_next_missing_orig_chunk_name int stat_get_next_missing_orig_chunk_name int stat_report_files_exists int stat_report_files_exists int stat_report_connection int stat_report_connection int stat_suggest_peer_to_try int stat_dok_nown_peer int stat_known_peer_exists int stat_known_peer_exists int stat_known_peer_exists int stat_sut_cont_connection int stat_known_peer_exists int stat_sut_cont_connection int stat_sut_known_peer_exists int stat_sut_cont_connection int stat_sut_nown_peer_exists int stat_sut_cont_connection int stat_sut_nown_peer_exists int stat_sut_nown_peer_exists int stat_sut_cont_connection int stat_sut_cont_connection int stat_sut_nown_peer_exists int stat_sut_nown_peer_exists int stat_total_in int stat_total_in int stat_total_out	Mutex	curl_mutex
string tracker_url string tracker_xml vector< known_peer > known_peers vector< orig_data_chunk > data_chunks unsigned long unsigned long original_chunks int stat_get_num_known_peers int stat_jop_next_known_peer int stat_is_suitable_peer int stat_opy_chunks_from_comptorrentparser int stat_prime_chunk_records int stat_set_chunk_done int stat_set_chunk_done int stat_set_chunks_done int stat_get_next_missing_orig_chunk_name int stat_get_next_missing_comp_chunk_name int stat_report_file_exists int stat_report_files_exists int stat_report_files_exists int stat_report_files_exists int stat_report_files_exists int stat_report_files_exists int stat_report_files_exists int stat_report_connection int stat_get_route_to_file int stat_get_route_to_file int stat_set_orig_data int stat_total_in int stat_total_in int stat_total_out		
string tracker_xml vector < known_peer > known_peers vector < orig_data_chunk > data_chunks unsigned long computed_chunks unsigned long original_hits int stat_get_num_known_peers int stat_pop_next_known_peer int stat_copy_chunks_from_comptorrentparser int stat_prime_chunk_records int stat_get_next_chunk_name int stat_set_chunk_done int stat_set_chunks_done int stat_report_file_exists int stat_get_comp_hash int stat_get_next_missing_orig_chunk_name int stat_get_next_missing_comp_chunk_name int stat_report_files_exists int stat_report_connection int stat_suggest_peer_to_try int stat_suggest_peer_to_try int stat_suggest_peer_to_file int stat_set_orig_data int stat_total_in int stat_total_out	Controller *	parent
vector known_peer vector orig_data_chunk > data_chunks unsigned long unsigned long original_chunks original_chunks unsigned int stat_get_num_known_peers int stat_pop_next_known_peer int stat_is_suitable_peer int stat_copy_chunks_from_comptorrentparser int stat_prime_chunk_records int stat_prime_chunk_name int stat_set_chunk_done int stat_set_chunks_done int stat_report_file_exists int stat_get_comp_hash int stat_get_next_missing_orig_chunk_name int stat_get_next_missing_comp_chunk_name int stat_report_files_exists int stat_report_files_exists int stat_report_connection int stat_register_node int stat_suggest_peer_to_try int stat_add_known_peer int stat_add_known_peer int stat_known_peer_exists int stat_stat_total_in int stat_total_out	string	tracker_url
vector< orig_data_chunk > data_chunks unsigned long unsigned original_chunks original_hits int stat_get_num_known_peers int stat_pop_next_known_peer int stat_copy_chunks_from_comptorrentparser int stat_prime_chunk_records int stat_set_chunk_done int stat_set_chunk_done int stat_report_file_exists int stat_get_next_missing_orig_chunk_name int stat_get_next_missing_comp_chunk_name int stat_report_files_exists int stat_report_files_exists int stat_report_files_exists int stat_report_files_exists int stat_get_next_missing_comp_chunk_name int stat_report_files_exists int stat_report_connection int stat_report_connection int stat_suggest_peer_to_try int stat_suggest_peer_to_file int stat_stat_nown_peer int stat_stat_nown_peer_exists int stat_stat_lotal_in int stat_total_in int stat_total_in	string	tracker_xml
vector< orig_data_chunk > data_chunks unsigned long unsigned original_chunks original_hits int stat_get_num_known_peers int stat_pop_next_known_peer int stat_copy_chunks_from_comptorrentparser int stat_prime_chunk_records int stat_set_chunk_done int stat_set_chunk_done int stat_report_file_exists int stat_get_next_missing_orig_chunk_name int stat_get_next_missing_comp_chunk_name int stat_report_files_exists int stat_report_files_exists int stat_report_files_exists int stat_report_files_exists int stat_get_next_missing_comp_chunk_name int stat_report_files_exists int stat_report_connection int stat_report_connection int stat_suggest_peer_to_try int stat_suggest_peer_to_file int stat_stat_nown_peer int stat_stat_nown_peer_exists int stat_stat_lotal_in int stat_total_in int stat_total_in	vector< known_peer >	known_peers
unsigned long unsigned int stat_get_num_known_peers int stat_pop_next_known_peer int stat_is_suitable_peer int stat_copy_chunks_from_comptorrentparser int stat_prime_chunk_records int stat_get_next_chunk_name int stat_set_chunk_done int stat_report_file_exists int stat_get_next_missing_orig_chunk_name int stat_get_next_missing_orig_chunk_name int stat_report_files_exists int stat_get_next_missing_orig_chunk_name int stat_report_files_exists int stat_report_files_exists int stat_report_files_exists int stat_report_files_exists int stat_report_files_exists int stat_report_connection int stat_register_node int stat_register_node int stat_suggest_peer_to_try int stat_get_route_to_file int stat_add_known_peer int stat_known_peer_exists int stat_stat_loid_in int stat_total_in		data_chunks
unsigned original_hits int stat_get_num_known_peers int stat_pop_next_known_peer int stat_is_suitable_peer int stat_copy_chunks_from_comptorrentparser int stat_prime_chunk_records int stat_get_next_chunk_name int stat_set_chunk_done int stat_set_chunks_done int stat_report_file_exists int stat_get_next_missing_orig_chunk_name int stat_get_next_missing_comp_chunk_name int stat_get_next_missing_comp_chunk_name int stat_report_files_exists int stat_report_files_exists int stat_report_connection int stat_register_node int stat_register_node int stat_suggest_peer_to_try int stat_get_route_to_file int stat_add_known_peer int stat_known_peer_exists int stat_set_orig_data int stat_stat_lotal_in int stat_total_in	unsigned long	computed_chunks
int stat_get_num_known_peers int stat_pop_next_known_peer int stat_is_suitable_peer int stat_copy_chunks_from_comptorrentparser int stat_prime_chunk_records int stat_get_next_chunk_name int stat_set_chunk_done int stat_set_chunks_done int stat_report_file_exists int stat_get_comp_hash int stat_get_next_missing_orig_chunk_name int stat_get_next_missing_comp_chunk_name int stat_report_files_exists int stat_report_files_exists int stat_report_connection int stat_report_connection int stat_suggest_peer_to_try int stat_add_known_peer int stat_known_peer_exists int stat_set_orig_data int stat_total_in int stat_total_out	unsigned long	original_chunks
int stat_pop_next_known_peer int stat_is_suitable_peer int stat_copy_chunks_from_comptorrentparser int stat_prime_chunk_records int stat_pet_next_chunk_name int stat_set_chunk_done int stat_set_chunks_done int stat_report_file_exists int stat_get_next_missing_orig_chunk_name int stat_get_next_missing_comp_chunk_name int stat_get_next_missing_comp_chunk_name int stat_report_files_exists int stat_report_files_exists int stat_register_node int stat_register_node int stat_regort_connection int stat_suggest_peer_to_try int stat_add_known_peer int stat_add_known_peer int stat_subject_noid_ int st	unsigned	original_hits
int stat_is_suitable_peer int stat_copy_chunks_from_comptorrentparser int stat_prime_chunk_records int stat_get_next_chunk_name int stat_set_chunk_done int stat_set_chunks_done int stat_report_file_exists int stat_get_comp_hash int stat_get_next_missing_orig_chunk_name int stat_get_next_missing_comp_chunk_name int stat_report_files_exists int stat_report_files_exists int stat_report_connection int stat_report_connection int stat_suggest_peer_to_try int stat_get_route_to_file int stat_add_known_peer int stat_set_orig_data int stat_total_in int stat_total_out	int	stat_get_num_known_peers
int stat_copy_chunks_from_comptorrentparser int stat_prime_chunk_records int stat_get_next_chunk_name int stat_set_chunk_done int stat_set_chunks_done int stat_report_file_exists int stat_get_next_missing_orig_chunk_name int stat_get_next_missing_comp_chunk_name int stat_report_files_exists int stat_report_files_exists int stat_register_node int stat_report_connection int stat_suggest_peer_to_try int stat_get_route_to_file int stat_add_known_peer int stat_set_orig_data int stat_total_in int stat_total_out	int	stat_pop_next_known_peer
int stat_prime_chunk_records int stat_get_next_chunk_name int stat_set_chunk_done int stat_set_chunks_done int stat_report_file_exists int stat_get_comp_hash int stat_get_next_missing_orig_chunk_name int stat_get_next_missing_comp_chunk_name int stat_report_files_exists int stat_report_files_exists int stat_register_node int stat_regort_connection int stat_suggest_peer_to_try int stat_get_route_to_file int stat_add_known_peer int stat_known_peer_exists int stat_set_orig_data int stat_total_in int stat_total_out	int	stat_is_suitable_peer
int stat_get_next_chunk_name int stat_set_chunk_done int stat_set_chunks_done int stat_report_file_exists int stat_get_comp_hash int stat_get_next_missing_orig_chunk_name int stat_get_next_missing_comp_chunk_name int stat_report_files_exists int stat_report_files_exists int stat_register_node int stat_report_connection int stat_suggest_peer_to_try int stat_get_route_to_file int stat_add_known_peer int stat_add_known_peer_exists int stat_set_orig_data int stat_total_in int stat_total_out	int	stat_copy_chunks_from_comptorrentparser
int stat_set_chunk_done int stat_set_chunks_done int stat_report_file_exists int stat_get_comp_hash int stat_get_next_missing_orig_chunk_name int stat_get_next_missing_comp_chunk_name int stat_report_files_exists int stat_register_node int stat_report_connection int stat_suggest_peer_to_try int stat_get_route_to_file int stat_add_known_peer int stat_add_known_peer int stat_set_orig_data int stat_total_in int stat_total_out	int	stat_prime_chunk_records
int stat_set_chunks_done int stat_report_file_exists int stat_get_comp_hash int stat_get_next_missing_orig_chunk_name int stat_get_next_missing_comp_chunk_name int stat_report_files_exists int stat_report_connection int stat_report_connection int stat_suggest_peer_to_try int stat_get_route_to_file int stat_add_known_peer int stat_known_peer_exists int stat_set_orig_data int stat_total_in int stat_total_out	int	stat_get_next_chunk_name
int stat_report_file_exists int stat_get_comp_hash int stat_get_next_missing_orig_chunk_name int stat_get_next_missing_comp_chunk_name int stat_report_files_exists int stat_register_node int stat_report_connection int stat_suggest_peer_to_try int stat_get_route_to_file int stat_add_known_peer int stat_add_known_peer int stat_set_orig_data int stat_total_in int stat_total_out	int	stat_set_chunk_done
int stat_get_comp_hash int stat_get_next_missing_orig_chunk_name int stat_get_next_missing_comp_chunk_name int stat_report_files_exists int stat_register_node int stat_report_connection int stat_suggest_peer_to_try int stat_get_route_to_file int stat_add_known_peer int stat_known_peer_exists int stat_set_orig_data int stat_total_in int stat_total_out	int	stat_set_chunks_done
int stat_get_next_missing_orig_chunk_name int stat_get_next_missing_comp_chunk_name int stat_report_files_exists int stat_register_node int stat_report_connection int stat_suggest_peer_to_try int stat_get_route_to_file int stat_add_known_peer int stat_known_peer_exists int stat_set_orig_data int stat_total_in int stat_total_out	int	stat_report_file_exists
int stat_get_next_missing_comp_chunk_name int stat_report_files_exists int stat_register_node int stat_report_connection int stat_suggest_peer_to_try int stat_get_route_to_file int stat_add_known_peer int stat_known_peer_exists int stat_set_orig_data int stat_total_in int stat_total_out	int	stat_get_comp_hash
int stat_report_files_exists int stat_register_node int stat_report_connection int stat_suggest_peer_to_try int stat_get_route_to_file int stat_add_known_peer int stat_known_peer_exists int stat_set_orig_data int stat_total_in int stat_total_out	int	stat_get_next_missing_orig_chunk_name
int stat_register_node int stat_report_connection int stat_suggest_peer_to_try int stat_get_route_to_file int stat_add_known_peer int stat_known_peer_exists int stat_set_orig_data int stat_total_in int stat_total_out	int	stat_get_next_missing_comp_chunk_name
int stat_report_connection int stat_suggest_peer_to_try int stat_get_route_to_file int stat_add_known_peer int stat_known_peer_exists int stat_set_orig_data int stat_total_in int stat_total_out		
int stat_suggest_peer_to_try int stat_get_route_to_file int stat_add_known_peer int stat_known_peer_exists int stat_set_orig_data int stat_total_in int stat_total_out	int	stat_register_node
int stat_get_route_to_file int stat_add_known_peer int stat_known_peer_exists int stat_set_orig_data int stat_total_in int stat_total_out	int	stat_report_connection
int stat_add_known_peer int stat_known_peer_exists int stat_set_orig_data int stat_total_in int stat_total_out	int	stat_suggest_peer_to_try
int stat_known_peer_exists int stat_set_orig_data int stat_total_in int stat_total_out	int	stat_get_route_to_file
int stat_set_orig_data int stat_total_in int stat_total_out	int	<u> </u>
int stat_total_in int stat_total_out		
int stat_total_out		
int stat_report_stats_to_tracker	int	
	int	stat_report_stats_to_tracker

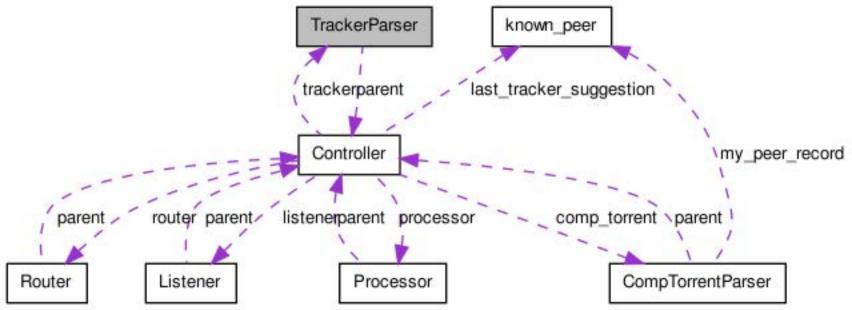
The documentation for this class was generated from the following files:

- · trackerparser.h
- trackerparser.cpp

Generated on Sat Mar 27 12:27:46 2010 for CompTorrent by 1.6.1







Main Page Classes Files

Class List Class Members

TrackerParser Member List

This is the complete list of members for **TrackerParser**, including all inherited members. add_known_peer(known_peer kp) (defined in TrackerParser) TrackerParser [protected] clear_xml() (defined in TrackerParser) TrackerParser computed_chunks (defined in TrackerParser) TrackerParser [protected] copy_chunks_from_comptorrentparser() (defined in TrackerParser) **TrackerParser** curl_mutex (defined in TrackerParser) TrackerParser [protected] TrackerParser [protected] data_chunks (defined in TrackerParser) debug() (defined in TrackerParser) TrackerParser [inline] doc (defined in TrackerParser) TrackerParser [protected] get_comp_hash(string chunk_name, string &resulthash) (defined in TrackerParser) **TrackerParser** get_data_from_tracker(string url, string &data) (defined in TrackerParser) **TrackerParser** get_next_chunk_name(string &chunk_name) (defined in TrackerParser) **TrackerParser** get_next_missing_comp_chunk_name(string &chunk_name) (defined in **TrackerParser** TrackerParser) get_next_missing_orig_chunk_name(string &chunk_name) (defined in **TrackerParser** TrackerParser) get_num_known_peers() (defined in TrackerParser) **TrackerParser TrackerParser** get_route_to_file(const string chunk_hash, string &routeid) (defined in TrackerParser) get_tracker_url() (defined in TrackerParser) TrackerParser [inline] get_xml_from_tracker() (defined in TrackerParser) TrackerParser [protected] init() (defined in TrackerParser) TrackerParser ipconnection_exists(string client, string server) (defined in TrackerParser) **TrackerParser** known_peer_exists(known_peer kp) (defined in TrackerParser) **TrackerParser** [protected] known_peers (defined in TrackerParser) TrackerParser [protected] load_xml() (defined in TrackerParser) **TrackerParser** original_chunks (defined in TrackerParser) TrackerParser [protected] original_hits (defined in TrackerParser) **TrackerParser** [protected] parent (defined in TrackerParser) TrackerParser [protected] pop_next_known_peer (known_peer &kp) (defined in TrackerParser) **TrackerParser** post_data_to_tracker(string data) (defined in TrackerParser) TrackerParser [protected] prime_chunk_records() (defined in TrackerParser) **TrackerParser** register_node(string ip, string port, string uuid, string comptorrentname, int routeid) **TrackerParser** (defined in TrackerParser) report_connection(string server, string client) (defined in TrackerParser) **TrackerParser** report_file_exists(string resulthash, string filename) (defined in TrackerParser) **TrackerParser** TrackerParser report_ipconnection(string client, string server) (defined in TrackerParser) report_stats_to_tracker() (defined in TrackerParser) TrackerParser send_data_to_tracker(string data) (defined in TrackerParser) TrackerParser [protected] send_work_xml() (defined in TrackerParser) **TrackerParser** set_chunk_done(string chunk_name, string resulthash, string orig_chunk_hash, bool **TrackerParser** report) (defined in **TrackerParser**) set_chunks_done(string comp_hash) (defined in TrackerParser) **TrackerParser** set_orig_data(string orig_data) (defined in TrackerParser) **TrackerParser** set_tracker_url(string tracker_url_) (defined in TrackerParser) TrackerParser [inline] stat_add_known_peer (defined in TrackerParser) TrackerParser [protected] stat_copy_chunks_from_comptorrentparser (defined in TrackerParser) TrackerParser [protected] stat_get_comp_hash (defined in TrackerParser) **TrackerParser** [protected] stat_get_next_chunk_name (defined in TrackerParser) TrackerParser [protected] stat_get_next_missing_comp_chunk_name (defined in TrackerParser) TrackerParser [protected]

stat_get_next_missing_orig_chunk_name (defined in TrackerParser)	TrackerParser [protected]
stat_get_num_known_peers (defined in TrackerParser)	TrackerParser [protected]
stat_get_route_to_file (defined in TrackerParser)	TrackerParser [protected]
stat_is_suitable_peer (defined in TrackerParser)	TrackerParser [protected]
stat_known_peer_exists (defined in TrackerParser)	TrackerParser [protected]
stat_pop_next_known_peer (defined in TrackerParser)	TrackerParser [protected]
stat_prime_chunk_records (defined in TrackerParser)	TrackerParser [protected]
stat_register_node (defined in TrackerParser)	TrackerParser [protected]
stat_report_connection (defined in TrackerParser)	TrackerParser [protected]
stat_report_file_exists (defined in TrackerParser)	TrackerParser [protected]
stat_report_files_exists (defined in TrackerParser)	TrackerParser [protected]
stat_report_stats_to_tracker (defined in TrackerParser)	TrackerParser [protected]
stat_set_chunk_done (defined in TrackerParser)	TrackerParser [protected]
stat_set_chunks_done (defined in TrackerParser)	TrackerParser [protected]
stat_set_orig_data (defined in TrackerParser)	TrackerParser [protected]
stat_suggest_peer_to_try (defined in TrackerParser)	TrackerParser [protected]
stat_total_in (defined in TrackerParser)	TrackerParser [protected]
stat_total_out (defined in TrackerParser)	TrackerParser [protected]
stats(string type, string statistic) (defined in TrackerParser)	TrackerParser
<pre>suggest_orig_chunks(string lastchunk, string &xml_suggestions) (defined in TrackerParser)</pre>	TrackerParser
<pre>suggest_peer_to_try(known_peer &kp) (defined in TrackerParser)</pre>	TrackerParser
tokenize(string &str, vector< string > &tokens, string delimiters) (defined in TrackerParser)	TrackerParser [protected]
tracker_url (defined in TrackerParser)	TrackerParser [protected]
tracker_xml (defined in TrackerParser)	TrackerParser [protected]
TrackerParser(Controller *s) (defined in TrackerParser)	TrackerParser
~TrackerParser() (defined in TrackerParser)	TrackerParser





WorkerThread Class Reference

List of all members.

Public Member Functions

virtual void run ()

The documentation for this class was generated from the following files:

- worker_thread.h
- worker_thread.cpp

Generated on Sat Mar 27 12:27:46 2010 for CompTorrent by 1.6.1



CompTorrent: Member List

Files Main Page Classes Class List **Class Members**

WorkerThread Member List

This is the complete list of members for **WorkerThread**, including all inherited members.

run() (defined in WorkerThread) WorkerThread [virtual]

WorkerThread() (defined in WorkerThread) WorkerThread ~WorkerThread() (defined in WorkerThread) WorkerThread

Generated on Sat Mar 27 12:27:46 2010 for CompTorrent by 1.6.1





Main Page Classes Files
File List

comptorrentparser.h

```
00001 #ifndef _CompTorrentParser_h_
00002 #define _CompTorrentParser_h_
00003
00004 #include <iostream>
00005 #include <cstdlib>
00006 #include <list>
00007 #include <map>
00008 #include "tinyxml/tinystr.h"
00009 #include "tinyxml/tinyxml.h"
00010 #include "SP2PCommon.h"
00011 #include "messagecollection.h"
00012 #include "known peer.h"
00013 #include "orig_data_chunk.h"
00014 #include "controller.h"
00015
00016 class Controller; // forward declaration
00017
00018 typedef map<string,string> filemap;
00019
      typedef filemap::iterator filemap_iter;
00020
00021 class CompTorrentParser {
00022
00023 public:
00024
00025
               CompTorrentParser(Controller* parent_);
00026
00027
               bool load xml(string);
00028
               bool create_working();
00029
00030
               string get_version();
00031
               string get_tracker_url();
               string get_tracker_port();
00032
00033
               string get_name();
00034
               string get_size();
00035
               string get_md5();
00036
               string get_algorithm_base64();
string get_algorithm_filename();
00037
00038
               string get_execution();
00039
00040
               unsigned long get_num_orig_file_chunks();
00041
00042
               bool comp_data_exists(string filename);
00043
               bool orig_data_exists(string filename);
00044
00045
00046
00047
               bool known_peer_exists(known_peer kp);
00048
00049
               void add_known_peer(known_peer kp);
00050
               void remove_known_peer(known_peer kp);
00051
00052
00053
               unsigned long get_num_known_peers();
00054
               unsigned long get_num_connected_peers();
00055
00056
               string chunk_name(unsigned long chunk_num);
00057
               string chunk_orig_hash(string chunk_name);
00058
               bool get_orig_hash(string chunk_name, string& resulthash);
00059
00060
               void
                       update_chunk(unsigned long chunk_num, computation_state state, bool
orig_file_exists, bool comp_file_exists);
               void
00061
                      update_chunk(string chunk_name, computation_state state, bool
orig_file_exists, bool comp_file_exists);
00062
00063
               void
                      update_comp_chunk(string chunk_name, computation_state state, bool
file exists);
               void
00064
                       update_orig_chunk(string chunk_name, computation_state state, bool
file_exists);
00065
00066
               unsigned long get num chunks();
00067
00068
               static int my file size( const string filepath );
```

```
00069
00070
               bool has comp converged (computation state state);
00071
00072
00073
00074
               string get_node_uuid();
00075
               unsigned long get_num_computed_chunks();
unsigned long get_num_original_chunks();
00076
00077
00078
               void increment_num_computed_chunks();
00079
08000
00081
00082
               vector<orig_data_chunk> data_chunk_list;
00083
00084
               filemap known files;
00085
00086
               void add directory to file(string filename, string comp or data, string&
filepath);
00087
               void strip_directory_from_file(string filepath, string& chunk_name);
00088
00089
               void calculate_file_hash(string filepath, string& resulthash);
00090
00091 protected:
00092
00093
              bool loaded ok;
00094
00095
               string get UUID(string device);
00096
00097
               string version, tracker_url, tracker_port, name, size, md5, algorithm_base64,
algorithm_filename, execution, uuid;
00098
00099
               vector<known peer> known peers;
00100
               known_peer my_peer_record;
00101
00102
               Mutex status_mutex;
00103
00104
               friend class Processor;
00105
               friend class CompTorrent;
00106
               friend class DistTorrent;
00107
00108
               unsigned long num_computed_chunks;
               unsigned long num_original_chunks;
00109
00110
00111
               Controller* parent;
00112
00113 };
00114
00115 #endif
```

Generated on Sat Mar 27 12:27:34 2010 for CompTorrent by 1.6.1





comptorrentpeer.h

```
00001 #include <time.h>
00002
00003 #include "abstractpeer.h"
00004 #include "worker thread.h"
00005 #include "comptorrentparser.h" 00006 #include "processor.h"
00007
00008 #ifndef _DistTorrent_h_
00009 #define _DistTorrent_h_
00010
00011 enum distributed_state { p_unconnected, p_overlay_wait, p_processing, p_connect_sent,
p data only };
00012
00013 class AbstractPeer; // forward decl
00014
00015 class CompTorrentPeer : public Thread {
00016
00017 public:
00018
00019
               CompTorrentPeer(Controller* s, AbstractPeer* simplep2p );
00020
               ~CompTorrentPeer();
               void push_msg(string msg);
bool has_file_available(string filename, string type);
00021
00022
00023
               void debug_stats();
00024
               string get_buddy_route() { return buddy_route; }
00025
               bool is higher route();
00026
00027 protected:
00028
00029
               AbstractPeer* simplep2p;
00030
               Controller* parent;
00031
00032
               bool add_available_file(string, string);
00033
00034
00035
               static bool string_find(const string needle, const string haystack);
00036
00037
00038
               void send welcome message();
               void send_noop();
00039
00040
               void run();
00041
00042
00043
               bool process_noop(string intmp);
               bool process_file_update(string intmp);
00044
00045
               bool process_welcome_message(string intmp);
00046
               bool process_connect_message(string intmp);
00047
               bool process_file_reply(string intmp);
00048
               bool process_file_request(string intmp);
00049
               bool process_overlay_message(string intmp);
00050
               bool process_overlay_update_message(string intmp);
00051
00052
               bool ask_for_next_comp_file();
00053
               bool ask_for_next_orig_file();
00054
00055
00056
               int num_messages;
00057
00058
               void save file reply(xmlmap& xml map);
00059
                                                           // am I the server or client for this
00060
               short peer_type;
connection
00061
               distributed_state state;
                                                  // state
00062
00063
               string buddy_route;
                                       // the route id of who I'm connected to
00064
00065
00066
               void parse_datachunks(string xml);
00067
               file chunk type files available; // the files that this connection has
00068
available
00069
```

```
00070
00071
                   static string trim_spaces( std::string const& str);
00071
00072
00073 };
00074
00075
00076 #endif
```

Generated on Sat Mar 27 12:27:34 2010 for CompTorrent by 1.6.1





controller.h

```
00001
00002 #ifndef _SP2P_h_
00003 #define _SP2P_h_
00004
00005 #include <iostream>
00006 #include <cc++/xml.h>
00007 #include "network_socket.h"
00008 #include "comptorrentparser.h"
00009 #include "processor.h"
00010 #include "messagecollection.h"
00011 #include "trackerparser.h"
00012 #include "router.h"
00013
00014 using namespace std;
00015 using namespace ost;
00016
00017 class Listener;
                                            // forward declaration
                                            // forward declaration
// forward declaration
00018 class TrackerParser;
00019 class CompTorrentParser;
00020 class CompTorrentPeer;
                                            // forward declaration
                                            // forward declaration
00021 class Router;
00022
00023 typedef map<string,string> xmlmap;
00024 typedef xmlmap::iterator xmlmap_iter;
00025
00026 class Controller {
00027
00028 public:
00029
00030
                Controller(bool index, bool is_processing);
00031
                ~Controller();
00032
00033
                void debug_stats();
00034
00035
                void start_distributed(string host_ip_, string host_port_);
00036
                void start_listener(string host_ip_, string host_port_);
00037
00038
                void make_connection(string host_ip_, string host_port_);
00039
                void attempt_outgoing_connection(string host_ip_, string host_port_);
00040
00041
                void attempt_incoming_connection(string host_ip_, string host_port_);
00042
00043
                bool parse_torrent(string file_path);
00044
                bool create_working();
00045
00046
                void log(const ostringstream s) { std::cout << s << "\n"; }</pre>
00047
                void log(const string s) { std::cout << s << "\n"; }</pre>
00048
                int get_num_connections() { return num_incoming_connections +
00049
num_outgoing_connections;
00050
                int get_num_incoming_connections() { return num_incoming_connections;
00051
                int get_num_outgoing_connections() { return num_outgoing_connections; }
00052
00053
                Mutex connect mutex;
00054
                void increment_num_incoming_connections()
void increment_num_outgoing_connections()
void decrement_num_outgoing_connections()
void decrement_num_incoming_connections()
{
    num_incoming_connections--;
    num_incoming_connections--;
}
00055
00056
00057
00058
00059
00060
                void set tracker url(string tracker url_);
00061
                string get_tracker_url();
00062
                unsigned long bootstrap_from_tracker();
00063
00064
                void clear_tracker_data();
00065
00066
                CompTorrentParser* comp_torrent;
00067
                TrackerParser* tracker;
                Processor* processor[NUM_PROCESSORS];
00068
00069
00070
                Mutex parser tracker mutex;
00071
                Mutex get next comp chunk mutex;
```

```
Mutex get next orig chunk mutex;
00073
00074
               static bool xml_to_map(const string xml, xmlmap& xml_map);
00075
               static string get_device_ip(string device);
00076
00077
               void set_routing_id(string routing_id_);
00078
               string get_routing_id();
00079
               string get next routing id();
00080
00081
               void set_ip(string ip_) { ip = ip_; }
00082
               void set_port(string port_) { port = port_; }
00083
00084
               string get_ip() { return ip; }
00085
               string get_port() { return port; }
00086
00087
               static string trim spaces ( std::string const& str);
00088
00089
               void set_network_device(string my_device);
               string get_network_device() { return network_device;
void report_change(bool change_) { change = change_;
00090
00091
00092
               bool change_occured() { return change; }
00093
00094
00095
               bool have_comp_chunk(string chunkname);
00096
               void add_comp_chunk(string filename, string resulthash);
               void get_known_comp_chunks_xml(ostringstream& xml);
int get_num_comp_chunks() { return comp_chunks_we_have.size(); }
00097
00098
00099
00100
               bool remove_required_comp_chunk(string chunk_name);
00101
               void add_required_comp_chunk(string filename);
00102
               bool get_next_required_comp_chunk(file_chunk& f);
00103
               int num_required_comp_chunks() { return comp_chunks_we_want.size(); }
00104
00105
               bool have_file(string chunkname, string filetype);
               void request file(string chunkname, string filetype);
00106
00107
00108
               bool have_orig_chunk(string chunkname);
00109
               void add_orig_chunk(string filename, string resulthash);
               void get_known_orig_chunks_xml(ostringstream& xml);
int get_num_orig_chunks() { return orig_chunks_we_have.size(); }
00110
00111
00112
00113
               bool remove_required_orig_chunk(string chunk_name);
00114
               void add_required_orig_chunk(string filename);
00115
               void add_required_orig_chunk_front(string filename);
00116
               bool get_next_required_orig_chunk(file_chunk& f);
               int num_required_orig_chunks() { return orig_chunks_we_want.size(); }
00117
00118
               void add_connected_peer(string ip, string port, void* distorrent);
00119
00120
               bool connected_peer_exists(string ip, string port);
               void out_broadcast(string msg);
00121
00122
               bool remove_connected_peer(string ip, string port);
00123
00124
               void add_peer_under_consideration(string ip, string port);
00125
               bool peer_under_consideration(string ip, string port);
               bool remove_considered_peer(string ip, string port);
00126
00127
               bool add backlog file(string filename, string filetype, CompTorrentPeer*
00128
who);
00129
               void check_backlog(xmlmap& xml_map);
00130
               bool is_index() { return index_node; }
00131
00132
00133
               Mutex comp_chunk_collection;
00134
               Mutex orig_chunk_collection;
00135
               friend class Router;
00136
00137
00138 protected:
00139
00140
               Router* router;
00141
00142
               peer_type connected_peers;
00143
               peer_type peers_being_considered;
00144
               Mutex peers_considered_mutex;
00145
               Mutex backlog_mutex;
00146
00147
               file_chunk_type comp_chunks_we_want;
00148
               file_chunk_type comp_chunks_we_have;
               file_chunk_type orig_chunks_we_have;
file_chunk_type orig_chunks_we_want;
00149
00150
00151
               file request type request backlog; // files that connected nodes have
```

```
asked for that I don't have but have requested on their behalf
00152
00153
              string network_device;
00154
              string routing_id;
              bool waiting;
00155
00156
              int num_incoming_connections;
00157
              int num_outgoing_connections;
                                                                                  // am I the
              bool index_node;
00158
index (seed) node
00159
                                                                                  // am I to
              bool is_processing;
do processing? This is more for stats gathering (to c.f. cli/serv architectures) than a
design feature
              string tracker_url;
00160
00161
              string ip, port;
00162
00163
              known_peer last_tracker_suggestion;
00164
              int wait_time;
00165
00166
              int connection_wait;
00167
              Listener* listener;
00168
              bool routing set;
              unsigned long next_route_id;
00169
00170
              bool change;
00171
00172 };
00173
00174 #endif
```

Generated on Sat Mar 27 12:27:34 2010 for CompTorrent by



Dhrystone Version 2.1 and 1.1 benchmark

It will be best to first create a directory (e.g., 'dhrystone')

to

hold all the files.

1. UNPACKING dhry.shar

Do the following to unpack dhry.shar:

sh dhry.shar

This creates all the dhrystone files as listed in the 'MANIFEST' file.

2. CHANGING COMPILER OPTIONS

To change compiler options edit 'Makefile'.

The compiler default options are: '-DUNIX -0'.
You can change this, for example, to: '-DUNIX'-DROPT -0',
where '-DROPT' specifies the 'Register' OPTion. Various compilers
also come with, generally, numerous other options that you are
free to select of course. See 'dhry.tbl' for example.

```
'-DUNIX' is in the 'DEFINES =' row of 'Makefile'.
'-O' is in the 'FLAGS =' row of 'Makefile'.
```

The '-DUNIX' option is one of several timer options avaliable.

The timer option MUST be specified for the program to compile.

The various timer options available are covered in 'timers_b.c' which you may need to examine to find the right match for your machine. Here are the current timer choices:

```
'-DAmiga'
                    for Amiga systems.
  '-DUNIX_Old' for UNIX systems (preferred)
'-DVMS' for VMS systems (avoid if possible)
   '-DBORLAND_C' for PC's with Borland C
                     for PC's with Microsoft C
   '-DMSC'
  '-DMAC' for Macintosh systems with Think C
'-DIPSC' for i860 IPSC systems
  '-DFORTRAN_SEC' for Cray systems (preferred)
  '-DGTODay' for UNIX systems
   '-DCTimer'
                  for Cray systems (avoid if possible) for Fujitsu UXP/M systems.
   '-DUXPM'
'-DMAC_TMgr' for Macintosh (preferred)
'-DPARIX' for PARIX
  '-DPOSIX'
                    for POSIX :)
'-DWTN32'
                  for WINDOWS NT
```

3. COMPILING WITH MAKE

Do the following to compile with make:

```
make dhry11
```

This creates the executable files: dhry21 and dhry11. Run both programs and submit results. Please see 'dhryrun.doc' file. 'dhry21' is the Dhrystone V2.1 program, and 'dhry11' is the Dhrystone V1.1 program.

4. COMPILING OTHERWISE

Do the following to compile without 'make' if necessary: [using the default compiler options which you can change of course to suite your machine]

```
cc -DUNIX -O -o dhry21 dhry_1.c dhry_2.c timers_b.c
```

This creates the executable program 'dhry21' which is the Dhrystone V2.1 program.

```
cc -DUNIX -O -o dhryll dhryll.c timers_b.c
```

This creates the executable program 'dhry11' which is the Dhrystone V1.1 program.

Run both dhry21 and dhry11 and submit results. See 'dhryrun.doc' file.

5. RUNNING THE PROGRAM

Do the following to run the programs:

dhry11 dhry21

6. REPORTING RESULTS

Edit the ascii file 'dhryrun.doc' and please add in the following:

EMAIL ADDRESS: [flora_walker@hitech.edu]

DATE: 22 Aug 1942

Email 'dhryrun.doc' to aburto@nosc.mil and I will include in, and maintain, table of results. I will also post results periodically to 'comp.benchmarks'. Results and other programs are available via anonymous ftp to 'ftp.nosc.mil' in directory 'pub/aburto'. The IP address is: 128.49.192.51

Thank you very much ...

Al Aburto aburto@nosc.mil 09 Oct 1996

DHRYSTONE 2.1 and 1.1 BENCHMARK REPORTING FORM

MACHINE/MODEL:

CPU:

CPU CLOCK SPEED:

OS/VERSION:

COMPILER/VERSION: COMPILER OPTIONS:

MIPS Rating (V1.1): MIPS Rating (V2.1):

COMMENT:

NAME:

EMAIL ADDRESS:

DATE:

MAIL TO: aburto@nosc.mil

RESULTS:

PHP Base Library Documentation, Release phplib_7_2

Boris Erdmann, boris@erdmann.com, Kristian Köhntopp, kk@netuse.de and Sascha Schumann, sascha@schumann.cx

\$Id: documentation.sgml,v 1.11 2000/02/24 23:33:02 sasha Exp \$

1. Quick Start

- 1.1 License
- 1.2 Target Group and Prerequisites
- 1.3 Quick Guide to Installation
- 1.4 Using core features of PHPLIB
- <u>1.5 Testing</u>

2. Overview and Installation

- 2.1 Files, classes and functions
- 2.2 Downloading and unpacking the distribution
- 2.3 Requirements and things to check for
- 2.4 Installation procedure
- 2.5 Using include() instead of auto prepend file=
- 2.6 PHPLIB with mod_php (Apache module)

3. Core Functionality

- 3.1 DB_Sql
- 3.2 Page Management
- 3.3 CT Sal
- 3.4 CT_Split_Sql
- 3.5 CT Shm
- 3.6 CT Dbm
- 3.7 CT_Ldap
- <u>3.8 Session</u>
- 3.9 Auth
- 3.10 Perm
- 3.11 User

4. Extended functionality

- 4.1 Cart
- 4.2 Template

5. HTML Widgets Classes

- <u>5.1 Sql_Query</u>
- 5.2 Table and CSV_Table
- <u>5.3 Menu</u>
- <u>5.4 Form</u>
- <u>5.5 tpl_form</u>
- <u>5.6 Tree</u>
- 5.7 STRINGS2 function set

6. Acknowledgments

Next Previous Contents

PHP Base Library Documentation, Release phplib_7_2
Boris Erdmann, boris@erdmann.com, Kristian Köhntopp,
kk@netuse.de and Sascha Schumann, sascha@schumann.cx
\$Id: documentation.sgml,v 1.11 2000/02/24 23:33:02 sasha Exp
\$

Table of Contents

1. Quick Start

- 1.1 License
- 1.2 Target Group and Prerequisites
- 1.3 Quick Guide to Installation
- 1.4 Using core features of PHPLIB
- 1.5 Testing

2. Overview and Installation

- 2.1 Files, classes and functions
 - 2.1.1 Customization
 - 2.1.2 Core functionality
 - 2.1.3 Extended functionality
 - 2.1.4 HTML widgets
- 2.2 Downloading and unpacking the distribution
- 2.3 Requirements and things to check for
 - 2.3.1 Interpreter requirements
 - 2.3.2 Database requirements
 - 2.3.3 Name space requirements
 - 2.3.4 Year 2000 compliance statement
- 2.4 Installation procedure
- 2.5 Using
- 2.6 PHPLIB with mod_php (Apache module)

3. Core Functionality

- 3.1 DB_Sql
 - 3.1.1 Instance variables
 - 3.1.2 Instance methods
 - 3.1.2.1 Accessible instance methods
 - 3.1.2.2 Internal instance methods
 - 3.1.3 Example
 - 3.1.4 Additional information about database connections
 - 3.1.5 Using
- 3.2 Page Management
 - 3.2.1 Accessible Functions
 - 3.2.2 Example
 - 3.2.3 The "cart" feature is gone
- 3.3 CT_Sql
 - 3.3.1 Instance variables

- 3.3.2 Example
- 3.4 CT Split Sql
 - 3.4.1 Instance variables
 - 3.4.2 Example
- 3.5 CT Shm
 - 3.5.1 Instance variables
 - 3.5.2 Example
- 3.6 CT Dbm
 - 3.6.1 Instance variables
 - 3.6.2 Example
- 3.7 CT_Ldap
 - 3.7.1 Instance variables
 - 3.7.2 Example
- 3.8 Session
 - 3.8.1 Instance variables
 - 3.8.2 Instance methods
 - 3.8.2.1 Accessible instance methods
 - 3.8.2.2 Internal instance methods
 - 3.8.3 Example
 - 3.8.4 Using "auto_init"
 - 3.8.5 Unregistering variables and deleting sessions
 - 3.8.6 Reading and understanding session data for debugging
 - 3.8.7 How "serialize()" operates
- 3.9 Auth
 - 3.9.1 Instance variables
 - 3.9.2 Instance methods
 - 3.9.2.1 Accessible instance methods
 - 3.9.2.2 Internal instance methods
 - 3.9.3 Example
 - 3.9.4 Using default authentication
 - 3.9.5 Using Challenge-Response Authentication
 - 3.9.6 The complete guide to authentication and user variables
 - 3.9.6.1 How is the
 - 3.9.6.2 How does
 - 3.9.6.3 How do \$sess and \$auth interact?
 - 3.9.6.4 Where is the beef?
 - 3.9.6.5 I still do not understand! What am I supposed to code?
 - 3.9.6.6 Ok, I did that and it works. I even understood it. Now, what exactly is that uid used for?
 - 3.9.6.7 But is the uid used internally by PHPLIB?
- 3.10 Perm
 - 3.10.1 Instance variables
 - 3.10.2 Instance methods
 - 3.10.2.1 Accessible instance methods
 - 3.10.2.2 Internal instance methods
 - 3.10.3 Example
 - 3.10.4 How permissions work
- 3.11 User
 - 3.11.1 Instance variables
 - 3.11.2 Instance methods
 - 3.11.2.1 Accessible instance methods
 - 3.11.2.2 Internal instance methods
 - **3.11.3** Example

4. Extended functionality

- 4.1 Cart
 - 4.1.1 Instance variables
 - 4.1.2 Instance methods
 - 4.1.2.1 Accessible instance methods
 - 4.1.3 Example
 - 4.1.4 On using Cart
- 4.2 Template
 - 4.2.1 Instance variables
 - 4.2.2 Instance methods
 - 4.2.2.1 Accessible instance methods
 - 4.2.2.2 Internal instance methods
 - 4.2.3 Example

5. HTML Widgets Classes

- 5.1 Sql_Query
 - 5.1.1 Instance variables
 - 5.1.2 Instance methods
 - 5.1.2.1 Accessible instance methods
 - 5.1.2.2 Internal instance methods
 - 5.1.3 Example
- 5.2 Table and CSV_Table
 - 5.2.1 Instance variables
 - 5.2.2 Instance methods
 - 5.2.2.1 High-level instance methods
 - 5.2.2.2 Mid-level instance methods
 - 5.2.2.3 Low-level instance methods
 - 5.2.3 Example
- 5.3 Menu
 - 5.3.1 Instance variables
 - 5.3.2 Instance methods
 - 5.3.2.1 Accessible instance methods
 - 5.3.2.2 Internal instance methods
 - 5.3.3 Example
- 5.4 Form
 - 5.4.1 Using OOH Forms
 - 5.4.2 Customizing OOH Forms
- 5.5 tpl form
 - 5.5.1 Instance variables
 - 5.5.2 Instance methods
 - 5.5.2.1 Accessible instance methods
 - 5.5.2.2 Internal instance methods
 - 5.5.3 Example
- 5.6 Tree
 - 5.6.1 Instance variables
 - 5.6.2 Instance methods
 - 5.6.2.1 Accessible instance methods
 - 5.6.3 The Tree Array
 - 5.6.4 Example
 - 5.6.5 Known Bugs / Tips
- 5.7 STRINGS2 function set

6. Acknowledgments

11.. QQuuiicckk SSttaarrtt

The Quick Start chapter tries to give you a ten-minute introduction to PHPLIB installation, outlines a few simple testing procedures and closes with an overview of PHPLIB features.

11..11.. LLiicceennssee

PHPLIB consists of the files in this directory and all its subdirectories. It is made available as free software under the LIBRARY GNU General Public license, as spelled out in the file COPYING in this directory. Also, it is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the license for more details.

11..22.. TTaarrggeett GGrroouupp aanndd PPrreerreeqquuiissiitteess

PHPLIB targets the PHP application developer. You need to have good knowledge of the PHP language, at least basic SQL database knowhow and at least basic knowledge on how to operate your web server to be able to use the library.

The library will help you to write medium to large sized data-driven web applications. "Medium to large sized applications" are applications that consist of multiple database queries, have to generate tables from database data, need a user interface that generates SQL queries or need a comfortable and user-friendly way to protect pages or functionality on pages. "Data-driven" applications are applications that make use of a supported SQL-database to create HTML content and that use HTML forms to drive database transactions.

To make use of the library you obviously need access to a web server with a working installation of a current PHP interpreter (we recommend 3.0.12 or newer for this release of the library) and access to a supported SQL database (currently, PHPLIB supports MySQL, PostgreSQL, mSQL, Oracle 7 and Oracle 8, Sybase, Microsoft SQL Server and ODBC databases). You need to be able to create and drop database tables in that database and your web server must be able to execute SELECT, INSERT, UPDATE and DELETE statements on these tables.

Throughout this manual, we assume that you are using the MySQL database server. PHPLIB will run with any supported SQL server, but we are using MySQL in the development of PHPLIB.

PHPLIB can be used in conjunction with the CGI version of PHP and with mod_php, integrated into Apache. Usage of the CGI version has an impact on overall speed, because you cannot take advantage of persistent database connection. We recommend the Apache module over the CGI version, although we personally use the CGI version for various reasons (easier to update and can be run with Apache suexec).

PHP 4 is still in beta. We do not support deployment of this library with beta software.

11..33.. QQuuiicckk GGuuiiddee ttoo IInnssttaallllaattiioonn

These instructions apply to PHPLIB running with CGI PHP. Most of them are valid for mod_php as well, though. _V_E_R_Y _I_M_P_O_R_T_A_N_T _N_O_T_E: This is a quick installation guide to get you started if you have an installation where you control the web server, PHP interpreter and database server completely. They are not suitable for a web hosting setup where you have only limited to no control over the installation. Refer to Chapter 2 of this documentation for the complete installation instructions and troubleshooting information.

Before installing PHPLIB, get your web server up and running and have it executing files with the extension .php3. Check that with a simple <?php phpinfo() ?> script. Make sure the web server accepts index.php3 as well as index.html as a default file for URLs ending in "/" (Apache: DirectoryIndex index.html index.php3).

Get your MySQL database server up an running. Create an empty database for your application and make sure the owner of your web server processes can access this database with SELECT, INSERT, UPDATE and DELETE access. Don't forget the mysqladmin reload after changing the user and db tables.

SStteepp 11

Create an include directory named php parallel to your web servers document root directory. Do not put the include directory below your web servers document root.

SStteepp 22

Unpack your PHPLIB distribution. Move the contents of the php distribution directory into the php directory you just created.

SStteepp 33

Get to the php3.ini file for your web servers PHP interpreter and update the include_path statement so that it points to that php directory. Update the auto_prepend_file statement so that it points to the prepend.php3 file in that include directory.

If you do not have control over your php3.ini file, you did not read the _V_E_R_Y _I_M_P_O_R_T_A_N_T _N_O_T_E above.

SStteepp 44

Also check that track_vars are enabled and that you have enabled magic_quotes_gpc. While you are at it, you might want to check sendmail_path, if you plan to send mail from your application. It has to be set to /usr/lib/sendmail -t on most UNIX systems to work.

If you do not have control over your php3.ini file, you did not read the _V_E_R_Y _I_M_P_O_R_T_A_N_T _N_O_T_E above.

SStteepp 55

cd into the php include directory. Edit local.inc. In class DB_Example supply the appropriate parameters for your database connection.

SStteepp 66

For this database, run create_database.mysql from the distribution to create active_sessions and auth_user. auth_user will be populated with a sample user named kris with a password test.

SStteepp 77

Move the contents of the pages directory and all its subdirectories into your document root directory.

SStteepp 88

Access the "/" URL of your web server with cookies enabled. If no index.html is present, index.php3 will be displayed. If you reload that page, the number shown must increment. Access your database with the mysql command client and select * from active_sessions. Check that there is a single session record for your browser and see how the text in val changes when you reload the page and select * from active_sessions again. If this works, the session class is functional with cookie mode.

SStteepp 99

Now access showoff.php3. Try to login as kris, password test. Check active_sessions again. You now should have a Example_Session entry (see the name column) and a Example_User entry in your table. Both should increment on reload.

SStteepp 1100

Try again with cookies disabled. You should get a new session (the cookie is lost) and you should be able to see your session

id as the get parameter part of your URL.

11.	.44	UUssiinngg ccc	oorree ffeeaattuurreess	ooff PPHHP	PLLIIBB
		C C D D I I I I I C C C	Joined Heedadiladileess		

Many applications don't use PHPLIB's advanced features, but see PHPLIB as a convenient way to protect pages or functionality with passwords. This section covers such core functionality usage of PHPLIB.

CCuussttoommiizziinngg tthhee llooggiinn ssccrreeeenn Edit loginform.ihtml in the include directory to suit your needs.

CCuussttoommiizziinngg tthhee ppeerrmmiissssiioonn lleevveellss Edit local.inc and change the class Example_Perm to enumerate your permissions. Your users in auth_user must have one or more comma separated permission names from that list. Edit perminvalid.ihtml for a suitable error message.

CCrreeaattiinngg NNeeww UUsseerrss

Use new_user.php3 from the pages/admin directory of the distribution. If you followed the installation instructions, it should be available under the /admin URL of your web server.

To manually create a user, run print md5(uniqid("some magic string") to get a user id. insert into auth_user values ("that userid", "username", "password", "permissions");.

CCrreeaattiinngg aann uunnpprrootteecctteedd sseessssiioonn ppaaggee Begin that page with

<?php page_open(array("sess" => "Example_Session")); ?>

End that page with

<?php page_close(); ?>

CCrreeaattiinngg aa pprrootteecctteedd sseesssiioonn ppaaggee Begin that page with

```
<?php
page_open(
array("sess" => "Example_Session",
        "auth" => "Example_Auth",
        "perm" => "Example_Perm"));
$perm->check("desired protection");
?>
```

and end that page with

```
<?php page_close(); ?>
```

CCrreeaattiinngg pprrootteecctteedd ffuunnccttiioonnaalliittyy Begin that page with

and end that page with

```
<?php page_close(); ?>
```

Enclose the protected functionality in

```
<?php
  if ($perm->have_perm("desired protection")):
?>
Put protected HTML or PHP here
<?php
  endif
?>
```

_N_o_t_e_: desired protection is any combination of permissions from Example_Perm. Using the default values from Example_Perm, "user", "user,author" or "admin" are all valid sample values. A user can access a page, if that user has all permissions that are being requested in a \$perm->check() or \$perm->have_perm() call.

_N_o_t_e_: Users can have multiple permission in their perms column of auth_user. A user with perms "user,author,editor" can access all pages requesting any combination of these permissions.

_N_o_t_e_: Don't use spaces. "user, author, editor" works. "user, author, editor" does not.

_N_o_t_e_: If \$auth->auth["uid"] is set on a protected page _a_n_d if (time < auth->auth["exp"]), then and only then the authentication is valid. You may then use \$auth->auth["uname"] as the user name, \$auth->auth["uid"] as a unique user id and \$auth->auth["perm"] for the current permissions of that user. Actually, you never want to touch \$auth->auth["perm"] manually, but use \$perm->have_perm("...") for that.

GGeettttiinngg aa ggrriipp oonn PPHHPPLLIIBB

Read on. Then read the source. Read it again - Session->serialize() and Auth->start() are ugly. Get a CVS account. Contribute. Become famous. Buy a ferrari.

_N_o_t_e_: You want to understand what registered variables are. You want to understand in what order form variables and session variables are imported into your page. You want to understand how to copy values from form values into session values without killing yourself. You do not want to make form variables persistent, ever. Then you will live happily thereafter...

11..55.. TTeessttiinngg

These instructions apply to PHPLIB running with CGI PHP. Most of them is valid for mod_php as well, though. This section offers an incremental approach to find installation problems, should the above installation process fail.

We do have a support mailing list available under the address phplib@lists.netuse.de. To subscribe to the list, send the command subscribe to the address phplib-request@lists.netuse.de.

CChheecckkiinngg tthhaatt tthhee wweebb sseerrvveerr iiss uupp aanndd rruunnnniinngg Make sure your web server is up and serving the virtual host you just set up. To do this, construct a small file test1.html in your DocumentRoot and access test1.html through your web server.

CChheecckkiinngg tthhaatt tthhee wweebb sseerrvveerr iiss eexxeeccuuttiinngg CCGGII pprrooggrraammss Make sure your web server is up and does run CGI. Check the current directory, the UID/GID it is running programs under and have a look at the environment variables. Install the shell script

```
#! /bin/sh --
echo "Content-Type: text/plain"
echo
id
echo
pwd
echo
env | sort
echo
```

in your cgi directory under the name of cgi-test and in your document root under the name of cgi-test.cgi. Make it executable. Try to access /cgi/cgi-test?par1=one&par2=two and /cgi-test.cgi?par1=one&par2=two and check the output. What UID/GID are you running under, what is the output of pwd and what environment variables are set? What does QUERY_STRING look like? What does the PATH variable look like, what does the LD_LIBRARY_PATH variable look like and are all libraries needed by PHP accessible to PHP

running in the CGI environment (Check by running the Unix ldd command on PHP).

In particular, if you built Oracle support into PHP and linked libclntsh dynamically: Can it be loaded from the CGI environment? If not, PHP will not come up later in the next step.

CChheecckkiinngg tthhaatt tthhee PPHHPP iinntteerrpprreetteerr iiss rruunnnniinngg ((AAssssuummiinngg CCGGII

PHP)" Copy your PHP binary into the cgi binary directory (which should NOT be below DocumentRoot!) and make it executable. Copy php3.ini into the same directory. In DocumentRoot, create a test2.php3 and put <?php phpinfo() ?> into it.

Are you running Apache? Add

Action php3-script /cgi/php AddHandler php3-script .php3 DirectoryIndex index.php3 index.html index.htm FancyIndexing on

to your config. This will map all requests to files ending in .php3 to the php3-script handler and define /cgi/php as the URL handling php3-script requests internally.

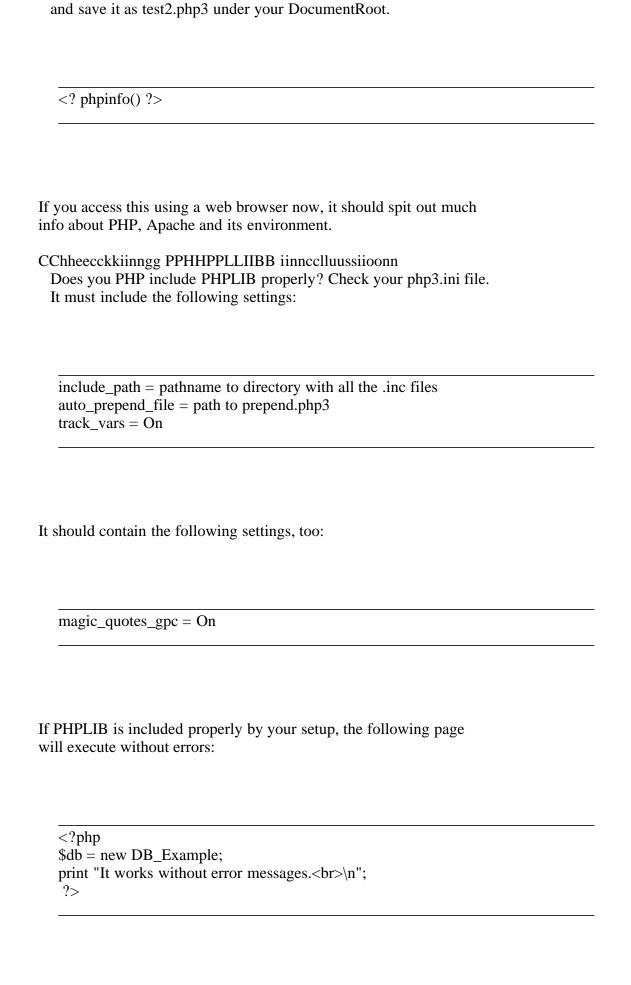
Request /test2.php3 and see that it is being executed. Make changes to your php3.ini (preferable some color definitions) and reload. Are they reflected in the output of phpinfo()? If not, your php3.ini is not being found and your are having a problem. Recompile with proper settings.

Check the output of phpinfo() carefully! Is your PHP version current (We have tested and developed this release with PHP 3.0.12)? Are your database interfaces present in the output of phpinfo()? If not, recompile again.

Can you access /test2.php3 under the URL /cgi/php/test2.php3 as well? If so, you did not compile your PHP interpreter with --enable-force-cgi-redirect. PHPLIB will not work with this interpreter. Recompile with the switch being set.

PPHHPP iinntteerrpprreetteerr ((AAssssuummiinngg mmoodd__pphhpp))

Assuming your server is already correctly setup (don't forget to activate the PHP lines in srm.conf!), enter the following file



CChheecckkiinngg ddaattaabbaassee ccoonnnneeccttiivviittyy PHPLIB installation requires that you adapt local.inc properly. Particularly, the provided class DB_Example must be customized for your database connection. Test that your web server can access the database with the following page:

```
<?php
include("table.inc"); // requires include_path to be functioning

$db = new DB_Example;
$db->query("select * from auth_user");

$t = new Table;
$t->heading = "on";
$t->show_result($db);
?>
```

When executing properly, this page will show you the user entry for kris, password test, permissions admin from the auth_user table. If this does not happen, your DB_Example definition in local.inc is broken.

CChheecckkiinngg tthhaatt sseessssiioonnss wwoorrkk Access the page /index.php3 that has been provided with the distribution. This page will try to set a cookie in your browser. Allow that cookie to be set.

The page will display a headline with a counter. Reload that page. The counter must increment. If not, either your browser cannot deal properly with cookies or PHPLIB cannot properly read or write the table active_sessions in your database. Check that the cookie is being set by viewing the output of phpinfo() (the fourth table will report the cookie and other per-call data). Check your database permissions with your database command line interface.

CChheeckkiinngg tthhaatt AAuutthheennttiiccaattiioonn wwoorrkkss Try loading /showoff.php3 that has been provided with the distribution. This page will require a login. Login as kris, using a password of test. If the login is successful, you will see the per-session counter and a per-user counter again. Reload that page: The counters must increment.

If you can't login, you probably have a problem with cookies. Check again that your browser accepts and sends session cookies. Another problem may be access to the auth_user table. You must be able to SELECT on that table and there must be at an entry for the user you are trying to login.

22.. OOvveerryviieeww aanndd IInnssttaallllaattiioonn

The following sections discuss the installation, verification and layout of PHPLIB: How to install PHPLIB? Which functionality and class definitions are contained in which files? How do you layout a web server with PHPLIB installed? Which installation options are available and how do these affect performance?

22..11.. FFiilleess,, ccllaasssseess aanndd ffuunnccttiioonnss

PHPLIB contains a set of core classes and functions that offer session tracking, per-session and per-user persistent variables, user authentication and permission checking. Building upon this core functionality, PHPLIB offers a set of commonly needed "background" classes and a set of "HTML widgets", classes that allow you to quickly generate HTML based user interfaces.

All PHPLIB definitions are designed that you don't need to change any of these files. Your customization of PHPLIB can be contained in two or three files, depending on the setup: local.inc, setup.inc and, in some cases, prepend.php3. You _N_E_V_E_R need to change any other file with PHPLIB. Details are outlined below.

22..11..11.. CCuussttoommiizzaattiioonn

The following three files are the only files from PHPLIB that require changes in normal PHPLIB applications.

AApppplliiccaattiioonn ccoonnffiigguurraattiioonn iinn local.inc: Your application will almost certainly not work with the default values supplied by the above classes. You are supposed to extend the classes described below as you see fit.

In your subclasses, you only have to specify what is different in your application. These are things like database host names, database names, table names and username/password combinations. You need to provide login screen definitions (HTML) and user

validation functions (SQL) to make the example work.

The distribution provides a local.inc to illustrate a typical setup. These definitions are also needed to get the administration and testing scripts provided with the distribution to run.

The file is required and you must change it for your setup.

AApppplliiccaattiioonn sseettuupp iinn setup.inc: The Session class provides the ability to execute initialization code at session setup. See the class description for instructions on how to set this up.

Per convention, we store such code in setup.inc in the include directory. The code is being executed whenever a new user connection to out application and a new session is started.

The file is optional. No default is provided.

SSeelleeccttiioonn ooff aauuttoommaattiiccaallllyy llooaaddeedd ccllaasssseess iinn prepend.php3
The file prepend.php3 determines which code is being loaded for all PHP3 interpreted pages. Normally, we include the class definitions for all core classes in this file: db_mysql.inc, session.inc, auth.inc, perm.inc, user.inc, then your local customizations from local.inc and the page management functions from page.inc.

You must change prepend.php3 to reflect the database interface that you are using: Change the require statement for db_mysql.inc appropriately.

If you are not using some core features from PHPLIB in your application or if you want some other features to be present on all your pages, you can delete or add require statements for their respective include files here.

The file is required. You must change it for your setup, unless you are using MySQL.

22...11...22... CCoorree ffuunnccttiioonnaalliittyy

The following files are included from prepend.php3 and provide

definitions for the core classes of PHPLIB. We recommend that you always include all of them, as they are a tightly integrated set of classes with many dependencies among them.

CCllaassss DB_Sql defined in exactly one of db_mysql.inc, db_msql.inc, db_pgsql.inc, db_odbc.inc, db_sybase.inc, db_mssql.inc, db_oracle.inc or db_oci8.inc:"

A database access class for your database server. PHPLIB depends on the presence of a SQL database server. Depending on the type of your database server, you have to select the appropriate include file. The file contains the definition of a class DB_Sql suitable for your database server.

The class manages a database connection (connection setup is implicit) and result memory is managed automatically.

An independent class.

CCllaasss Session defined in session.inc:

Manages an arbitrary amount of arbitrarily named session variables of scalar, array and object types (Object support requires that you implement two instance variables in your classes). Tracks sessions via cookies or a get-variable appended to each URL.

Depends on DB_Sql.

CCllaassss Auth defined in auth.inc:

Manages session authentication. Sessions are authenticated against usernames and passwords in a database. Authentication can be time limited.

Depends on Session and DB_Sql.

CCllaassss Perm defined in perm.inc:

Manages permission checks on authenticated session pages. Protected pages are only accessible to users with the specified rights.

Depends on Auth, Session and DB_Sql.

CCllaassss User defined in user.inc:

Manages user dependent variables. Unlike session variables these

are bound to a user id, not to a session id. They are persistent over multiple sessions, but are only available after a user has been authenticated.

Depends on Auth, Session and DB_Sql, extension of Session.

ffuunnccttiioonnss page_open() and page_close() defined in page.inc:" Setup and Shutdown functions, must be present on any session page.

Depend on Session.

22..11..33.. EExxtteennddeedd ffuunnccttiioonnaalliittyy

The extended functionality classes offer GUI-less background features that are commonly needed in HTML-applications. They may make use of core functionality (indicated for each class below).

Cart in cart.inc:

Manages a simple shopping cart. Items can be put into the cart, taken out of the cart and the carts contents can be enumerated.

Depends on Session to be useful. Requires that you add the statement require("cart.inc") to prepend.php3.

Template in template.inc:

Manages templates and variable replacement. Templates can be stored in files. They are loaded on demand and variables are replaced in these files.

An independent class. Requires that you add the statement require("template.inc") to prepend.php3 or that you include it manually on each page where you want to use it.

22..11..44.. HHTTMMLL wwiiddggeettss

HTML widgets are classes that generate some HTML-code (often forms or tables) to display GUI-elements. We try to provide functionality commonly used in applications, in a way that the actual look of the GUI-elements can be easily customized.

CSV_Table in csv_table.inc:

Creates a dump of a two dimensional array or a query result in CSV format, suitable for loading into a database or a spreadsheet program.

Depends on Table, extension of Table.

Sql_Query in sql_query.inc:

Create a selection widget that allows a user to choose arbitrary conditions on one or more table columns. SQL is being created from these selections that can be used in the where-clause of a larger SQL select statement.

Depends on Session and DB_Sql. Requires that you add the statement require("sqlquery.inc") to prepend.php3.

Table in table.inc:

Creates HTML tables from two dimensional arrays or from database query results. The class can either filter out the desired columns from an array or you can explicitly name which columns to show. A heading can be turned on if desired. All generated HTML elements are tagged with a classname you specify for stylesheet support, if needed. When used in a form tag, each table row can be prefixed with a checkbox input element to allow for row selection.

An independent class.

Form in oohforms.inc:

Creates HTML forms from feature->value arrays. This provides a single syntax for creating all of the different types of form elements. The class provides easy access to Javascript and server side validation, and supports 'freezing' some or all of the form elements to display static data. In addition, the library relies on object oriented implementations for the various form elements and these can easily be extended and customized.

An independent class.

22...22... DDoowwnnllooaaddiinngg aanndd uunnppaacckkiinngg tthhee ddiissttrriibbuuttiioonn

The base library is supplied at the PHP Base Library download location. Two different formats are provided: A tar.gz version and a shar version.

If you are on a windows system, you can use phplib.tar.gz, if you have WinZIP installed. Current versions of WinZIP know how to handle compressed tar archives. The uncompressed files may be installed on your windows system or transferred to your Unix system.

If you can't handle binary files, you may download phplib.shar. This is a pure ASCII file containing a self extracting shell script. Just save the file, make it executable and feed it to your Unix shell (for example, by typing sh phplib.shar).

The PHPLIB support mailing list is available should you run into problems with the library. To subscribe send the command subscribe to the mailing list subscription address.

22...33.. RReeqquuiirreemmeennttss aanndd tthhiinnggss ttoo cchheecckk ffoorr

22..33..11.. IInntteerrpprreetteerr rreegquuirreemmeennttss

The PHP base library requires a working web server with CGI capability and the CGI version of PHP 3.0.12 or higher installed. Alternatively mod_php can be used. Lower versions of PHP do not work at all: The session class uses the base64_encode() and base64_decode() functions which are known to be buggy in lower versions (up to 3.0.7) of the library. Also, the OOH Forms classes are using constructor syntax, which has been introduced into the PHP language in 3.0.5 and later versions. An issue with the \$PHP_SELF variable and CGI PHP has been resolved with version 3.0.5 and later. Perl regular expression functions are being used in the Template class and these are not really avilable up to 3.0.12.

_N_o_t_e_: If you are using CGI PHP, it _m_u_s_t have been compiled with the --enable-force-cgi-redirect switch for \$PHP_SELF to have the correct value.

Basically, if PHP_SELF is the exact local part of your \$URL, all is well. If it instead contains the modified URL with /your cgi-bin/php prefixed, you have a buggy version of CGI PHP. Either upgrade your version of PHP or replace all occurrences of \$PHP_SELF with \$PATH_INFO in PHPLIB.

_N_o_t_e_: PHPLIB requires that you have track_vars compiled in and enabled.

_N_o_t_e_: PHPLIB does not require short_open_tag to be enabled. The library always uses <?php as the PHP command introducer.

_N_o_t_e_: PHPLIB does not require magic_quotes_gpc to be enabled. The library always uses addslashes() when necessary.

22..33..22.. DDaattaabbaassee rreeqquuiirreemmeennttss

The PHP base library requires a database connection in the default setup for storage of session variables, but this can be circumvented by selection another storage container type at installation time. Currently, storage containers are available for SQL databases (the default), SQL databases with limited string length (ct_split_sql.inc), System V shared memory (requires a PHP interpreter with SYSVSHM and SYSVSEM support), LDAP servers (requires a PHP interpreter with LDAP support), flat files, and DBM files.

Using SQL, currently MySQL is fully supported and PostgreSQL, mSQL, Sybase, Microsoft SQL Server, ODBC and Oracle have limited support (the limitation is only relevant if you intend to access metadata information, i.e. table definitions and the like). Database interfaces are not difficult to write and you can easily write your own interface.

You need a database server connection with select, insert, update and delete privileges from your CGI environment. You need create and drop privileges from an administrative account outside your CGI environment as well.

PHPLIB core functionality requires two tables as part of your application table name space: active_sessions (select, insert, update and delete privilege required for the application user) and auth_user (select privilege required for the application user. insert, update and delete privilege required for the application user if user management is to be done from within the application).

Extended functionality may require additional tables.

22..33..33.. NNaammee ssppaaccee rreegquuiirreemmeennttss

PHPLIB tries to be as name space neutral as possible with its core features. Is issues no HTML by default and it occupies only few names in the global name space. These are the class names for the classes defined: DB_Sql, DB_SAM, CT_Sql, Session, Auth, Perm, User. Additionally, the classnames DB_Example, Example_CT_Sql, Example_Session, Example_Auth, Example_Challenge_Auth, Example_Perm and Example_User are defined by the sample setup in local.inc, but these names can and shall be customized by the application developer. PHPLIB defines the function names page_open(), page_close, sess_load() and sess_save() for the page management functions. The global variable \$_PHPLIB (a hash) is taken. Only if page_open() is being used, globals are defined by the library by default, but one global for each "feature" requested in the page_open() statement is taken. These are at most \$sess, \$user, \$auth and \$perm.

Including extension functionality or HTML widgets may occupy additional classnames, function names or variables in the global name space.

PHPLIB uses date fields within the column changed in the table active_sessions in your database. The changed field is used in garbage collection, that is, to clean out abandoned sessions. The date field is a 14 character field of the format YYYYMMDDhhmmss, that is, the date field has four digit years and will cope properly with the new millennium.

PHPLIB sets cookies in the client browser. These cookies by default have session lifetime, that is, they do not expire but are not written to disk. Date calculations are not involved.

It is possible to have PHPLIB set cookies with a limited lifetime by defining the \$lifetime slot of the Session class. If this is done, it depends on the date handling of the client browser and client operating system, if the result is Y2K compliant. There are known issues with longterm cookies and any browser on MS-DOS/Windows 3.11 systems.

PHPLIB does some date arithmetic internally that involves mktime() and date() functions of the PHP3 language and Unix time_t data types. The signed 32 bit Unix time_t data type counts seconds since 01-Jan-1970 Midnight GMT and will overflow sometime in the year 2038.

PHPLIB itself will function up to 2038 and longer, if the Unix time_t is being extended in time. PHPLIB does not protect you from date and Y2K issues in your PHPLIB application, the PHP3 interpreter, the server operating system oder server software, the client browser, the client operating system or other parts of your installation.

22..44.. IInnssttaallllaattiioonn pprroocceedduurree

mod_php note: The following instructions apply to the CGI version of PHP as well as to the module version. If you are working with mod_php, you must restart your web server to force a reload of the php3.ini file, though.

If you are using mod_php, you have additional configuration options: See the section below on using PHPLIB mit mod_php.

LLiibbrraarryy SSeettuupp Create a directory php next to your cgi:

/home/www/servers/phplib.netuse.de/pages <- document root
cgi <- php binary

php <- includes and prepends

Make this php directory your php include directory: Put include_path = /home/www/servers/phplib.netuse.de/php into cgi/php3.ini. If you already have an include path defined in your setup, add the PHPLIB include path to the existing include path using the separator character applicable for your operating system (":" on UNIX, ";" on Windows). Defining an include path will not actually include code on your pages; it only tells the PHP interpreter in which directories to look for files referenced in require() and include() statements.

Into the php directory go all the files from the php directory of the distribution. Into documentroot, put all the files and directories from the pages directory of the distribution.

Have documentation.txt handy and read it.

DDaattaabbaassee aacccceessss wwiitthh MMyySSQQLL
The following information applies to MySQL only. No information is provided for other database servers as of now. You are encouraged to copy this section, adapt it for your database server and submit it to the authors. It will be included in further releases of PHPLIB.

Edit prepend.php3. Change the first require() statement to require("db_mysql.inc");. This will include the MySQL database interface (Interfaces for other databases are provided in db_<databasename>.inc. The require() statement has to be adapted to reflect this).

Assuming your database server is named database.netuse.de and your CGI user is webuser and you are accessing the database myapp, do

mysql -h database -u webuser myapp

If it does not work, connect your database as administrator and create the proper mysql access permissions. Adapt and run create_database.mysql from the stuff subdirectory of the distribution to create the databases active_sessions and auth_user as well as the sample user kris with password test. Try again to connect like shown above. Can you do select * from active_sessions? and insert into active_sessions values ("1", "2", "3", "") as well as delete from active_sessions? Can you select * from auth_user?

_N_o_t_e_: Additional database creation scripts are provided for several different databases in the stuff directory of the distribution.

MMeerrggiinngg tthhee lliibbrraarryy wwiitthh yyoouurr PPHHPP ffiilleess Decide if you want to use include or auto_prepend_file. We do use auto_prepend_file here and we add the statement auto_prepend_file = /home/www/servers/phplib.netuse.de/php/prepend.php3 to our php3.ini.

Not all classes are included/required by prepend.php3, only core functionality files are: db_xxx.inc, ct_sql.inc, session.inc, auth.inc, perm.inc, user.inc, local.inc and page.inc. The library provides other, less essential classes that can be included manually on a page-by-page basis. Some classes make themselves persistent, if used. These classes require that you include their definitions in the prepend.php3 file where indicated to function correctly. See the usage instructions for these classes for details.

Having done this, access /index.php3. The counter should increment when that page is being reloaded. Also, checking active_sessions in the database should reflect that session.

SSuubbssccrriibbee ffoorr ssuuppppoorrtt
Subscribe to the mailing list phplib@lists.netuse.de. Do so by sending a mail body of subscribe to phplib-request@lists.netuse.de and follow instructions. Share your experiences.

22..55.. UUssiinngg iinncclluuddee(()) iinnsstteeaadd ooff aauuttoo__pprreeppeenndd__ffiillee==

If you do not want to use auto_prepend_file to load the PHPLIB core functionality, you can load the class definitions for the core manually on each page that requires them.

You will have to define a valid include_path=-statement in your php3.ini file as outlined previously to reflect the location of the *.inc files. Then, all core functionality can be loaded with include("prepend.php3") as the first statement at the top of each page.

To further optimize performance, you can minimize the contents of the prepend file, if you do not need all core functionality. You _m_a_y leave out auth.inc, perm.inc and user.inc, if you do not require these features (note that there are dependencies among these classes!).

22..66.. PPHHPPLLIIBB wwiitthh mmoodd__pphhpp ((AAppaacchhee mmoodduullee))

Installing PHPLIB onto a web server that has PHP3 as a module (actually Apache) mainly differs in where you can set up runtime settings for PHP3 itself. PHP3 can be compiled with a wealth of parameters (see the PHP section in phpinfo()), most of which can get overridden by the php3.ini file. The location of this file is shows as part of the output of phpinfo().

With PHP3 as a module you have a wider choice on placing these settings: they are overridden, in this order, by what is defined in httpd.conf and in your per-directory .htaccess file. Directives in these files are identical to their php3.ini brothers, but are prefixed with php_ to avoid clashes with Apache configuration keywords. Also, as they are Apache configuration keywords, they have no equals ("=") sign in them. If x=y is a configuration directive from php3.ini, you should be using php3_x y within the Apache configuration instead. That is, you should prepend php3_ to the keyword and omit the equals sign. If you misspell a configuration directive, you will get an error 500 from your webserver and find more details about the error in the logfile you configured with ErrorLog in your webserver setup.

_E_x_a_m_p_l_e_: If below we talk about setting in your php3.ini the configuration

include_path = "/bla"

mod_php users may alternatively configure in their httpd.conf the following:

<Directory /home/www/servers/phplib.netuse.de/pages>
php3_include_path "/bla"
</Directory>

Of special interest to PHPLIB users are the following directives:

......

; Data Handling ;

```
magic quotes gpc = Off; magic quotes for incoming
               ; GET/POST/Cookie data
magic quotes runtime = Off; magic quotes for runtime-generated data,
               ; e.g. data from SQL, from exec(), etc.
magic quotes sybase = Off; Use Sybase-style magic quotes
               ; (escape 'with "instead of \')
                     ; enable $PHP_GET_VARS[], $PHP_POST_VARS[]
track vars = On
               ; and $PHP_COOKIE_VARS[] arrays
; automatically add files before or after any PHP 3.0 document
auto_prepend_file = (add path to prepend.php3 here)
auto append file =
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
; Paths and Directories ;
include_path = (add path to the directory with all .inc files)
```

All of this comes very handy when you have multiple virtual hosts (e.g. you are an ISP). In this case you can comfortably place the php3 directives in the <VirtualHost> block or in an .htaccess file in the client directory.

33.. CCoorree FFuunnccttiioonnaalliittyy

Each class contains instance variables and instance methods. Some of these variables and methods are available for customization, some are internal to the classes themselves. All are documented, but tampering with internal variables and methods is not supported. Internal interfaces are subject to change without notice from one version of the library to another.

This section covers PHPLIB core functionality in reference form. Classes are presented in order of dependency, though, because the core structure is easier understood in this order. You will need to understand the complete core structure to successfully use all of PHPLIB's features.

33..11.. DDBB__SSqqll

DB_Sql is used by CT_Sql and Auth to access a SQL database. You are encouraged to use it directly, too.

33..11..11.. IInnssttaannccee vvaarriiaabblleess

Accessible instance variables.

Internal instance variables.

33..11..22.. IInnssttaannccee mmeetthhooddss

33..11..22..11.. AAcccceessssiibblee iinnssttaannccee mmeetthhooddss

DDBB__SSqqll((\$\$qquueerryy == Constructor. When creating an instance, you may optionally supply a query string.

\$db = new DB_Sql_Subclass("select * from mytable");)

qquueerryy((\$\$qquueerryy__ssttrriinngg))

query_string is a SQL statement that is sent to the database. After sending the statement, Error and Errno are updated. If the query is syntactically incorrect (no valid result id is being produced), halt() is called with a meaningful error message.

If there is no active link to the database, a pconnect() is made using the information from the Host, Database, User and Password instance variables.

Returns the result of the query() statement, which is guaranteed to be a valid result id (or false, if Halt_On_Error isn't "yes").

nneexxtt__rreeccoorrdd(())

next_record() advances the cursor through the current query result and updates the Record, Row, Errno and Error instance variables.

Returns true, if there is a new result record. Returns false, if done with the current result set. If Auto_Free is true, free_result() is called automatically before false is returned.

nnuumm__rroowwss(()),, nnff(())

Returns the number of rows returned by the current SELECT query.

_N_o_t_e_: This information is not available in all database interfaces. Some of the more advanced databases begin to return query results asynchronously while the backend is still appending result rows. In such environments the complete size of the result set is never known.

You should duplicate your WHERE clause of the query in such environments and ask for the COUNT(*). This will be less inefficient as it seems as the query path and query result have been cached by the database.

aaffffeecctteedd__rroowwss(())

Returns the number of rows affected by the current INSERT, UPDATE or DELETE query.

nnuumm__ffiieellddss(())

Returns the number of columns returned by the current query.

nnpp(())

Prints the number of rows returned by the current query.

ff((\$\$ffiieelldd))

Identical to accessing Record[\$field].

pp((\$\$ffiieelldd))

Identical to printing Record[\$field].

hhaallttmmssgg((\$\$mmssgg))

This function is called by halt() and will actually print the database error message. You may override this method in your subclass of DB_Sql and format the error message to be consistent with the layout of the rest of your application. You may also add additional error handling such as informing the application operator by mail that a database error has occured.

sseeeekk((\$\$ppooss))

Positions the Row pointer within the result set. Useful for reading the same result set twice or otherwise jumping around

within the result. \$pos is not checked in any way for validity.

_N_o_t_e_: If Auto_Free is true, seek() may not be useable, because the result set has already been free'ed when next_record() when behind the last record of the result set.

_N_o_t_e_: Not all database interfaces provide a cursor that is capable of seeking. This function will be unavailable in such environments.

lliinnkk__iidd(())

This function will return the current link ID, as returned by the pconnect() executed internally by the database class.

You should not need this information.

qquueerryy__iidd(())

This function will return the current result ID, as returned by the query() executed internally by the database class.

You should not need this information.

mmeettaaddaattaa((\$\$ttaabbllee ==

\$table is a SQL table name in the current database. The function returns an array of hashes indexed on the (0 based) column number of \$table. Each hash is indexed by table (table of which this column is part of), name (name of this column), type (column data type), len (column width) and flags (database specific column flags, if applicable) with one row per table column. Each row describes a column in your table.

The data returned by metadata() is suitable for passing it to the Table class. If you specify the full parameter, an additional column meta is added, which is indexed by field name and returns the field number of that name. Also, a column num_fields is added, containing the width of the table.

If \$table is omitted, the function returns metadata on the result of the last executed query. _N_o_t_e_: This is currently implemented only for the MySQL interface. You are encouraged to implement this feature for other interfaces.

_N_O_T_E_: At the moment, the PostgreSQL and ODBC interface only report the table, name and type data reliably. You are encouraged to fix this.

ttaabbllee nnaammeess(())

Returns an array with table name and tablespace name.

table name : \$return[\$i]["table_name"]

tablespace_name : \$return[\$i]["tablespace_name"]

Tables are from \$i=0 to last table;

Implemented in db_oracle.inc,db_oci8.inc,db_mysql.inc,db_pgsql.inc

nneexxttiidd((\$\$sseeqquueennccee__nnaammee))

This function will return a sequence number from the sequence named by \$sequence_name. This number is guaranteed to be obtained in an atomic manner and can be used as a primary key.

33..11..22..22.. IInntteerrnnaall iinnssttaannccee mmeetthhooddss

```
ccoonnnneecctt(())
```

Used internally to generate a Link_ID, if necessary. Link creation is implicit, there is no need to call connect() manually, ever.

hhaalltt((\$\$mmssgg))

Used by query() if the initial database connection cannot be made or the target database does not exist. Depending on the setting of Halt_On_Error, this method will call haltmsg() to report the error.

ffrreeee(())

Used internally by next_record() to free the result set, if so configured.

33..11..33.. EExxaammppllee

Use a subclass to provide the appropriate parameters for a database connect. You may overwrite halt() to customize the error message, although a sensible default is provided.

class DB_Article extends DB_Sql {
 var \$classname = "DB_Article";

```
var $Host = "sales.doma.in";
var $Database = "shop_project";
var $User = "webuser";
var $Password = "";

function haltmsg($msg) {
  printf("<b>Database error:</b> %s<br/>", $msg);
  printf("<b>MySQL Error</b>: %s (%s)<br/>",
  $this->Errno, $this->Error);
  printf("Please contact shopmaster@doma.in and report the ");
  printf("exact error message.<br/>");
}
```

Use an instance of the subclass to manage your queries:

```
$\frac{1}{\sq} = \text{new DB_Article;}$$
$\quad = \text{sprintf("select * from articles where article like '%%%s%%'", $\frac{1}{\sept{searchword};}$$
$\quad = \text{sprintf("select * from articles where article like '%%%s%%"', $\frac{1}{\sept{searchword};}$$
$\quad = \text{sprintf("select * from articles where article like '%%%s%%"', $\frac{1}{\sept{searchword};}$$
$\quad \quad \q
```

33..11..44.. AAddddiittiioonnaall iinnffoorrmmaattiioonn aabboouutt ddaattaabbaassee ccoonnnneeccttiioonnss

PHP reuses connections, if possible. When a connection is being made

to the same Host with the same Username and Password as an existing connection, no second connection is being made by PHP. Instead the existing connection is returned to the caller. This is true for both, the *_connect() and *_pconnect() calls of all PHP database interfaces.

This has implications for MySQL users: Never use the MySQL "use" command to change the current database. If you do, session management will fail to operate properly. Instead, create all PHPLIB tables as part of your application.

Some databases (for example Oracle) have very expensive connect() operations. For these databases, performance is dramatically improved if you switch from CGI PHP to mod_php. This is, because PHPLIB uses the "*_pconnect()" method to connect to your database. In mod_php, the database connection is kept around by the web server process after the page has been processed and is reused if a further connect requires a connection with the same Host/Username/Password pattern.

This means that there will be at most "number of web server processes" times "number of Host/Username/Password-combinations" many simultaneous connections to your database server. Keep that in mind when planning licenses and server load. Using CGI PHP will probably reduce the number of concurrent connects to your database server at the expense of connection setup time. For database servers where connection setup time is negligible (MySQL for example) this is a viable solution (don't try it with Oracle) though.

33..11..55.. UUssiinngg nneexxttiidd(())

The nextid() function can be used to obtain a sequence number which can be used as a primary key. The function manages an arbitrary number of named sequences, you have to provide the name of a sequence upon call.

```
$\frac{1}{3} \text{db} = \text{new DB_Article;}$

$\text{artnr} = \text{$db->nextid("article_sequence");}$

$\text{query} = \text{sprintf("insert into articles ( artnr, ...) values ('%s', ...)",}$

$\text{artnr, ...);}$

$\text{$db->query(\text{$query});}$

$\text{reset(\text{$articles});}$

$\text{while(list(\text{$itemnr, \text{$itemdesc})} = each(\text{$articles})) {}}$
```

```
$itemnr = $db->nextid("item_sequence");
$query = sprintf("insert into items (artnr, itemnr, ...) values ('%s', '%s', ...)",
$artnr, $itemnr, ...);
$db->query($query);
}
```

33..22.. PPaaggee MMaannaaggeemmeenntt

33..22..11.. AAccceesssiibblee FFuunnccttiioonnss

Page Management currently consists a collection of functions:

```
ppaaggee__ooppeenn((aarrrraayy((
    This function is to be called with an array of page features/classname pairs. Valid features are at the moment:
```

sseessss

This page makes use of session variables.

aauutthh

This page uses session authentication. If you specify the auth feature, you MUST specify the sess feature, also.

ppeerrmm

This page is protected by permissions and only accessible to authenticated users with matching rights. If you specify the perm feature, you MUST specify the auth and sess features, also.

uusseerr

This page makes use of user variables. If you specify the user feature, you MUST specify the auth and sess features, also.

Each feature specifies the name of the class that implements that feature, for example

page_open(array("sess" => "Shop_Session"));

The function creates an instance of Shop_Session as \$sess and initializes it. It also checks feature dependencies. Note that you are expected to provide an implementation of the class Shop_Session. This is usually done in local.inc and usually you do so by extending the provided Session class.

Examples on how to do this is given in the documentation below when the classes are introduced.

```
ppaaggee__cclloossee(())
```

At the end of your page (after all results have been calculated) you have to call page_close(). This will save all page state, session and user variables into database. Changes to session or user variables after page_close() has been called are not recorded. Currently it is allowed to call page_close() multiple times on a single page (not guaranteed for future versions!). Each time session state will be saved.

_N_o_t_e_: This is going to change. When we introduce record locking, it is important that you call page_close() only once per page, because that will implicitly unlock your session record. Also, it is important that you call page_close() as early as possible on a page so that the locking time is kept minimal.

```
sseessss llooaadd((aarrrraayy((
```

_A_d_v_a_n_c_e_d _f_e_a_t_u_r_e. Some applications have need to manually load data belonging to one or multiple session classes. @@TODO

```
sseessss_saavvee((aarrrraayy((
_A_d_v_a_n_c_e_d_f_e_a_t_u_r_e. @@TODO
```

33..22..22.. EExxaammppllee

```
<?php
page_open(array("sess" => "Shop_Session"));
$sess->register("s"); // See "Session" below for explanation.
?>
<html>
<h1><?php print ++$s ?></h1>
</html>
```

33..22..33.. TThhee ""ccaarrtt"" ffeeaattuurree iiss ggoonnee

There used to be a feature "cart" for page_open() in versions of PHPLIB up to release-5. The cart has been removed from the core functionality of PHPLIB to keep the library small, maintainable and structured. Consequently the "cart" feature is gone. The Cart class is still present and exists as an extended feature. You have to include and instantiate your cart manually on that pages that use it, though. See the Cart class for more information.

33..33.. CCTT__SSqqll

The Session class used to contain a bit of SQL to read and write session data from and to a database. To make sessions database independent, this SQL has been isolated and put in a separate class, CT_Sql. Session now makes all storage accesses through a container class, which may or may not be an SQL container.

33..33..11.. IInnssttaannccee vvaarriiaabblleess

Accessible instance variables.

33..33..22.. EExxaammppllee

Use a subclass to provide the appropriate parameters to your container. Usually your subclass looks like this:

```
class My_Sql extends CT_Sql {
    var $classname = "My_Sql";
    var $database_table = "active_sessions";
    var $database_class = "DB_Session";
}
```

You can then use My_Sql in class Session. Reference it by putting "My_Sql" in the "that_class" variable.

33..44.. CCTT__SSpplliitt__SSqqll

The Session class used to contain a bit of SQL to read and write session data from and to a database. To make sessions database independent, Session now makes all storage accesses through a container class. The CT_split_sql container is very similar to CT_Sql container, with the difference that if serialized data exceeds a specified amount of bytes, multiple rows will be used to memorized the entire field.

This class is NOT compatible with CT_Sql class, since table layout is different and column names are different in order to avoid reserved words in various database implementation. This uses a DB_Sql like class so you can access all supported databases with this container.

33..44..11.. IInnssttaanncee vvaarriiaabblleess

Accessible instance variables.

33..44..22.. EExxaammppllee

Use a subclass to provide the appropriate parameters to your container. Usually your subclass looks like this:

```
class My_Sql extends CT_Split_Sql {
    var $classname = "My_Sql";
    var $database_table = "active_sessions_split";
    var $database_class = "DB_Session";
    var $split_length = 4096;
}
```

You can then use My_Sql in class Session. Reference it by putting "My_Sql" in the "that_class" variable.

33..55.. CCTT__SShhmm

The Session class used to contain a bit of SQL to read and write session data from and to a database. To make sessions database independent, Session now makes all storage accesses through a container class. To let Session use shared memory as container, you use CT_Shm.

33..55..11.. IInnssttaannccee vvaarriiaabblleess

Accessible instance variables.

33..55..22.. EExxaammppllee

Use a subclass to provide the appropriate parameters to your container. Usually your subclass looks like this:

```
class My_Shm extends CT_Shm {
    var $classname = "My_Shm";
    var $max_sessions = 500;
    var $shm_key = 0x1234232;
    var $shm_size = 64000;
}
```

You can then use My_Shm in class Session. Reference it by putting "My_Shm" in the "that_class" variable.

33..66.. CCTT__DDbbmm

The Session class used to contain a bit of SQL to read and write session data from and to a database. To make sessions database independent, Session now makes all storage accesses through a container class. To let Session use a DBM database file as a container, you use CT_Dbm.

33..66..11.. IInnssttaannccee vvaarriiaabblleess

Accessible instance variables.

33..66..22.. EExxaammppllee

Use a subclass to provide the appropriate parameters to your container. Usually your subclass looks like this:

```
class My_Dbm extends CT_Dbm {
    var $dbm_file = "data/session.dbm";
}
```

You can then use My_Dbm in class Session. Reference it by putting "My_Dbm" in the "that class" variable.

33..77.. CCTT__LLddaapp

The Session class used to contain a bit of SQL to read and write session data from and to a database. To make sessions database independent, Session now makes all storage accesses through a container class. To let Session use a LDAP database as a container, you use CT_Ldap.

33..77..11.. IInnssttaannccee vvaarriiaabblleess

Accessible instance variables.

33..77..22.. EExxaammppllee

Use a subclass to provide the appropriate parameters to your container. Usually your subclass looks like this:

```
class My_Ldap extends CT_Ldap {
    var $classname = "My_Ldap";
    var $ldap_host = "localhost";
    var $ldap_port = 389;
    var $basedn = "dc=your-domain, dc=com";
    var $rootdn = "cn=root, dc=your-domain, dc=com";
    var $rootpw = "secret";
    var $objclass = "phplibdata";
}
```

You can then use My_Ldap in class Session. Reference it by putting "My_Ldap" in the "that_class" variable.

33..88.. SSeesssiioonn

The session class keeps a list of global variable names and provides a set of functions to load and save these variables from and to a data storage container (we will call it container for shortness). The named variables may be scalar variables (strings, integers and floats) or arrays. Objects are handled as well, provided they implement two instance variables naming their class and enumerating their (persistent) slots.

33..88..11.. IInnssttaannccee vvaarriiaabblleess

Accessible instance variables.

Internal instance variables.

33..88..22.. IInnssttaannccee mmeetthhooddss

33..88..22..11.. AAcccceessssiibblee iinnssttaannccee mmeetthhooddss

rreeggiisstteerr((\$\$vvaarrnnaammee))

Registers a global variable name as a session variable. The name may identify a scalar variable, an array or an object. If an object is to be made persistent, it must have two instance variables:

ccllaassssnnaammee

A string with the name of the objects class.

ppeerrssiisstteenntt_sslloottss

An array with the names of all object slots to save.

uunnrreeggiisstteerr((\$\$vvaarrnnaammee))

Unregisters a global variable name as a session variable. The variable is not deleted, but its value will be lost at the end of a page. It is no longer saved to the database.

iiss__rreeggiisstteerreedd((\$\$vvaarrnnaammee))

Returns true if the variable named \$varname is registered with the session, false otherwise.

ddeelleettee(())

Destroy the current session and put_id() the current session id.

After delete() has been executed, all session data has been removed from the database. Also, the session object is unusable on this page. Consequently, page_close() may not be called for this session. Session variables are still available on this page, even after the delete(), but will be lost on the following pages.

In cookie mode, it is possible to page_open() a new session after delete() has been called, if no HTML has been output so far so that the new cookie can be set. If you do this, you can

also re-register some of the previous session variables and can call page_close() for the new session. This allows you to change the session on the fly and selectively carry over session data from the previous session.

```
uurrll(($$uurrll))
 Return an URL referencing the current session. If in get mode,
 the current session id is attached to this URL, else the URL is
 returned unmodified.
ppuurrll(($$uurrll))
 A shorthand for print $this->url($url);
sseellff uurrll(())
 Return an URL referencing the current page, including PHP_SELF
 and QUERY STRING information. If in get mode, the session id is
 included.
ppsseellff uurrll(())
 A shorthand for print $this->self_url().
hhiidddeenn__sseesssiioonn(())
 Adds a hidden form element containing the session name and id.
aadddd__qquueerryy(($$qqaarrrraayy))
 Return string to be appended to the current URL for parameters
 in GET query format. Intended usage is like this:
  <a href="<<?
  $sess->pself_url().$sess->padd_query(array("again"=>"yes"))
  ?>"> Reload</a> and log in?
ppaadddd__qquueerryy(($$qqaarrrraayy))
 A shorthand for print $this-> add_query($qarray).
```

 $file: ///C / Users/stevensc/Desktop/Goldsmith/code/thesis_code/comptracker/phplib-7.2 d/doc/documentation.txt [26/11/2014~1:15:13~PM] \\$

When a FORM variable is made persistent, that form variable is

rreeiimmppoorrtt__ggeett__vvaarrss(())

imported into PHP, then page_open() is being called and the new variable value is overwritten from the database. The FORM value is lost.

If you had enabled track_vars and were accessing HTTP_GET_VARS directly, which is recommended, this were not a problem. Some legacy scripts rely on persistent FORM input variables, though.

These scripts may call the appropriate reimport_x_vars() functions. These functions will re-read the tracked variable arrays and reinitialize the appropriate global variables after session variables have been restored.

Use of this function is discouraged.

```
rreeiimmppoorrtt_ppoosstt_vvaarrss(())
See reimport_get_vars().
```

```
rreeiimmppoorrtt__ccooookkiiee__vvaarrss(())
See reimport_get_vars().
```

sseett__ccoonnttaaiinneerr(())

You shall not call this function directly. It is called back by the start() function of Session() during initialization. It is documented so that you can override its implementation in your subclass of Session if you know what you are doing. This function creates and starts the container class used by this instance of session.

sseett_ttookkeennnnaammee(())

You shall not call this function directly. It is called back by the start() function of Session() during initialization. It is documented so that you can override its implementation in your subclass of Session if you know what you are doing.

This function determines and sets the internal session name.

rreelleeaassee ttookkeenn(())

You shall not call this function directly. It is called back by the start() function of Session() during initialization. It is documented so that you can override its implementation in your subclass of Session if you know what you are doing.

This function determines the current method of session propagation and determines if a new session token has to be generated.

ppuutt__hheeaaddeerrss(())

You shall not call this function directly. It is called back by the start() function of Session() during initialization. It is documented so that you can override its implementation in your subclass of Session if you know what you are doing.

This function determines which header lines are to be generated by the session, including cache control headers.

33..88..22..22.. IInntteerrnnaall iinnssttaannccee mmeetthhooddss

```
ggeett__iidd(())
See get_id().
```

ggeett__iidd((\$\$iidd__ttoo__uussee))

get_id() is used internally to determine a session identifier. Currently, a session identifier is a hex number of 32 characters (128 bits) and it is generated by md5(uniqid(\$this->magic)) to make it hard to guess.

get_id() may be called with an optional session id to use as a parameter. This is useful if you want to change a session id without breaking the session (taking over an old, left over session).

get_id() can be overwritten by a subclass, if you want a different system to create session ids. For example, some applications want to use a constant session id that is not propagated to the client to use a shared pool of persistent variables (a guestbook for example). These applications need locking (to be implemented soon).

ppuutt__iidd(())

put_id() is used internally to "unuse" a session it. At the
moment it deletes the client side cookie and deletes
\$HTTP_COOKIE_VAR[\$this->name] for that cookie. The variable
\${\$this->name} is _n_o_t deleted.

sseerriiaalliizzee((\$\$pprreeffiixx,, &&\$\$ssttrr)) serialize() is used internally to append to str all PHP code needed to reconstruct the variable named in prefix.

ffrreeeezzee(())

freeze() serializes all register()ed variables and writes the resulting code into the database, tagged with the current session id and the current session name.

tthhaaww(())

thaw() loads a set of freeze()ed variables for the current session id and session name out of the database and recreates them.

ggcc(())

The active_sessions table contains one row for each session. That row is uniquely identified by the sid and name values (name is the name of the session class that has written the row). Each time that row is written, the column changed is updated with the current time.

The gc() function deletes all rows that are older than gc_time minutes and have a matching name field. For speed reasons, gc() is not not called every time an update to active_sessions is being made. Instead it is called randomly with a probability of gc_probability.

rreeiimmppoorrtt__aannyy__vvaarrss((\$\$aarrrraayynnaammee)) Used to implement the three official reimport functions.

ssttaarrtt(())

Initialization function, to be called after object instantiation. Calls get_id() to get the current session id, creates a database connection, then calls thaw() to load all session variables. Randomly activates gc(). Checks allowcache to send proper headers to control browser caching.

33..88..33.. EExxaammppllee

Use a subclass to provide the appropriate parameters to your session. Usually your subclass looks like this:

```
class My_Session extends Session {
  var $classname = "My_Session"; ## Persistence support

  var $mode = "cookie";
  var $lifetime = 0; ## use session cookies

  ## which container to use
  var $that_class = "Session_sql";
}
```

Remember that you have to provide a DB_Sql subclass with the parameters needed to access your database.

Use the page management functions (see above) to use your session subclass. The feature name for session management is sess; provide the name of your session subclass as a parameter to the sess feature:

```
page_open(array("sess" => "My_Session"));
```

Use the register() instance method to register variables as persistent. If \$sess is your session object, use

```
$sess->register("s");
```

to make the global variable \$s persistent. \$s may be a scalar value, an array or an object with persistence support slots.

Do not use the instance methods freeze() and thaw() directly, but use the page management functions instead.

To have some pages cached and others not cached, use multiple instances of the session object. For example, for those pages that should be cached, use a session object instance like

```
class My_Cached_Session extends My_Session {

## pages that use this session instance are cached.

var $allowcache = "private";
}
```

Be careful when using the public cache option. Publically cached pages may be accessible to unauthenticated users. The private cache option

prevents unauthenticated access, but is only functional in HTTP/1.1 browsers.

```
33..88..44.. UUssiinngg ""aauuttoo__iinniitt""
```

You may define \$sess->auto_init to the name of an include file in your extension of session. Per convention, the name setup.inc is being used.

```
class My_Session extends Session {
  var $classname = "My_Session";
  var $magic = "Calvin+Hobbes";
  var $mode = "cookie";
  var $gc_probability = 5;

  var $auto_init = "setup.inc"; // name of auto_init file.
}
```

Whenever a new session is established, that is, a user without a session id connects to your application, the auto_init file is included and executed exactly once. The file is executed from within the context of the page_open() function, that is, _n_o_t within a global context. To define or access global variables from the auto_init file, you have to global them.

When auto_init is being executed, all features of your page already exist and are available globally. That is, you can safely rely on the existence of the \$sess, \$auth, \$perm and \$user variables, if your application specifies them. _N_o_t_e that you cannot in general know which particular page triggered the execution of auto_init, though. If you have some pages that request authentication and others that don't, you cannot rely on the presence of the \$auth object in general, but have to test for it with is_object(\$auth) before accessing it.

The auto_init file is the appropriate place to initialize and register all your session variables. A sample setup.inc may look like this:

```
<?php
global $lang; // application language
$lang = "de"; // german by default
$sess->register("lang");
```

```
global $cur; // application currency

$cur = "EUR"; // Euro by default

$sess->register("cur");

global $cart;

$cart = new Shop_Cart; // Create a shopping cart object as defined in local.inc

$sess->register("cart"); // register it.

?>
```

_N_o_t_e_: If you don't use a fallback_mode and you get users that turn off cookies, these users will force a new session each time they hit any page of your application. Of course this will force inclusion and execution of setup.inc for each page they visit, too. Nothing can be done about this.

33..88..55.. UUnnrreeggiisstteerriinngg vvaarriiaabblleess aanndd ddeelleettiinngg sseesssiioonnss

To get rid of a persistent variable, call \$sess->unregister() with the name of that variable. The value of the formerly registered variable is still available after the call to unregister, but the variable is no longer persistent and will be lost at the end of the current page.

To get rid of all session related data including the session record in the database, the current session id and the session cookie in the users browser, call \$sess->delete(). In shopping applications this is commonly done when the user commits his order to get rid of the current shopping cart and everything else. You may want to remember selected information about that user, though, as shown below.

```
<?php
page_open(array("sess" => "Shop_Session"));

// send order as mail
mail_order($shopowner, $user, $cart);

// delete the current session
$sess->delete();

// now get a new session id, but retain the users
// address and name:
page_open(array("sess" => "Shop_Session")); // will force auto_init again!
$sess->register("user"); // could be done in auto_init as well
?>
```

33..88..66.. RReeaaddiinngg aanndd uunnddeerrssttaannddiinngg sseessssiioonn ddaattaa ffoorr ddeebbuuggggiinngg

When debugging PHPLIB applications, it is often useful to be able to read and understand the contents of the active_sessions table. Each session is represented by a single line in this table. The primary key to this table is the pair name and sid. name is the content of \$this->name and is usually the classname of your session class. sid is the content of \$this->id and is usually the MD5 hash of a uniqid and some magic string.

By choosing a pair, it is possible for PHPLIB to have more than one session type (for example, session and user data, see the User class below) per application and store all this data in a single table. If you are debugging a session class, for example Example_Session, only records where name = "Example_Session" are of interest to you. Determine the current session id of your Example_Session by printing \$sess->id and select the record with that name and sid from the database.

The changed field indicates when this record has been updated the last time. It is a 14 character (Y2K compliant) string of the format YYYYMMDDhhmmss. Ordering by changed desc will show you the most current session records first (the MySQL "limit" clause may come in handy here).

The val column of a session record contains a PHP program that can be safely fed to stripslashes() first and eval() after that. The PHP program consists entirely of assignments and contains all instructions necessary to recreate the persistent variables. The structure and order of instructions within this program is always the same.

First item is always an assignment to \$this->in. If set to 1, auto_init has been executed by this session. If _n_o_t set to 1, auto_init has not been executed, yet. This may be because no auto init file is defined for that session.

After that comes code like this: \$this->pt = array(); followed by a bunch of assignments like \$this->pt["somestring"] = 1;. Each somestring is the name of a registered variable. Variable registrations are persistent themselves and are saved with the \$this->pt array. Even if the variable in question is not set, it may be registered and stays so until it is unregistered or the session is deleted. Check the contents of the pt array is you want to see which variables are currently registered with your session.

Finally, the actual contents of your variables are saved. This is always done by accessing the \$GLOBALS array and always by enumerating the scalar values that make up the persistent variable. For a scalar, you will see code like \$GLOBALS[somevar] = "value";.

For an array, first \$GLOBALS[someary] = array(); is generated. Then the scalars that make up the array, if any, are written out, generating code that looks like \$GLOBALS[someary][index] = "value".

And for objects, code to create an object instance is saved: \$GLOBALS[someobj] = new Classname;. "Classname" is taken from the objects \$classname slot, which _m_u_s_t be present and accurate. Then the scalars that are to be saved are written out, according to the contents of the objects persistent_slots array: \$GLOBALS[someobj]->slot = "value"; is written.

If you want to see what values have been saved to the database, you just have to look at the \$GLOBALS assignments for that session.

33..88..77.. HHooww ""sseerriiaalliizzee(())"" ooppeerraatteess

The following information is applicable only to library developers, that is, programmers that want to change the internal workings of PHPLIB. You may safely skip this section; some information here requires advanced understanding of the PHP language.

The heart of the session class is the serialize() internal function. This function takes an expression called prefix and generates PHP code that will assign the value of that expression to the expression when executed. For example, if the expression is \$GLOBALS["a"] and the global variable \$a has the value 17, then serialize will create the PHP program \$GLOBALS["a"] = "17";. To save memory, serialize() operates on a reference parameter \$str, where is will append the code generated.

First thing serialize() does is to determine the type of the current expression using the PHP gettype() function. The current type is stored in \$t. The type of the expression may indicate either a scalar value (integer number, float number or string), an array or an object.

Scalar values are the easiest to handle: serialize() just evaluates the current expression and remembers the result value in \$1. An assignment is generated that will assign the current value to the current expression. Since the current value may be a string and that string may contain bad characters (any of backslash, double quotes or dollar sign), these characters are backslashed. We are done, serialize() ends here for scalars.

In the case of \$t indicating an array, code is generated to create an empty array (expression = array();). Then the keys of current expression are enumerated and for each key serialize() is called recursively with the current key appended to the expression. That will append code for each array slot.

Should \$t indicate an object, code is generated to create that object (expression = new Classname;). Since one cannot find out the name of

the class of an object for arbitrary objects in PHP, objects handled by serialize() must have a slot named classname. The object handler will then enumerate the contents of the objects slot persistent_slots and call serialize() recursively for each of these slots with the appropriate prefix.

Since many of the expressions used in serialize() require variable variable names or even variable code, eval() is used liberally. Unfortunately, this makes the code hard to read.

33..99.. AAuutthh

Authentication management can be used to authenticate a session, that is, to identify the user at the client side of the session.

Authentication is done inline, with HTML forms, _n_o_t with HTTP authentication (that's the browser popup you get when you hit a page protected with htaccess). Inline authentication has several advantages over HTTP authentication:

- · It can be undone: A session can be un-authenticated, the user can "log out".
- · It can expire: A session can automatically be un-authenticated after a given idle time.
- · It can be customized: You are not limited to user/password pairs. Instead you could use a customer number, operator id and a password to log in. Also, you have full control over the login screen, which is a normal HTML page with logos, help and forms as you see fit.
- · It is database based. Authentication is being done against a database of your design, not a htpasswd text file.
- · It is per page. You decide on a per-page basis which pages are authenticated and which aren't.
- · It can be user authenticating and optionally self registering. In _r_e_g_i_s_t_r_a_t_i_o_n mode, a user without a valid login is encouraged to register and an account is created for this user.
- · It works with CGI PHP. HTTP authentication is available only in mod_php.
- · It is integrated with a permission checking scheme.

33..99..11.. IInnssttaanncee vvaarriiaabblleess

Accessible instance variables.

Internal instance variables. 33..99..22.. IInnssttaanncee mmeetthhooddss

33..99..22..11.. AAcccceesssiibblee iinnssttaannccee mmeetthhooddss

uurrll(())

A function that can be used in auth_loginform()a and auth_registerform. It returns the appropriate "action=" attribute to the form tag.

ppuurrll(())

A function that can be used in auth_loginform()a and auth_registerform. It prints the appropriate "action=" attribute to the form tag.

llooggiinn__iiff((\$\$tt))

A function that can be used to change the current user identity. See the section and example on using default authentication below.

uunnaauutthh((\$\$nnoobbooddyy == ffaallssee))

This function destroys the authentication information in \$this->auth, forcing the user to relogin the next time a protected page is being loaded.

\$this->auth["uname"] is being kept, so that the correct username is available as a default.

Since V6: To give the user the credentials of `nobody', pass true as the first parameter to unauth. This will also change \$this->auth["uname"].

Since V7.2: Passing \$nobody to this method is deprecated.

llooggoouutt((\$\$nnoobbooddyy == \$\$tthhiiss-->>nnoobbooddyy))
This function destroy all authentication information in
\$this->auth, forcing the user to relogin the next time a
protected page is being loaded.

Most applications want to use \$this->unauth() instead.

Since V6: To give the user the credentials of `nobody', pass true as the first parameter to logout. This defaults to the value you set in the class definition (\$nobody). logout() will call unauth() (passing \$nobody), so the behaviour is identical (except logout() will always clear \$this->auth["uname"] and unregister the auth class).

Since V7.2: Passing \$nobody to this method is deprecated.

iiss__aauutthheennttiiccaatteedd(())

Will return false, if the current authentication is invalid or expired. Will return the authenticated uid otherwise.

aauutthh__pprreeaauutthh(())

This function can be overridden in a subclass to Auth. It is being called as the very first step in the authentication process and has the opportunity to authenticate the user without a loginform being displayed (by deriving all necessary information telepathically, or by using cookies, or divining the user identities from the incestines of a dead squirrel).

If it returns a UID value, the user is authenticated and neither auth_loginform() nor auth_validatelogin() are called. If it returns false, all goes on as usual.

aauutthh__llooggiinnffoorrmm(())

This function must be overridden by a subclass to Auth. It should output HTML that creates a login screen for the user. We recommend that you use an include() statement to include your HTML file.

aauutthh__vvaalliiddaatteellooggiinn(())

This function is called when the user submits the login form created by auth_loginform(). It must validate the user input.

If the user authenticated successfully, it must set up several fields within the \$auth[] instance variable:

must contain the user id associated with that user.

must contain the user name as entered by the user.

must not be tampered with (field is maintained by start(), contains the time when the login expires).

if you want to use the permission feature, you must store the permissions of the validated user here. (Hint: due to a name conflict with sybase, "perm" is called "perms" in all the databases tables. Look for this small difference!)

See the example below for more information.

aauutthh__rreeffrreesshhllooggiinn(())

This function is called every refresh minutes. It must refresh the authentication informations stored in auth array by auth_validatelogin() method. It is not called if the user is logged in as nobody.

It must return true on success, false otherwise (i.e.: the userid is no longer valid).

aauutthh__rreeggiisstteerrffoorrmm(())
See auth_doregister().

aauutthh ddoorreeggiisstteerr(())

These functions mirror auth_loginform() and auth_validatelogin() in registration mode.

33..99..22..22.. IInntteerrnnaall iinnssttaannccee mmeetthhooddss

ssttaarrtt(())

Initialization function, does the authentication. If we are in log (login) mode, auth_loginform() is called to draw a login screen. When the login screen is submitted back, auth_validatelogin() is called to validate the login. If the validation was successful, the actual page content is shown, otherwise we're back at auth_loginform().

In reg mode, auth_registerform() is called to draw a registration form. When the registration form is submitted back, auth_doregister() is called to register the user and to validate the session. If registration was successful, the actual page content is shown, otherwise we're back at auth_registerform().

33..99..33.. EExxaammppllee

Use a subclass of Auth to provide parameters for your authentication class and to implement your own auth_* functions. class My_Auth extends Auth { var \$classname = "My_Auth"; # Object serialization support var \$lifetime = 15;## DB_Sql subclass and database table to use var \$database_class = "DB_Session"; var \$database_table = "auth_user";

```
## Some magic value to make our uids harder to guess.
var $magic = "Abracadabra";
## Use an own login form
function auth_loginform() {
 global $sess:
include("loginform.ihtml");
}
function auth validatelogin() {
 global $username, $password; ## form variables from loginform.ihtml
 ## If authentication fails, loginform.html will
 ## find $this->auth["uname"] set and use it.
 $this->auth["uname"]=$username;
 ## Value to return in case auth fails.
 $uid = false:
 ## Check the database for this user and password pair.
 $query = sprintf(
  "select * from %s where username = '%s' and password = '%s'",
  $this->database table,
  addslashes($username),
  addslashes($password)
 $this->db->query($query);
 ## If we found a matching user, grab the uid and permissions...
 while($this->db->next record()) {
  ## Required.
  uid = \frac{his}{\phi} = \frac{his}{\phi}
  ## Optional, for the perm feature.
  $this->auth["perm"] = $this->db->f("perms");
  ## if you use perm feature be aware, that the db-field in our
  ## example table is called "perms" due to a name conflict with sybase
return $uid;
```

Your loginform.ihtml contains HTML and PHP code to draw a login form. \$\text{\$this->auth["uname"] will be empty on the first login attempt and set on all further login attempts. You can use this to detect repeated login attempts and display an appropriate error message. You must print the result of \$\text{\$this->url()} to create your forms action attribute.

See the provided loginform.ihtml for an example.

Use the page management functions (see above) to use your authentication subclass. The feature name for authentication management is auth; provide the name of your Auth subclass as a parameter to the auth feature. The auth feature requires the sess feature:

```
page_open(array("sess" => "My_Session", "auth" => "My_Auth"));
```

33..99..44.. UUssiinngg ddeeffaauulltt aauutthheennttiiccaattiioonn

Many applications want to use \$auth and \$perm objects to protect functionality on a page, but do want to make the unprotected part of this page available to users with no account. This presents a kind of dilemma, because you need \$auth and \$perm objects to protect functionality on a page, but you don't want a login screen to appear by default.

Default authentication solves this dilemma by providing a special uid and uname "nobody", which is guaranteed to fail every permission check. If you set the nobody flag, \$auth will not create a login screen to force a user to authenticate, but will authenticate the user silently as nobody. The application must offer a login button or other facility for users with accounts to change from that id to their real user id.

To use default authentication, create a subclass of My_Auth as shown above with the nobody flag set (_N_o_t_e_: No need to extend in two steps. The only important thing here is that the nobody flag is set.)

```
class My_Default_Auth extends My_Auth {
  var $classname = "My_Default_Auth";

  var $nobody = true;
}
```

To create a page that uses default authentication, use the page management functions. Check for relogin requests with the login_if()

function. Create a relogin link on your page.

```
<?php
// using Default Authentication
page_open(array("sess" => "My_Session", "auth" => "My_Default_Auth"));
$auth->login_if($again);

if ($auth->auth["uid"] == "nobody"):
?>
<A HREF="<?php $sess->purl("$PHP_SELF?again=yes") ?>">Relogin</A>
to this page.
<?php endif ?>
```

33..99..55.. UUssiinngg CChhaalllleennggee--RReessppoonnssee AAuutthheennttiiccaattiioonn

As distributed, local.inc contains an example class named Example_Challenge_Auth, which uses a Challenge-Response authentication scheme. If the client browser supports Javascript, this login screen does not transmit passwords in clear over the network. If the client does not support Javascript, login is still possible, but passwords are transmitted in clear, as regular Example_Auth always does.

Example_Challenge_Auth is there to demonstrate advanced usage of PHP and Javascript and to show off the flexibility of the library base classes: The Challenge-Response authentication scheme has been implemented completely and naturally in local.inc by subclassing Auth with no alteration of library code.

Example_Challenge_Auth includes crloginform.ihtml. It also requires that the file md5.js is present in the document root directory of your web server. That file contains an implementation of the MD5 message digest algorithm done by Henri Torgemane. The basic idea behind this authentication scheme is simple: \$auth->auth_loginform() creates a challenge value which is incorporated into this form. When the user tries to submit the form, MD5("username:password:challenge") is calculated and filled into the reply field. The password field is

erased. The server can calculate the expected reply from the username received, the password in the database and the challenge, which it knows. It can compare the expected reply to the actual reply value. If they match, the user is authenticated.

If the reply field is empty and password is set, the server knows that the client cannot do Javascript. The user can still be authenticated, but the password is visible on the network.

The class is a dropin-replacement for Example Auth.

33..99..66.. TThhee coommppleettee gguuiiddee ttoo aauutthheennttiiccaattiioonn aanndd uusseerr vvaarriiaabblleess

This feature has originally been written for the PHPLIB mailing list by Kristian Köhntopp and was included into the documentation later.

33..99..66..11.. HHooww iiss tthhee AAuutthh ccllaassss uusseedd uussuuaallllyy??

Usually, you write code like this into the top of the page you want to protect:

```
<?php
page_open(array(
    "sess" => "My_Session",
    "auth" => "My_Auth"));
?>
<!-- your code here -->
<?php
page_close()
?>
```

33..99..66..22.. HHooww ddooeess \$\$aauutthh wwoorrkk iinntteerrnnaallllyy??

When you access this page, the call to page_open() is being made as the first thing on that page. page_open() creates an instance of My_Auth named \$auth and starts it. \$auth then detects that you are not authenticated (how it does, I will explain below) and displays loginform.ihtml. \$auth then exits the interpreter, so that <!-- your code here --> is never being executed or displayed.

The user now sits in front of a loginform.ihtml screen, which is shown under the URL of the page the user originally tried to access. The loginform has an action URL, which just points back to itself.

When the user filled out the loginform and submits it, the very same URL is requested and the above page_open() is reexecuted, but this time a username and a password are submitted. When the \$auth object is created and started, it detects these parameters and validates them, resulting in either a NULL value or a valid user id. If the validation failed, creating an empty user id, the loginform is displayed again and the interpreter exits. Again <!-- your code here --> is not executed.

If a UID is returned, that UID and a timestamp are being made persistent in that session and \$auth returns control to page_open(). When page_open() finishes, which it may or may not do, depending on the presence and result of an optional \$perm check, <!-- your code here --> is being executed or shown.

Later calls to other pages or the same page check for the presence of the UID and the timestamp in the sessions data. If the UID is present and the timestamp is still valid, the UID is retained and the timestamp is refreshed. On page_close() both are written back to the user database (Note: Authenticated pages REQUIRE that you page_close() them, even when you access them read-only or the timestamp will not be refreshed).

If the UID is not present (\$auth->logout() or \$auth->unauth() have been called, for example) or the timestamp has expired, \$auth will again intercept page display and draw the loginform.

The only way to get into a page with an \$auth object on it is to have a UID and a valid timestamp in your session data (Note: This is true even for default authentication. These create a dummy UID and timestamp in your session data).

33..99..66..33.. HHooww ddoo \$\$sseessss aanndd \$\$aauutthh iinntteerraacctt??

Your browser has a session cookie, named after your session class. This is the only thing that is ever shipped between your browser and PHPLIB, as far as core functionality is concerned. The session cookie value is used as a reference into active_sessions, to retrieve PHPLIB generated PHP code, which is then eval()ed and recreates your session variables within page_open().

Part of the \$auth object now makes itself persistent and is retrieved when the \$sess part of page_open() is being executed. This is just before the \$auth part of page_open() gets its turn, so that \$auth can rely on its persistent data being present when it is being called.

From the PHPLIB source you all know that \$auth has only one persistent slot, called \$auth->auth[], of type hash. This hash contains the slots uid, exp and uname. \$auth->auth["uid"] is the currently authenticated user id, \$auth->auth["exp"] is the currently active expiration timestamp (Unix time_t format) for that uid. \$auth->auth["uname"] is completely irrelevant as far as the regular PHPLIB Auth class is concerned. It is relevant in the context of the supplied default Auth subclass Example Auth, though.

So a session is authenticated, if it contains \$auth->auth["uid"] != false and time() < \$auth->auth["exp"].

33..99..66..44.. WWhheerree iiss tthhee bbeeeeff??

The original Auth class as included in PHPLIB makes no assumptions at all on how a loginform looks or how and where uids come from. There is no code at all in Auth that ever checks anything but the above two conditions. It is your responsibility to modify a subclass of Auth in a way that these conditions can ever be met.

Auth helps you in doing this by calling its own function \$auth->auth_loginform() when it wants to draw a loginform. Unfortunately this function is empty in Auth itself, so you have to provide an implementation for that. The suggested standard implementation in local.incs Auth subclass Example_Auth is

```
function auth_loginform() {
  include("loginform.ihtml");
}
```

and you put your code into that file. We also provide sample code for that file, but you are not limited to that code and may write a loginform.ihtml as it meets your needs.

When the loginform has been filled in and submitted back by the user, Auth calls \$auth->auth_validatelogin(). Again, this function is empty in Auth itself and so Auth by itself will never function correctly. You have to subclass Auth and provide your own implementation of \$auth->auth_validatelogin() in local.inc to make it work.

What you actually do in that function is completely irrelevant to Auth itself. It only exspects that you either return false, if the user-supplied authentication data was invalid, or a user id, if the user could be validated. Auth then takes care to create the appropriate entries (\$auth->auth["uid"] and \$auth->auth["exp"]) in the session

record.

33..99..66..55.. II ssttiilll ddoo nnoott uunnddeerrssttaanndd!! WWhhaatt aamm II ssuuppppoosseedd ttoo ccooddee?

You write your code into local.inc, after you have removed the classes Example_Auth, Example_Default_Auth and Example_Challenge_Auth from that file (keep a copy around, just for reference).

You code a class called My_Auth and you use that name later in your calls to page_open as an argument to the auth feature, as show at the start of this message. Follow the standard rules for deriving persistent classes in PHPLIB when you create your code, that is, do it like this:

```
class My_Auth extends Auth {
  var $classname = "My_Auth";
  // we inherit $persistent_slots and do not need to modify it.

// later code is inserted here
}
```

Now configure the lifetime of the authentication, that is, how many minutes in the future shall the current value of \$auth->auth["exp"] be? Also, name a database connector class and name the table that you will be using to check usernames and passwords.

```
// insert this code as indicated above.
var $lifetime = 15;
var $database_class = "DB_Example";
var $database_table = "my_special_user_table";
// later code is inserted here
```

Okay, now we have a basic implementation of My_Auth that is only lacking the required functions auth_loginform() and

auth_validatelogin(). Our implementation of auth_loginform() will have access to all \$sess variables by globaling \$sess into our context (because these can come in handy) and to all \$auth variables (via \$this).

```
function auth_loginform() {
  global $sess;
  include("loginform.ihtml");
}
```

The loginform is free to do whatever it damn well pleases to create a form for the user to supply the needed values for authentication. It has access to anything \$sess and anything \$this related.

The loginform will display some input fields for the user, for example a given name, a surname and a password. When the form is submitted back, auth_validatelogin() is being called. The form values are global variables (or \$HTTP_x_VARS[]) and must be imported into \$auth->auth_validatelogin(). Then, \$auth->auth_validatelogin() is free to do whatever it must do to produce a unique identifier for that user (or return false).

Suppose you created input fields named given_name, surname and password. So go ahead, global \$given_name, \$surname and \$password and set \$uid to false. Then create the SQL needed to access you user table and retrieve the user record from your database as indicated by \$given_name and \$surname and \$password.

The query may succeed, if a record with matching \$given_name, \$surname and \$password is present. In that case return the uid, which uniquely identifies exactly that (given_name, surname) pair. Else return false.

In code:

```
function auth_validatelogin() {
// import authentication data
global $given_name, $surname, $password;
 uid = false;
 $query = sprintf("select uid
             from %s
            where given_name = '%s'
             and surname = '%s'
             and password = '%s'",
       $this->database_table,
       $given_name, $surname, $password);
// we really should use addslashes() here,
// or have magic_quotes active.
// $auth->db is our DB_Example database connection
$this->db->query($query);
// now check for any results
 while($this->db->next_record()) {
  uid = this->db->f("uid");
 }
// either $uid is false now (no results)
// or set to the last retrieved value from the uid
// column.
// Anyway we are set now and can return control
return $uid;
```

Okay, that's all and useable now. There is room for some improvements, though: First we did not retrieve permission data, so this will not work, if we want to use the perm feature as well.

This is easily changed: Modify the query to select uid, perms instead of select uid alone. Of course, you may call your perm column whatever you like, just adapt the SQL accordingly. Also, add a line after the \$uid assignment so that the code looks like this:

```
$uid = $this->db->f("uid");
$this->auth["perm"] = $this->db->f("perms");
```

This will store the retrived perms value under the key perm within the \$auth->auth[] array. It will be kept around in that place in case \$perm is called and starts looking for the current permissions of that user.

Another possible improvement becomes apparent when you try to login and fail to do so correctly: auth_validatelogin() returns false and you hit the loginform again. Empty loginform that is, because we did not remember what you typed into the given_name and surname fields before. If we remembered what you typed, we could easily supply these values back to you so that you can correct them. We would also be able to detect if this is a second, third, ... attempt to login and display an appropriate error message somewhere in that loginform to inform the user of his or her typo. A convenient place to store these values is the \$auth->auth array, which is persistent anyway.

Standard Example_Auth uses the field \$auth->auth["uname"] to store that value, but you may use any field and as many fields as you like as long as you make sure not to clash with any of the three officially used fields, uid, exp, and perm.

Do not try to turn the global variables \$given_name and \$surname into persistent variables by calling \$sess->register("given_name") and \$sess->register("surname")! Remember: These are form variables! Never ever make form variables persistent and never ever trust unvalidated user supplied from the Internet!

So add the folling code just below the "global" line:

```
$this->auth["gname"] = $given_name;
$this->auth["sname"] = $surname;
```

and check for these two variables in loginform.ihtml at the appropriate places.

33..99..66..66.. OOkk,, II ddiidd tthhaatt aanndd iitt wwoorrkkss.. II eevveenn uunnddeerrssttoooodd iitt.. NNooww,, wwhhaatt

eexxaaccttllyy iiss tthhaatt uuiidd uusseedd ffoorr??

It is simply a token to indicate that the user is authenticated. We use a different token for each user, so that we can decide which user we are currently dealing with. You can think of the uid as a primary key for your auth_user table (or whatever it is being called in your current application). The (given_name, surname) tuple would also be a possible primary key, albeit a compound one. It is the external, human-readable (and probably sometimes very long) representation of the internal uid. The password field is functionally dependent on either of both key candidates.

The internal user id should never be presented to the user; the (given_name, surname) pair is much more natural to handle for the user and easier to remember (A user who does not remember his or her name would probably not be in a state of mind to operate the rest of the application anyway:-).

The internal user id should always be used to identify a user internally within an application, though. That is, because the uid is of a fixed length and has a known form and structure, so you can make assumptions. A given_name or surname may be of any length and may contain about any character, so you probably do not want to use this as a user-reference internally.

33..99..66..77.. BBuutt iiss tthhee uuiidd uusseedd iinntteerrnnaallllyy bbyy PPHHPPLLIIBB??

Yes, if you make use of the user feature of page_open(), that is, if you create user variables.

The User class is actually a subclass of Session. That is, user variables are just like session variables. They are even stored in active_sessions. The only difference is that the session has a different name (it is called Example User instead of Example Session,

if you use the classes and names supplied in local.inc).

And in Example_User, the user id of the authenticated user becomes the session id in the active_sessions table. That is the reason why we recommend md5(uniqid("abracadabra")) style uids.

33..1100.. PPeerrmm

Permission management relies on an authenticated session. It associates a set of required permissions with a page. The actual page content is only visible to users with ALL matching permissions; all other users are shown a screen of your design.

33..1100..11.. IInnssttaannccee vvaarriiaabblleess

Accessible instance variables.

33..1100..22.. IInnssttaannccee mmeetthhooddss

33..1100..22..11.. AAcccceessssiibblee iinnssttaannccee mmeetthhooddss

cchheeckk((\$\$rreeqquuiirreedd))

Checks that the currently authenticated user has all the rights that are specified in required. If not, perm_invalid() is called.

If one or more of the required rights or user rights are invalid (not to be found in the permissions hash), perm_invalid() is called as well.

hhaavvee__ppeerrmm((\$\$rreeqquuiirreedd))

Similar to check() in usage, only that it doesn't halt the session if the user doesn't have the appropriate rights: This function returns true, if the user has the required rights, false otherwise.

ppeerrmm__sseell((\$\$nnaammee,, \$\$ccuurrrreenntt ==
This function returns a SELECT-tag with the given name. Within this tag, all available permission values from \$perm->permissions are contained as OPTION tags.

If you supply a value for current, the permission value that matches current is SELECTED. If you supply a value for class, the tags are marked with that CSS stylesheet class.

33..1100..22..22.. IInntteerrnnaall iinnssttaannccee mmeetthhooddss

```
ppeerrmmssuumm(($$rriigghhttss))
  Logically or's all the rights and returns a pair (valid,
```

Logically or's all the rights and returns a pair (valid, or_result). If valid is true, an or_result is provided. If valid is false, the or_result is undefined and one or more of the rights do not exist at all. This is a severe error and the application should be halted at once.

```
ppeerrmm__iinnvvaalliidd(($$ddooeess__hhaavvee,, $$mmuusstt__hhaavvee))
Called in case of an access violation. does_have is a string
listing the rights the user actually has. must_have are the
rights the page requires.
```

33..1100..33.. EExxaammppllee

Use a subclass of Perm to provide parameters for your permission class and to implement your own perm_invalid function.

```
class My_Perm extends Perm {
 var $classname = "My_Perm";
 var $permissions = array (
  "user"
             => 1.
  "author"
             => 2.
  "editor"
              => 4.
  "moderator" => 8,
  "admin"
               => 16
 );
 function perm_invalid($does_have, $must_have) {
  global $perm, $auth, $sess;
  include("perminvalid.ihtml");
}
```

Use the page management functions (see above) to use your permission subclass. The feature name for permission management is perm; provide the name of your Perm subclass as a parameter to the perm feature. The perm feature requires the sess feature and the auth feature:

```
page_open(array("sess" => "My_Session", "auth" => "My_Auth", "perm" => "My_Perm"));
```

Use the check() instance method to protect your page:

```
$perm->check("admin"); ## This page is for users with admin rights only.
```

Use have_perm() to create protected functionality on a page:

```
<?php
  if ($perm->have_perm("admin")):
  ?>
  <h1>Admin only functionality</h1>
<?php
  endif;
  ?>
```

33..1100..44.. HHooww ppeerrmmiissssiioonnss wwoorrkk

Your subclass of Perm defines an array \$permissions, which translates permission names into bit patterns. For example, the definition of Example_Perm in the distributed local.inc defines the names user, author, editor, supervisor and admin, all of which translate into a bit pattern with a single bit set.

A user may be assigned any number of permissions as a comma separated list of permission names (no spaces!) in the perms column of the

auth_user table. The effective permissions of the user are determined by logically OR'ing the bit patterns of these permissions.

A page may require any permissions as a comma separated list of permission names (again no spaces!) with the \$perm->check() function. The required permissions are again determined by logically OR'ing the bit patterns of these permissions. Similarly, a page function may be protected by requiring permissions with \$perm->check().

Access is granted to a protected page or a protected page function, if the effective permissions of the authenticated user have all the required bits set, that is: If the effective permissions of the user logically AND'ed with the required permissions are equal to the required permissions.

With the permission names as defined in Example_Perm from the distribution, a user kris may be defined with admin permission in the auth_user table. A page that requires admin,user permission with \$perm->check("user,admin") is inaccessible to this user. This is how it is calculated:

Effective Permissions of User: admin

translates into: 16

Required Permissions of Page: user,admin

translates into: 1 OR 16 == 17

Permission Check:

Effective Permissions 16 Required Permissions 17 AND

ARE 16 & 17 = 16

MUST BE Required Permissions 17 -> access denied

The example permissions as defined in Example_Perm from the distribution are called _a_t_o_m_i_c permissions, because each of them has only a single bit set. Atomic permissions are the simplest of all

schemes, because they allow for easy permission checks: To access a page protected with user, admin, you need to have at least user, admin

rights in your auth_user table.

Another common scheme used in permission definitions are inclusive permissions. In this scheme, each permission definition has all bits of its predecessor set plus one addition bit. For example

defines a set of inclusive permissions. In this example, a user kris with admin permissions can easily access a page protected with editor permissions. This is how it is calculated:

```
Effective Permissions of User: admin translates into: 31
```

Required Permissions of Page : editor translates into: 7

Permission Check:

Effective Permissions 31
AND Required Permissions 7
ARE 31 & 7 = 7

MUST BE Required Permissions 7 -> access granted

Inclusive Permissions are easy to deal with, too, because a user with a _h_i_g_h_e_r access level may access all pages or page functions with a _l_o_w_e_r access level.

Due to limitations of your machines integer size you can only define up to 31 permission levels.

33..1111.. UUsseerr

The user class is an extension (a subclass) of the Session class. It keeps a list of global variable names and provides a set of functions to load and save these variables from and to a database. The same restrictions as for session variables apply to user variables.

Unlike session variables, user variables are not lost when the user stops and restarts the browser or moves to a different workplace (the session id is then lost and consequently all session variables are lost, since they are bound to the session id).

User variables require that the user logs in, because they depend on the availability of a User id to bind variables to this id. Thus, User is dependent on Auth.

The User class is an extension of the Session class. It has all instance variables and instance methods of Session, only that some are implemented different. This documentation only describes these differences.

Note that Session and User can successfully share a single active_sessions table in a database due to the different values in the name column.

33..111..11.. IInnssttaannccee vvaarriiaabblleess

Accessible instance variables.

Internal instance variables. 33..111..22.. IInnssttaanncee mmeetthhooddss

33..111..22..11.. AAcccceessssiibblee iinnssttaannccee mmeetthhooddss

```
rreeggiisstteerr(($$vvaarrnnaammee))
 Works as expected.
uunnrreeggiisstteerr(($$vvaarrnnaammee))
 Works as expected.
ddeelleettee(())
 Works as expected.
uurrll(($$uurrll))
 Not useful with User.
ppuurrll(($$uurrll))
 Not useful with User.
sseellff__uurrll(())
 Not useful with User.
ppsseellff__uurrll(())
 Not useful with User.
rree iimmppoorrtt\_\_ggeett\_\_vvaarrss(())
 Works as expected.
rreeiimmppoorrtt__ppoosstt__vvaarrss(())
 Works as expected.
rreeiimmppoorrtt__ccooookkiiee__vvaarrss(())
 Works as expected.
```

33..1111..22..22.. IInntteerrnnaall iinnssttaannccee mmeetthhooddss

```
ggeett__iidd(())
   This is only a stub implementation that depends on the user id
   provided by the page management functions. The page management
   functions will use $auth->auth["uid"], which is set up by Auth.
 ppuutt__iidd(())
   Empty. Not useful with User.
 sseerriiaalliizzee(($$pprreeffiixx,, &&$$ssttrr))
   Works as expected.
 ffrreeeezzee(())
   Works as expected.
 tthhaaww(())
   Works as expected.
 ggcc(())
   Works as expected. You do not want to use it, though.
 rreeiimmppoorrtt__aannyy__vvaarrss(($$aarrrraayynnaammee))
   Works as expected.
 ssttaarrtt(())
   Initialization function, to be called after object
   instantiation. Calls get_id() to get the current session id,
   creates a database connection, then calls thaw() to load all
   session variables. _N_o_t_e_: gc() activation is commented out!
   Remove the comments if you really want gc with User variables.
33..1111..33.. EExxaammppllee
Use a subclass to provide the appropriate parameters to your user
variables. Usually your subclass looks like this:
   class My_User extends User {
    var $classname = "My_User"; ## Persistence support
```

var \$that_class = "CT_Sql";

}

Remember that you have to provide a DB_Sql subclass with the parameters needed to access your database.

Use the page management functions (see above) to use your User subclass. The feature name for user variables is user; provide the name of your User subclass as a parameter to the user feature:

```
page_open(array("sess" => "My_Session", "auth" => "My_Auth", "user" => "My_User"));
```

Use the register() instance method to register variables as persistent. If \$user is your user object, use

```
$user->register("u");
```

to make the global variable \$u persistent. \$u may be a scalar value, an array or an object with persistence support slots.

Do not use the instance methods freeze() and thaw() directly, but use the page management functions instead.

_N_o_t_e_: Using default authentication and user variables is going to be a problem, because currently User does not do any locking. This is, because the DB_Sql has currently no portable locking mechanism.

44.. EExxtteennddeedd ffuunnccttiioonnaalliittyy

The section on extended functionality covers non-GUI classes that provide often needed application functions without a user interface. Some extended classes depend on core functionality, some contain independent classes.

Extended classes are treated differently from core classes in that their code is not automatically included by prepend.php3. You have to include the class definition manually where needed or you modify prepend.php3.

44..11.. CCaarrtt

The Cart class is programmatically independent, but makes sense only if its instances are made persistent in some way. The Cart class automatically registers itself as a session variable in its start() function.

Cart implements a shopping cart. At the moment, items within the shopping cart are independent of each other; the cart can only hold simple things. Support for compound articles that require other articles to function and provide a base for dependent articles is to be added at a future time.

An example of a simple article is any article with no options, for example an apple or a book. Common examples for compound articles are a pizza (which requires a foundation in either American or Italian style, a selection of toppings, and cheese, to function correctly) and a computer system (which requires a housing, a motherboard, RAM, a video card, etc to function correctly).

_N_o_t_e_: Cart was a core class up to _r_e_l_e_a_s_e_-5. If your applications uses the Cart class, you _m_u_s_t manually add the statement include("cart.inc") to your prepend.php3 file where indicated in that file.

_N_o_t_e_: The page management functions do no longer support the feature cart to set up and start the cart class. It is recommended that you use Session's auto_init feature instead to start your cart automatically or that you manually set up your cart.

44..11..11.. IInnssttaannccee vvaarriiaabblleess

Accessible instance variables.

44..11..22.. IInnssttaanncee mmeetthhooddss

44..11..22..11.. AAcccceesssiibblee iinnssttaannccee mmeetthhooddss

cchheecckk((\$\$aarrtt))

Checks that an item with the given article number \$art is in the cart. Returns an array of a boolean value and an integer number. If the boolean is true, there are number many articles of that article number in the cart.

rreesseett(())

Deletes all items in current cart, resetting \$this->currentItem to 1. Always returns true.

nnuumm__iitteemmss(())

Returns the number of articles in the current shopping cart, or false if cart is empty. For compatibility reasons, this function is available as tot_arts as well (but will print a warning if used by this name).

aadddd__iitteemm((\$\$aarrtt,, \$\$nnuumm))

Add \$num many articles of article number \$art to the current shopping cart. Returns the position number of \$art in the shopping cart.

rreemmoovvee__iitteemm

Remove \$num many articles of article number \$art from the shopping cart, if there are at least that many articles in the cart. Returns the position number of \$art in the shopping cart or false, if there weren't enough \$art to remove them from the cart. If the function does return false, the cart has not been modified.

sseett__iitteemm

Set the quantity of article number \$art in the shopping cart to exactly \$num. If \$num is set to zero, article is removed from cart. Returns the position number of \$art in the shopping cart.

sshhooww__aallll(())

If the shopping cart is empty, it will call show_empty_cart() once and then return.

Calls show_item_open() once at the beginning of a shopping cart listing. Then calls show_item() once for each item in the shopping cart. Calls show_item_close() once at the end of a shopping cart listing.

sshhooww iitteemm((\$\$aarrtt,, \$\$nnuumm))

This function should be provided by the user. It renders the HTML to display a single item from the cart. \$\frac{1}{2}\$ art is the article

number of the item and there are \$num of these in the cart.

sshhooww__ccaarrtt__ooppeenn(())

This function should be provided by the user. It renders the prologue HTML to display a shopping cart listing.

sshhooww__ccaarrtt__cclloossee(())

This function should be provided by the user. It renders the epilogue HTML to display a shopping cart listing.

sshhooww__eemmppttyy__ccaarrtt

This function should be provided by the user. It should render an appropriate message to symolize an empty cart.

44..11..33.. EExxaammppllee

Use a subclass of Cart to provide an implementation of show_item().

```
class My_Cart extends Cart {
 var $classname = "My_Cart";
// Look up article numbers...
 var $database_class = "DB_Article";
 var $database_table = "articles";
 var $db:
 var $sum = 0;
 function show_cart_open() {
 printf("\n");
  this->sum = 0;
 function show_cart_close() {
 printf("\n");
 printf("That's a total of %s.\n", $this->sum);
 }
 function show_item($art, $num) {
  if (!is object($this->db)) {
   $class = $this->database class;
   $this->db = new $class;
  $query = sprintf("select * from %s where artid = '%s'",
   $this->database_table,
   $art):
  $this->db->query($query);
  while($this->db->next_record()) {
   printf(" \n %s\n",
    $this->db->Record["name"]);
   printf(" %s\n",
    $this->db->Record["price"]);
   printf(" %s\n",
    $num):
   $rowsum = $num * $this->db->Record["price"];
   $this->sum += $rowsum;
   printf(" %s\n",
    $rowsum):
  printf(" \n");
```

To use a cart, create an instance of your Cart subclass and call start(). This will automatically register cart. It is recommended that you set in your Session subclass in local.inc the slot \$auto_init to the value setup.inc and create an include file of that name which contains the following code: global \$cart; ## \$cart is a global variable. \$cart = new My_Cart; ## Make a My_Cart instance named \$cart \$cart->start(); ## and have it register itself. Use add_item() and remove_item to work with your Cart: \$cart->add_item("101", 2); ## Add two pieces of "101" \$cart->remove_item("101", 1); ## Drop one piece of "101" Use show_all() to display the contents of your cart. \$cart->show_all(); ## What's in a cart, anyway?

44...11...44... OOnn uussiinngg CCaarrtt

To make use of the Cart class, you need to define a new table in your database that lists all articles you shop should sell. With PHPLIB

and MySQL we recommend that you create a new instance of PHPLIB for each virtual web server and a new database for each customer. This database should hold the active_sessions and auth_user tables as well as all application specific tables like for example the article list. In other words, with MySQL we strongly discourage that you use PHPLIB and the MySQL directive use _d_a_t_a_b_a_s_e___n_a_m_e together. There is no support if you do (there is no support if you do as we say, too, because PHPLIB is an open source product you are using on your own risk, but ...).

So let us assume you define a very simple new table articles with a structure like this:

```
# Table structure for table 'articles'

# CREATE TABLE articles (
    name text,
    price float(8,2),
    artid int(11) DEFAULT '0' NOT NULL auto_increment,
    PRIMARY KEY (artid)
);
```

This table has an article number called artid, and for each artid there is an article description name and a price. You may extend this minimal definition for your purposes by adding article groups, BLOBs with article images and more, but this will suffice for our example purposes.

Populate this table with some products that suit your taste.

The next step is to teach PHPLIB about the cart class. Three steps are necessary to do so:

• the Cart class has to be included on every page. Even on that pages that do not make use of the Cart class.

On that pages that use Cart, a cart subclass is instantiated and saved. On all subsequent pages, that Cart object is recreated and to be able to recreate the Cart object, PHP must know what a Cart object is. Since you cannot know which pages a user loads after he has put the first item into the Cart, we need to have a definition for Cart on _a_l_l pages.

The proper place to include the Cart definition from cart.inc is consequently prepend.php3. Edit prepend.php3 and require("cart.inc") as indicated by the comments in that file.

· a subclass of Cart has to be created to suit your tastes.

Your subclass of Cart will be called Example_Cart in this example. You may actually name it as you like, but you have to be consistent.

The definition of Example_Cart goes into local.inc anywhere below your definition for Example_Session. It looks like this

```
class Example_Cart extends Cart {
  var $classname = "Example_Cart";
}
```

and we will add additional code later in this example. That additional code will teach your shopping cart about the database table that holds your articles and so on.

 finally, you need to create an instance of your shopping cart class so that you have an object that actually holds the articles selected by the user.

We will use a very nifty feature of PHPLIB to create that object instance: If you set up PHPLIB properly, it is able to load and execute an include file every time a session is being created. We call this feature auto_init, after the instance variable of Session that controls it.

Go into local.inc and edit your subclass of Session. You will have some code like

```
class Example_Session extends Session {
  var $classname = "Example_Session";
  ...
}
```

in your local.inc. Add a line like

```
var $auto_init = "setup.inc",
```

to your definition of Example_Session and create a file setup.inc in the same directory that holds your local.inc. Whatever code is in this file will be executed every time we create a new session. The code is being executed after your \$sess, \$auth and \$perm objects are loaded and initialized, but does run from within a function context. You have to global everything you define to export it from that function context.

In setup.inc, create a global instance of Example_Cart named \$cart and register that variable with PHPLIB:

```
<?php
global $cart;
$cart = new Example_Cart;

// $sess is already global
$sess->register("cart");
?>
```

Now you have a \$cart object available by default on every page that uses PHPLIB. That object is created automatically at session startup, is carried from page to page by PHPLIBs session management and is destroyed by the garbage collection that reaps session records. You do not have to worry anymore about that cart, but simply use it anytime between page_open() and page_close(). PHPLIB does the rest for you.

The Cart class is actually dead stupid. It maintains an array \$cart->item[] that holds records about what the user bought. Each \$cart->item[\$x] consists of a \$cart->item[\$x]["art"], which is the article number of an item the user wants to buy and of a \$cart->item[\$x]["num"], which is the # of items with that article number that are wanted. \$cart->currentItem is the next \$x\$ to use for articles added to \$cart->item[].

You add articles to the shopping cart with

 $\$x = \$cart->add_item(\$art, \$num)$

This will add \$num items with the article number \$art to your cart contents. If you already have an item with that article number in your cart, the count for that article is increased by \$num. Otherwise a new article entry is being created and set to \$num. The function does return the \$x index into the \$cart->item[] array for that article.

To remove an item from the shopping cart, code

\$x = \$cart->remove_item(\$art, \$num)

This will remove \$num items with the article number \$art from your cart, if there are that many items in your shopping cart. If you do not have the \$art in your cart or there are not \$num many \$art in your cart, the function will return false and not remove anything from the cart. Otherwise, \$num articles with article number \$art are taken out of the cart and if the count for that article drops to zero while doing this, we even unset the array element.

You may check how many articles with a given article number are in the cart:

list(\$have, \$num) = \$cart->check(\$art)

The check function does return a two-element array. The first element \$have is true, if we have the wanted article in the cart. If \$have is true, \$num holds the number of articles with that number in the cart, otherwise \$num is undefined (actually, it is 0, but you must not rely on that).

Finally, we have a function

<pre>\$cart->show_all()</pre>		

which you may call to walk your shopping cart and have Example_Cart to generate a list of articles in your cart. That function will first call \$cart->show_cart_open(), for which you may provide code in your subclass. It will then call \$cart->show_item(\$art, \$num) for each item in the cart. We have a stupid default implementation for that function in Cart, but you may provide more sophisticated code in Example_Cart for that, too. Finally, at the end of your cart listing, \$cart->show_cart_close() is being called, which again may be code of yours.

The example in the previous section shows a more sophisticated implementation of a Cart subclass. That implementation uses show_cart_open() to create an opening table tag (formatted with a CSS class) and sets a counter \$cart->sum to zero.

In show_cart_close(), the table is being closed and the \$cart->sum counter is printed.

As you might have guessed, show_item(\$art, \$num) queries the database for each article number, retrieves the article description and prices and finally sums up all prices, taking the number of articles per article into consideration. It also generates table rows, printing a nice receipt for the customer.

44..22.. TTeemmppllaattee

_N_o_t_e_: If you think that this is like FastTemplates, read carefully. It isn't.

The template class allows you to keep your HTML code in some external files which are completely free of PHP code, but contain replacement fields. The class provides you with functions which can fill in the replacement fields with arbitrary strings. These strings can become very large, e.g. entire tables.

44..22..11.. IInnssttaanncee vvaarriiaabblleess

Accessible instance variables.

Internal instance variables.

44..22..22.. IInnssttaannccee mmeetthhooddss

44..22..21... AAcccceesssiibbllee iinnssttaannccee mmeetthhooddss

TTeemmppllaattee((\$\$rrooott ==

Constructor. May be called with two optional parameters. The first parameter sets the template directory (see set_root(), the second parameter sets the policy regarding handling of unknown variables.

sseett__rroooott((\$\$rroooott))

The function checks that \$root is a valid directory and sets this directory as the base directory where templates are being stored.

sseett__uunnkknnoowwnnss((\$\$uunnkknnoowwnnss == The function sets the policy for dealing with unresolved variable names. Must be either "remove", "comment" or "keep". If set to "keep", those are left untouched. If set to "comment", unresolved variable names are transformed into HTML comments reporting the error. If set to "remove", unresolved variable names are silently removed (the default).

sseett__ffiillee((\$\$hhaannddllee,, \$\$ffiilleennaammee == The function defines a filename for the initial value of a variable. It may be called with either a \$handle/\$filename pair or with a hash of \$handle/\$filename pairs. The files are not referenced yet, but only when needed.

sseett_bblloocckk((\$\$ppaarreenntt,, \$\$hhaannddllee,, \$\$nnaammee == A variable \$parent may contain a variable block named by \$handle. The function removes that block from \$parent and replaces it with a variable reference named \$name. If \$name is omitted, it is assumed to be the same as \$handle.

sseett_vvaarr((\$\$vvaarrnnaammee,, \$\$vvaalluuee == The functions sets the inital value of a variable. It may be called with either a \$varname/\$value pair or with a hash of

\$varname/\$value pairs.

```
ssuubbsstt(($$hhaannddllee))
```

The function returns the value of the variable named \$handle, with all defined variable values filled in. The resulting string is not "finished", that is, the unresolved variable name policy has not been applied yet.

ppssuubbsstt((\$\$hhaannddllee))

This is a shorthand for print \$this->subst(\$handle).

ppaarrssee((\$\$ttaarrggeett,, \$\$hhaannddllee,, \$\$aappppeenndd == ffaallssee)) The function substitutes the values of all defined variables in the variable named \$handle and stores or appends the result in the variable named \$target.

If \$handle is an array of variable names, \$append is ignored. The variables named by \$handle are being sequentially substituted and the result of each substitution step is stored in \$target. The resulting substitution is available in the variable named by \$target, as is each intermediate step for the next \$handle in sequence.

ppppaarrssee((\$\$ttaarrggeett,, \$\$hhaannddllee,, \$\$aappppeenndd == ffaallssee)) A shorthand for print \$this->parse(...).

ggeett__vvaarrss(())

Returns a hash of all defined values, keyed by their names.

ggeett__vvaarr((\$\$vvaarrnnaammee))

Returns the value of the variable named by \$varname. If \$varname references a file and that file has not been loaded, yet, the variable will be reported as empty.

When called with an array of variable names, an hash of values, keyed by their names, will be returned.

ggeett_uunnddeeffiinneedd((\$\$hhaannddllee))

The function will return a hash of unresolved variable names in \$handle, keyed by their names (that is, the hash has the form \$a[\$name] = \$name).

ffiinniisshh((\$\$ssttrr))

The function will returned the finished version of \$str, that is, the policy regarding unresolved variable names will be applied to \$str.

pp((\$\$vvaarrnnaammee))

The function will print the finished version of the value of the variable named by \$varname.

ggeett((\$\$vvaarrnnaammee))

The function will return the finished version of the value of the variable named by \$varname.

hhaallttmmssgg((\$\$mmssgg))

This function can be overridden by your subclass of Template. It will be called with an error message to print.

44..22..22..21. IInntteerrnnaall iinnssttaannccee mmeetthhooddss

ffiilleennaammee((\$\$ffiilleennaammee))

When called with a relative pathname, this function will return the pathname with \$this->root prepended. Absolute pathnames are taken unchanged.

The resulting filename must exist, or an error is generated.

vvaarrnnaammee((\$\$vvaarrnnaammee))

The function will construct a variable name regexp for a given variable name.

llooaaddffiillee((\$\$hhaannddllee))

If a variable is undefined or empty and is backed by a filename, the backing file will be loaded and the files contents will be assigned as the variables value.

hhaalltt((\$\$mmssgg))

This function is called whenever an error occurs and will handle the error according to the policy defined in \$this->halt_on_error.

44..22..33.. EExxaammppllee

The class manages a set of variables which are text strings. These strings may contain references to other variables in the form of "{variable}". When parsed or substituted, a variable reference is being replaced by the value of that variable.

A variable value may be defined manually by calling set_var("name", "value"); or it may be defined from a file by calling set_file("name", "filename.ihtml");. In the latter case, the contents of the file are being loaded when needed (as late as possible) and set as the value of that variable.

A third way to define a variable value is to call set_block("parent", "block", "name");. In this case, the variable named parent is being searched for a block that starts with <!-- BEGIN block --> and ends with <!-- END block -->. This string is removed from the variable parent and assigned to the variable named block. In parent, a variable reference to name is placed instead. If the optional parameter "name" is left out, "block" is being used instead.

Use Template directy or define a subclass of Template as needed.

Define a template file named page.ihtml as follows:

```
<html>
<head><title>{PAGETITLE}</title></head>
<body bgcolor="#ffffff">

ctd colspan=2><h1>{PAGETITLE}</h1>

<0UT}</td>

Content

</body>
</html>
```

This file contains a reference to the variable pagetitle and a reference to the variable named out. Another template file, named box.ihtml, contains a block named row with three variable references {TITLE}, {NUM} and {BIGNUM}:

```
<!-- start box.ihtml -->

    ctd colspan=2><b>{TITLE}</b>

        <!-- BEGIN row -->

        {NUM}

        {BIGNUM}

        <!-- END row -->
<!-- end box.ihtml -->
```

The following php3 file demonstrates how to use these templates:

```
<?php
 include("./template.inc");
 # create Template instance called $t
 $t = new Template("/page/to/webserver/template", "keep");
 # define variables named page and box, referencing files
 $t->set file(array(
  "page" => "page.ihtml",
  "box" => "box.ihtml"));
 # extract the block named "row" from "box", creating a
 # reference to {rows} in "box".
 $t->set_block("box", "row", "rows");
 # define the variables TITLE and PAGETITLE
 $t->set_var(array("TITLE" => "Testpage",
           "PAGETITLE" => "hugo"));
 # define NUM and BIGNUM, then append "row" to "rows"...
 for ($i=1; $i<=3; $i++) {
  n = i;
  n = i*10;
  $t->set_var(array("NUM" => $n, "BIGNUM" => $nn));
  $t->parse("rows", "row", true);
```

```
# build out from box, then build out from page...
$t->parse("OUT", array("box", "page"));

# finish out and print it.
$t->p("OUT");
?>
<hr>
<?php
# report leftover variables, if any.
print implode(", ", $t->get_undefined("rows"));
?>
```

55.. HHTTMMLL WWiiddggeettss CCllaasssseess

55..11.. SSqqll__QQuueerryy

Sql_Query will generate a query form for simple table queries: A list of field names, comparision operators and input fields is presented. The user may search for any values in any of the presented columns using SQL standard operators. Multiple query conditions are possible and these conditions can be joined using AND and OR operations.

The number of query conditions can be made variable. If so, the user may shrink and grow the query widget using the appropriate buttons.

All button labels and other messages of the interface are variable and held in language dictionaries. Currently, _d_e and _e_n dictionaries are provided.

55..11..11.. IInnssttaannccee vvaarriiaabblleess

Accessible instance variables.

Internal instance variables.

55..11..22.. IInnssttaanncee mmeetthhooddss

55..11..22..11.. AAcccceessssiibblee iinnssttaannccee mmeetthhooddss

ssttaarrtt(())

Initialization function. Currently empty.

ffoorrmm((\$\$bbaassee,, \$\$ooppttiioonn,, \$\$ccllaassss,, \$\$ttaarrggeett))

The function will generate and return HTML for the SQL Query selection form. All variables in the form will start with the prefix \$base and have numeric indices appended after an underline character. It is possible to have multiple Sql_Query instances on a single page, if they use different base characters.

The function must know the field names of the SQL table that is to be queried. \$option can be either a simple array of these field names (\$translate set empty) or a hash field name to long name (\$translate set to on).

All tags in the generated form are tagged with a CSS stylesheet class, if \$class is set to a CSS classname. \$class is optional and if it is left empty, no class attributes are generated. \$target is the URL of the SQL Query form target. It is optional and if it is left empty, a self referencing form is generated (recommended).

The function returns a string containing the HTML to render the SQL Query selection form.

wwhheerree((\$\$bbaassee,, \$\$iinnccrr))

When the form() generated page is submitted, a lot of parameters have to be evaluated and transformed into a SQL _w_h_e_r_e condition matching the user selections. The where() function takes care of all this; it just needs to be told which \$base prefix has been used in the form() call.

The \$incr parameter is optional and determines how many query condition rows are added or subtracted when the "More" and "Fewer" buttons are used. The default value is 1.

The function returns a string which can be successfully used behind a "where" keyword in a SQL query.

55..11..22..22.. IInntteerrnnaall iinnssttaannccee mmeetthhooddss

ppllaaiinn wwhheerree((\$\$bbaassee))

This function does all the work for where(), but does not resize the query condition window.

55..11..33.. EExxaammppllee

The Sql_Query class can be used directly. It is more useful when made persistent, so it is recommended that you add the line require("sqlquery.inc") to your prepend.php3 file where indicated in that file.

See the Table class in this section for a nice method to display and format query results. See the DB_Sql class (a core class) for a nice method to connect to databases.

The following code fragment is quite large, but contains a complete and working example using the Sql_Query, DB_Sql and Table classes to query a database table.

```
<?php
// We require() sqlquery.inc and table.inc in prepend.inc
// to make this example work!
page_open(array("sess" => "Example_Session"));

$db = new DB_Example; // That's a DB_Sql subclass.
$t = new Table; // For formatting results
$t->heading = "on"; // We want table headings..
?>
<html>
<head><title>Testseite</title>
<style type="text/css"><!--</td>
```

```
h1
         { font-family: arial, helvetica, sans-serif; color: #d33e30 }
table.test { background-color: #eeeeee }
th.test
        { font-family: arial, helvetica, sans-serif }
         { font-family: arial, helvetica, sans-serif }
td.test
table.query { background-color: #ccccc }
td.query { font-face: arial, helvetica, sans-serif }
--></style>
</head>
<body bgcolor="#ffffff">
<h1>Testpage</h1>
<?php
 // the following fields are selectable
 field = array(
  "username" => "Login Name",
  "password" => "Password",
  "perms"
            => "Permissions"
 );
 // When we hit this page the first time,
 // there is no $q.
 if (!isset($q)) {
  $q = new Sql_Query; // We make one
  $q->conditions = 1; // ... with a single condition (at first)
  $q->translate = "on"; // ... column names are to be translated
  $q->container = "on"; // ... with a nice container table
  $q->variable = "on"; // ... # of conditions is variable
               = "en"; // ... in English, please
  $q->lang
  $sess->register("q"); // and don't forget this!
 }
 // When we hit that page a second time, the array named
 // by $base will be set and we must generate the $query.
 // Ah, and don't set $base to "q" when $q is your Sql_Query
 // object... :-)
 if (isset($x)) {
  query = q->where("x", 1);
 // In any case we must display that form now. Note that the
 // "x" here and in the call to $q->where must match.
 // Tag everything as a CSS "query" class.
 printf($q->form("x", $field, "query));
 printf("<hr>");
 // Do we have a valid query string?
 if ($query) {
  // Show that condition
  printf("Query Condition = %s<br/>br>\n", $query);
  // Do that query
  $db->query("select * from auth_user where ". $query);
  // Dump the results (tagged as CSS class test)
```

```
printf("Query Results = %s<br/>h", $db->num_rows());
$t->show_result($db, "test");
}
page_close();
?>
</body>
</html>
```

55..22.. TTaabbllee aanndd CCSSVV__TTaabbllee

The Table class is a neat way to format two-dimensional associative arrays of data or the results of a database query into a table. Table and its subclasses allow you to simply pass them either an array or a query result and they spit out the proper HTML for a table containing all the values. Table has some primitive filtering capabilities making it useful even without subclassing, but for the full power of Table you have to write your own subclass.

When used with the check option, it is assumed that the table is part of a HTML FORM element. Code is generated to create an INPUT TYPE=CHECKBOX element before each table row. The checkboxes will form an array indexed by row number. The name of the array will whatever you set the check instance variable to.

Exactly one of two types of possible column filtering take place when each table row is generated. If the fields instance variable is set, only the columns keyed by the named fields in that array are shown in that order. That is, if you fill in the fields instance variable with array("a", "c", "e"), only the columns a, c and e become part of the generated table.

If fields has not been set, all data columns are traversed with each() and all columns whose names match the regexp in filter are shown in the table. By default, this regular expression lets through all column names that start with an alphabetic character and continue with either alphanumeric characters or "_" (underscore). This default has been chosen, because the DB_Sql database class uses mysql_fetch_array() internally to get data from the database and this function returns all columns twice with both a numeric index and proper column names. The default filter will make all data show up only once and with proper column names.

Additionally, the map_cols instance variable provides column name remapping. If map_cols is set, it will remap the name of the found column with a new name.

For instance, a table with the following columns, fname, lname, and mydate can be remapped to First Name, Last Name, and Date using the following code (where \$t is your instantiated Table class object):

```
$t->map_cols = array("fname" => "First Name",

"lname" => "Last Name",

"mydate" => "Date");
```

The map_cols instance variable also allows you to map column names to different languages using this same technique.

For derived classes, the instance variable add_extra has been added. If this variable is set, then the functions table_heading_row_add_extra() and table_row_add_extra() are called. In the Table class, these functions do nothing, but in derived classes override these functions to provide additional functionality that may be needed. For instance, hyperlinks to provide edit, delete, or view capabilities for that row could be easily added into these functions (in your derived Table class) allowing greater customization.

A subclass of Table, CSV_Table, is being provided to allow to create CSV representations of your data with minimal effort. CSV (comma separated values) can be imported by MySQL's LOAD DATA INFILE statement and many spreadsheet import functions.

The Table class now provides both high-level, mid-level and low-level functions through modularization. This allows programmers to use either the simplified high-level functionality or, depending on the degree of complexity needed, the power of the mid- or low-level functions. Every effort to maintain backwards compatibility has been applied. However, it would be a good idea to become familiar with the new functions if you use the Table class extensively. Typically, the high- and mid-level support functions begin with show_ while the low-level functions do not.

55..22..11.. IInnssttaanncee vvaarriiaabblleess

Accessible instance variables.

55..22..22.. IInnssttaannccee mmeetthhooddss

55..22..21... HHiigghh--lleevveell iinnssttaannccee mmeetthhooddss

```
sshhooww(($$aarryy,, $$ccllaassss ==
```

Will format and print the two dimensional array (or hash) \$ary as a table according to the filtering rules explained above. If \$class is set, each HTML element will be tagged as belonging to the named class; this is useful with cascading style sheets.

sshhooww__ppaaggee((\$\$aarryy,, \$\$ssttaarrtt,, \$\$nnuumm,, \$\$ccllaassss == Just as show(), but will show only num elements starting at start.

sshhooww__rreessuulltt((\$\$ddbb,, \$\$ccllaassss == Will format and print the result set of \$db. \$db is exspected to be a subclass of DB_Sql that has just been sent a query. Table will grab all available results from the result set of that query by calling \$db->next_record() repeatedly and format them into a table.

sshhooww__rreessuulltt__ppaaggee((\$\$ddbb,, \$\$ssttaarrtt,, \$\$nnuumm,, \$\$ccllaassss ==

Just as show_result(), but will show only num elements starting at start.

55..22..22.. MMiidd--lleevveell iinnssttaannccee mmeetthhooddss

sshhooww__ttaabbllee__rroowwss((\$\$aarryy,, \$\$ccllaassss== Walks the passed array displaying each row of data as an HTML table row.

sshhooww__ttaabbllee__rroowwss__rreessuulltt((\$\$ddbb,, \$\$ccllaassss== Walks the passed database object displaying each record as an

```
sshhooww__ttaabbllee__ppaaggee__rroowwss(($$aarryy,, $$ssttaarrtt,, $$nnuumm,, $$ccllaassss==
   Walks the passed array displaying each row of data as an HTML
   table row. However, data does not start displaying until $start
   element and end after $num rows.
 sshhooww ttaabblee ppaaggee rroowwss rreessuulltt(($$ddbb,, $$ssttaarrtt,, $$nnuumm,, $$ccllaassss==
   Walks the passed database object displaying each record as an
   HTML table row. However, data does not start displaying until
   $start record and ends after $num records have been displayed.
 sshhooww__ttaabbllee__hheeaaddiinngg__rrooww(($$aarryy,, $$ccllaassss==
   Uses the passed array to create an HTML header row.
 sshhooww__ttaabbllee__hheeaaddiinngg__rrooww__rreessuulltt(($$ddbb,, $$ccllaassss==
   Uses the passed database object to create an HTML header row.
 sshhooww__ttaabbllee__hheeaaddiinngg__cceellllss(($$ddaattaa,, $$ccllaassss==
   Walks the passed array and displays each item in an HTML table
   header cell.
 sshhooww__ttaabbllee__cceellllss(($$rrooww,, $$rrooww__kkeeyy,, $$ddaattaa,, $$ccllaassss==
   Walks the passed array and displays each item in an HTML table
   cell.
55..22..23.. LLooww--lleevveell iinnssttaannccee mmeetthhooddss
 ttaabbllee__ooppeenn(($$ccllaassss ==
   This function can be overridden by a subclass of Table. It is
   called as the very first step in table creation and should
   output HTML that opens a table (for example
   printf("<table%s>\n", $class?" class=$class":"");).
 ttaabbllee cclloossee(())
   This function can be overridden by a subclass of Table. It is
   called as the very last step in table creation and should output
   HTML that closes a table (for example printf("\n");/).
 sseelleecctt__ccoollnnaammeess(($$ddaattaa))
```

Internal function to generate a list of column names.

```
ttaabbllee_hheeaaddiinngg_rrooww(($$ddaattaa,, $$ccllaassss ==
 Internal driver function to generate a table heading row.
ttaabbllee_hheeaaddiinngg_cceellll(($$ccooll,, $$vvaall,, $$ccllaassss))
 This function can be overridden by a subclass of Table. It is
 called each time a table heading cell is to be generated.
 $col is the current column number, $val is the name of the
 column. $class is the HTML CSS class of the element that is to
 be generated.
ttaabbllee_hheeaaddiinngg_cceellll_ooppeenn(($$ccllaassss==
 Starts a header cell.
ttaabbllee_hheeaaddiinngg_cceellll_cclloossee(($$ccllaassss==
 Ends a header cell.
ttaabbllee_hheeaaddiinngg_rrooww_aadddd_eexxttrraa(($$ddaattaa,, $$ccllaassss==
 Virtual function for derived classes. This function is called
 after all header cells have been created. It allows the
 programmer to add additional HTML code to the header row before
 it is closed.
ttaabbllee__rrooww(($$ddaattaa,, $$ccllaassss ==
 Internal driver function to generate a table row.
ttaabbllee__rrooww__ooppeenn(($$rrooww,, $$ddaattaa,, $$ccllaassss ==
 This function can be overridden by a subclass of Table. It is
 called as the very first step in row creation and should output
 HTML that opens a table row.
 $row is the current row number. $data is a hash of column
 name/value pairs for that row and $class is an optional HTML CSS
 class name for all generated elements.
ttaabbllee__rrooww__cclloossee(())
 This function can be overridden by a subclass of Table. It is
 called as the very last step in row creation and should output
 HTML that closes a table row.
ttaabbllee__cceellll(($$rrooww,, $$cceellll,, $$kkeeyy,, $$vvaall,, $$ccllaassss))
 This function can be overridden by a subclass of Table. It is
 called each time a table cell is to be generated.
```

\$row is the current row number, \$cell is the current cell number. \$key is the current column name, \$val is the value of the cell. \$class is the HTML CSS class of the element that is to be generated.

```
ttaabbllee_cceellll_cclloossee(($$ccllaassss==
Ends a cell.

sseett_cchheecckkbbooxx_hheeaaddiinngg(($$ccllaassss==
This function creates an empty header cell to coincide with the checkbox option for that column.

ttaabbllee_cchheecckkbbooxx_cceellll(($$rrooww,, $$rrooww_kkeeyy,, $$ddaattaa,, $$ccllaassss==
Outputs HTML code to display a checkbox. This function runs if the member variable $check has been set. $check should be set to some key within the $data array (ex: if $data["myKey"], then set $check="myKey").

sseett_cchheecckkbbooxx(($$rrooww,, $$rrooww_kkeeyy,, $$ddaattaa,, $$ccllaassss==
Creates an HTML checkbox based on the passed data, only if the instance variable $check is set.
```

55..22..33.. EExxaammppllee

Table is not automatically included or prepended into each page. Include the table class into the pages that are to use Table. Then create an instance of Table:

```
<?php
// Include Table
require("table.inc");

// make a Table instance
$t = new Table;

// We want table headings to be printed.
$t->heading = "on";
```

Now create a two dimensional array or prepare a database query and have table print it.

```
// Create a database object
$db = new DB_Session;

// create a twodim array called $tab
$tab = $db->metadata("active_sessions");

// print that array
$t->show($tab, "metadata");

// prepare a database query
$db->query("select * from active_sessions");

// print that result
$t->show_result($db, "data");
```

55..33.. MMeennuu

Menu will generate a hierarchical menu of clickable items suitable as a navigation bar. Menu takes a tree definition of items as the basis for this navigation bar and knows which subtrees to fold, depending on the current position in the menu tree. Menu uses the current URL as presented in PHP_SELF to determine the current position in the menu tree automatically.

Menu does not depend on a hierarchical organisation of files in URL-space to generate a menu hierarchy. The organisation of menu items and the organisation of files in URL-space are in fact completely independent and Menu uses a mapping hash to derive a menu position from an URL. In the following class documentation we'll say URL when we mean files the latter and menustring when we mean the former. In the context of the Menu class, URLs are always relative URLs starting at the root of the local servers URL space, as we'll see them in PHP_SELF. They may look like /menu/index.php3. A menustring is usually numeric and all components have the same length, if necessary with leading zeroes. It may look like /001/007, denoting an item in main menu 1, submenu 7.

55..33..11.. IInnssttaanncee vvaarriiaabblleess

Accessible instance variables.

Internal instance variables.

55..33..22.. IInnssttaanncee mmeetthhooddss

55...33...22...11.. AAcccceesssiibblee iinnssttaannccee mmeetthhooddss

MMeennuu(())

Constructor. Calls Menu::setup() internally.

sshhooww(())

A shorthand notation for print \$this->get().

ggeett(())

This function will calculate the menu items visible from the current map position. The menu will be constructed by calling Menu::start_menu() first. For each visible menu item, Menu will check the current indentation level and the indentation level of the current menu cell. If the indentation level increases, Menu::shift_in(\$oldlevel, \$level) is called once, if it decreases, Menu:shift_out(\$oldlevel, \$level) is called once.

After that, Menu::get_cell(\$number, \$level) is called once. The number is an index into the visible array.

After all menu cells have been drawn, Menu::end_menu() will be called once.

```
ggeett__cceellll(($$nn,, $$lleevveell))
```

You are expected to implement this function yourself. It should render a single menu item. You may use the visible and item arrays for that purpose: \$m = \$this->visible[\$n] will return a menu string and \$attr = \$this->item[\$m] is a hash of attributes for that menu string. \$hilite = (\$this->visible[\$n] == \$this->map) is true for the current menu item, which should be rendered in way to stand out from the rest of the menu items.

ggeett__ttiittllee(())

This function will calculate the title of the current page based on the position of the current page in the menu hierarchy. This function uses \$this->title_delim to separate the components of the title. This function sets \$this>title to the calculated title and returns the title as a string.

sseettuupp(())

This function initializes the internal arrays of Menu and should be called once from the constructor. It actually is the constructor, but language stupidity has it that PHP3 constructor names vary with class names, which means that you have to write a new constructor for each subclass and call this function manually.

55..33..22..22.. IInntteerrnnaall iinnssttaannccee mmeetthhooddss

nnoorrmmaalliizzee__ppooss((\$\$ppooss))

This function looks at the current URL in \$PHP_SELF and tried to translate this into a menustring. If the URL matches a menustring directly, this is easy.

If not, the current URL will be sequentially shortened by applying the dirname PHP function to it until it matches. This allows you to create a single menu item for all files in a directory.

sspplliitt__ppaatthh((\$\$pp))

This function is used in the construction of the set of visible menu items. Given a menustring or a pathname, it constructs a series of pathnames which converge elementwise against the given pathname. That is, given the menustring /3/2, this function will return an array with the elements "" (the empty string), /3 and /3/2.

ffiinndd vviissiibbllee((\$\$rr))

This function calculates the actual set of visible URLs given a series of converging pathnames. It will include the set of children of each of these pathnames in the visible set, then sort this set numerically.

55..33..33.. EExxaammppllee

To use Menu, you must enable the require statement for menu.inc in prepend.php3. To use Menu_Button, you must enable the require

statement for menu.inc and menu_button.inc in prepend.php3.

Use a subclass of either Menu or Menu_Button to create a menu. Define a class Example_Menu in your local.inc file with a number of menu items in it. Do not forget to build a constructor.

```
class Example_Menu extends Menu {

# Map of PHP_SELF URL strings to menu positions

var $urlmap = array(

"/menu/index.php3" => "",

"/menu/item1.php3" => "/1",

"/menu/item11.php3" => "/1/1",

"/menu/item12.php3" => "/1/2",

"/menu/item13.php3" => "/1/3",

"/menu/item2.php3" => "/2",

"/menu/item21.php3" => "/2/1",

"/menu/item22.php3" => "/2/2",

"/menu/item221.php3" => "/2/2",

"/menu/item221.php3" => "/2/2",
```

```
"/menu/item222.php3" => "/2/2/2",
  "/menu/item23.php3" => "/2/3",
  "/menu/item24.php3" => "/2/4"
 );
# Information about each menu item
 var $item = array(
       => array("title" => "Main"),
  "/1" => array("title" => "Text 1"),
  "/1/1" => array("title" => "Text 1.1"),
  "/1/2" => array("title" => "Text 1.2"),
  "/1/3" => array("title" => "Text 1.3"),
  "/2" => array("title" => "Text 2"),
  "/2/1" => array("title" => "Text 2.1"),
  "/2/2" => array("title" => "Text 2.2"),
  "/2/2/1" => array("title" => "Text 2.2.1"),
  "/2/2/2" => array("title" => "Text 2.2.2"),
  "/2/3" => array("title" => "Text 2.3"),
  "/2/4" => array("title" => "Text 2.4")
 function Example_Menu() {
  $this->setup();
}
}
```

In each of your files mentioned in the above urlmap, create an instance of Example_Menu and call the show() method of that instance.

```
<?php
$m = new Example Menu;
?><html>
<head>
<title><?php $m->get_menu() ?></title>
</head>
<body bgcolor="#ffffff">
<h1><?php print $m->title ?></h1>
<?php $m->show() ?>
Content
</body>
</html>
```

55..44.. FFoorrmm

The form class (sometimes called OOH Forms) is a convenience library for dealing with html forms. It provides Javascript and server-side form validation, and is customizable and extensible.

55..44..11.. UUssiinngg OOOOHH FFoorrmmss

The OOH Forms library consists of five files: oohforms.inc of_checkbox.inc of_radio.inc of_select.inc of_text.inc of_textarea.inc. oohforms.inc automatically includes the others. You may wish to modify this so you can manually include the files for just the form elements you use. Or you may wish to cut and paste the contents of the element files into oohforms.inc to save the overhead of multiple includes. Determining the appropriate configuration of the files for your site is left an exercise for the reader; for most purposes require("oohforms.inc") will suffice.

In general, the structure of a page that uses oohforms is as follows:

```
// include the library
require("oohforms.inc");
f = \text{new form}
                           // create a form object
$f->add element(...);
                        // set up form elements
$f->add_element(...);
$f->add element(...);
if ($submitname)
                           // Is there data to process?
 if ($err = $f->validate()) { // Is the data valid?
  echo $err;
                 // No; Display error
  $f->load_defaults(); // Load form with submitted data
 else {
  /* Process data */
                          // Data ok; Do something with it
 }
$f->start(...);
                        // Start displaying form
$f->show element(...); // Show elements
$f->show_element(...);
$f->show element(...);
$->finish();
                        // Finish form
```

There are obviously many variations on this theme, but that covers the basics. Specific methods are documented below.

```
ssttaarrtt(($$jjvvssnnaammee,,$$mmeetthhoodd,,$$aaccttiioonn,, $$ttaarrggeett))
Outputs the initial <form> tag and sets up some initial state
needed by the class. All of the arguments are optional, though
you'll usually want to use at least one in order to get
Javascript validation. $jvsname is an arbitrary string used to
link the Javascript to the form; if it is empty (the default),
no Javascript validation is provided. $method is the HTTP
method used to submit the form; default is "POST". $action is
the URL that receives the form submission; default is $PHP_SELF.
$target is the frame target for the form results; default is
```

ffiinniisshh((\$\$aafftteerr,,\$\$bbeeffoorree))

Outputs the any hidden fields that were added to the form, the final </form> tag, then the Javascript validation function (if necessary). \$after and \$before are both optional; if either is a nonempty string it is output as additional Javascript to be run on submission of the form, either before or after the validation code. Though the order of the arguments may seem counterintuitive, it makes the common case easier to type; in general you'll want to wait until after the validation has taken place to do anything fancy with Javascript. Note that unlike with validation, OOH Forms has no way of giving you server side functionality equivalent to the Javascript you use here.

aadddd__eelleemmeenntt((\$\$eelleemmeenntt))

add_element is used to define the attributes of a particular form element so that the other class methods can use and manipulate it properly. add_element takes exactly one argument: an associate array whose key value pairs are used to define the form element type and it's various attributes. Some of these attributes correspond to html attributes, while others are needed for the value added features of oohforms. The attributes and the syntax and semantics of the values they take are documented below; note that not all element types use all of the attributes

ttyyppee

The type of element this is; can be "submit", "hidden", "text", "textarea", "select", "radio", "checkbox", or "file".

nnaammee

A string naming this element. This name will be used as an argument to other methods and will be the name="" value in the generated html (and hence the variable name in PHP). DDoo nnoott append [] to the name if you want an array valued element; set the multiple attribute instead.

vvaalluuee

The default value of this form element. If the form element has the multiple attribute set, value can be an array. If the this is a select element, value can refer to either the textual name (label in the options array) or the submission value (value in options).

mmuullttiippllee

A flag to tell oohforms to assume this element is array

valued. The use of this flag is most straightforward with select elements, but it can be use with text and checkbox elements as well. See the show_element documentation for more information about how oohforms deals with such elements.

eexxttrraahhttmmll

Extra html code that is inserted verbatim into the opening form tag. For select elements, the extra html goes into the select tag, not the option tags.

ssiizzee

For text elements, used to set the html size attribute that gives the width in characters of the text entry box. For select elements, the size (number of options visible at once) of the selection box. Validation is only performed on select elements if size is set to 1, since select validation doesn't make much sense if you can see multiple options at once. For file elements, the maximum size file upload to accept.

ppaassss

If set for a text element, renders the html as a password element, i.e. entry displays as asterisks.

ssrrcc

If set for a submit element, convert to an image element and use the value of src as the source URL for the image.

mmaaxxlleennggtthh

Used verbatim as the maxlength html attribute in text elements.

mmiinnlleennggtthh

If length_e is set, this is the minimum length text element entry accepted by the validator.

lleennggtthh__ee

If set, validate the text element to assure it has at least minlength characters. The value of length_e is the error string to be used in the event of failed validation.

vvaalliidd ee

If set, perform validation on a text, radio, or select element. For a text element, validation assures a match with valid_regex. radio element validation assures that one of the options in the group has been chosen. select validation only works for select elements with multiple unset and size equal to 1; the validator will not accept the first option of accept menu, assuming that it is some sort of prompt (e.g.

"Please select an item"). In all cases, the value of valid_e is the error string used for failed validations.

vvaalliidd__rreeggeexx

Regular expression used to validate entry into a test field if valid_e is set. Note that if you must use ^...\$ if you want the regex to match the whole entry.

iiccaassee

If set, regex matching is case insensitive.

cchheecckkeedd

Only used for a checkbox element that does not have multiple set. If checked is set, the element will display as checked.

rroowwss

Used verbatim as the rows= element in a textarea element.

ccoollss

Used verbatim as the cols= element in a textarea element.

wwrraapp

Used verbatim as the wrap= element in a textarea element.

ooppttiioonnss

Array of options to be displayed in a select element. If the elements of the array are simple values (strings or numbers), they are simply displayed verbatim and used as the value for that particular option. The elements may themselves be associate arrays with keys "label" and "value". In that case, the value of "label" is displayed and the value of "value" is used on submission.

Examples:

```
$f->add_element(array("type"=>"checkbox",
                 "name"=>"compress",
                 "multiple"=>1));
$f->add element(array("type"=>"textarea",
                 "name"=>"comment",
                 "rows"=>6.
                 "cols"=>40.
                 "value"=>""));
$o = array(array("label"=>"Please Select","value"=>0),
      array("label"=>"Apple", "value"=>1),
      array("label"=>"Orange","value"=>2),
      array("label"=>"Pear","value"=>3),
      array("label"=>"Grape","value"=>4));
$f->add_element(array("type"=>"select",
                 "name"=>"menu",
                 "options"=>$o,
                 "size"=>1.
                 "valid_e"=>"Please select a fruit",
                 "value"=>"apple"));
```

sshhooww__eelleemmeenntt((\$\$nnaammee,,\$\$vvaalluuee))

Outputs the form element named \$name. Usually, the second argument is not used. It is always necessary for radio elements and checkbox elements with the multiple attribute set, since many of these may have the same name. It also must be used for submit elements to label the submission button; the value attribute is not used for submit elements. For other elements that may be array valued (notably text elements) multiple calls to show_element will show successive values.

llooaadd__ddeeffaauullttss((\$\$eelleemmeenntt__lliisstt))
Sets the default value of form elements to the value of the PHP variables with the same name. This is generally used to redisplay a form with the same values which were submitted. The argument is optional; if given it is an array of element names; only these elements area affected.

vvaalliiddaattee((\$\$rreessuulltt,,\$\$eelleemmeenntt__lliisstt))
Validates a form submission. If all of the elements are valid,
return \$result, otherwise return the relevant error message (the
valid_e or length_e attribute of some form element). \$result
defaults to false. The second argument is also optional; it is
an array of names of elements to validate.

ffrreeeezzee((\$\$eelleemmeenntt__lliisstt))

Freezes the form elements whose names are given in the array

passed as the argument. If no argument is given, freeze all of the elements. Frozen elements are rendered as plain, static html rather than form widgets. This static rendering is accompanied by appropriate hidden elements to simulate the affect of using the unfrozen version of the element.

55..44..22.. CCuussttoommiizziinngg OOOOHH FFoorrmmss

Since OOH Forms is object oriented, it can be easily customized by extending the classes that define the element types. In general, you must make sure your derived class has a constructor and you may override any of the self_* functions of of_element. The source for the existing elements is the best documentation for how to do this properly, but a few general notes follow.

sseellff__sshhooww((\$\$vvaall,,\$\$wwhhiicchh))

Display an instance of this element unfrozen. \$\square\$val is the \$\square\$value argument of show_element if there was one; \$\square\$which can be used as an index for array valued elements; it is equal to the number of times show_element has been called for this element previously. This function must return the number of hidden tags output.

sseellff_sshhooww_ffrroozzeenn((\$\$vvaall,,\$\$wwhhiicchh))
Display an instance of this element frozen. In addition to the
html to show the frozen element, you must output tags for hidden
fields to duplicate the effect of submitting an unfrozen element
of this type. The function must return the number of hidden
tags output;

sseellff vvaalliiddaattee((\$\$vvaall))

Validate \$val for this element. If it is valid, return false, otherwise return the relevant error string.

sseellff__pprriinntt__jjss((\$\$nnddxx__aarrrraayy))

Print out Javascript code to validate this element. \$ndx_array is an array of the indices of the elements to be validated as used in the Javascript form.element[] array. This is needed since the extra [] in array valued element names confuses Javascript.

sseellff__llooaadd__ddeeffaauullttss((\$\$vvaall))

Set the default value for this element to \$val. Usually the default definition of this function, which just copies \$val to \$this->value is all that is needed, but there may be special cases that need to do something else. See the implementation of the checkbox element for an example.

55..55.. ttppll__ffoorrmm

The tpl_form class is intended to provide a general framework for HTML form deployment. It heavily depends on OOH Forms library, so it is required that you read and understand the relative documentation.

The main idea is that you create a black box by sub-classing tpl_form, provide some HTML code mixed with OOH Forms calls to actually render the form itself. Your application will then use that black box to obtain some input from the user. Your application doesn't have to know how to handle user input, nor how to validate input data, since internal methods will take care of that.

This approach is very similar (I think) to OOH Forms one, only at a higher level. OOH Forms elements have no way to communicate with each other, and are only able to perform "simple" checks on data integrity, while tpl_form adds a consistent interface for complex data evaluation and processing.

Furthermore, the get_default_values and set_default_values methods can be used to maintain user input between sessions, without worrying about serialization of form variables (a BAD THING(tm)), using an hash array containing field names and values.

You'll note that an array is used to share data with the application. You may object this is kinda futile, since all user input data may be found in global vars and HTTP_POST or HTTP_GET global hashes. This is true, and in the general case you'll pass back and forth an empty array. The values variable is intended for use in very complex dataentry setup, where a form behaviour may depend on previous data entered by the user. In this case, if all forms cooperate reading and writing to values hash array, final result may be constructed step by step across multiple HTML pages.

55..55..11.. IInnssttaanncee vyaarrijaabblleess

Internal instance variables.

55..55..22.. IInnssttaannccee mmeetthhooddss

iinniitt((\$\$vvaalluueess))

This is a sort of a constructor for the class. \$values is an hash array intended to store form values to be passed back to the application via get_values method.

ggeett__ddeeffaauulltt__vvaalluueess(())

Returns an array containing all data submitted by the user for the form. This array is intended to be passed to set_defaults_values some time later.

sseett__ddeeffaauulltt__vvaalluueess((\$\$ffvv))
Restore form defaults from an array as returned by get_default_values.

ddiissppllaayy(())

Actually display form fields. This method should not be overridden in descendants. User should instead provide a file named as the derived class and with ".ihtml" extension which will be automatically included.

ggeett__vvaalluueess(())

This method should not be overridden. It is intended as the main interface between the application and the form. Once the form has been properly derived to suit designer's needs, application calls get_values and receives back the array passed to init, eventually modified by process_input method, or false if user input is invalid. In that latter case, the application should call display to (re)present the form to the user, eventually filled with proper default values.

cclleeaarr(())

Sort of a "destructor". There should no real need to call it, except maybe freeing some memory. May be called from the application, otherwise is not executed. Returns true.

55..55..22..22.. IInntteerrnnaall iinnssttaannccee mmeetthhooddss

sseettuupp(())

Init the Form object, which will contain all fields info. The hidden field form_name, automatically added by this routine, is used by other methods to determine if form has already been

submitted by the user. You shouldn't override this in descendants, use setup_fields instead. Returns true.

sseettuupp__ffiieellddss(())

Override this method in order to provide form fields definition that suit your needs.

vvaalliiddaattee(())

Validates user input. This method should not be overridden in descendants. See validate_input instead. Returns false on error and sets error variable accordingly.

vvaalliiddaattee__iinnppuutt(())

This method should be overridden in descendants, in order to provided complex validation methods (i.e. field2 should not be empty IF field1 == "other"). Should return false on error and set error variable with a sensible error message.

pprroocceessss(())

Process user data. This method should not be overridden by descendants. See process_input and process_default instead. Returns true on success, false otherwise.

pprroocceessss__iinnppuutt(())

This method should be overridden in descendants. It is executed after validation has taken place. The data passed to the form could be used to fill values array.

pprroocceessss__ddeeffaauulltt(())

This method should be overridden in descendants. It is executed when form validation fails or before presenting the form for the first time. Should be used to bypass form displaying if data can be extracted from previous actions, divination, penguin fly watching or whatever.

55..55..33.. EExxaammppllee

Suppose you have a form that the user should fill with her (eheh) name and e-mail. You want to check wether this e-mail is valid, or your blind date setup is lost. A... er... simple regular expression for validating syntactically the e-mail is presented in the example code below.

\$this->form_data->add_element(array(

"type"=>"text",

```
"name"=>"email",  
"valid_e"=>"Syntax error in E-Mail address.",  
"valid_regex"=>"^([-a-zA-Z0-9.]+@[-a-zA-Z0-9]+(\.[-a-zA-Z0-9]+)+)*$"  
));
```

Now, this piece of code should do the job, but since you're feeling very paranoid today, you'd also like to validate the host name part of the address with DNS. So, you put together some code which takes an hostname in input and reports true on valid hostname, false otherwise (HINT: on PHP Code Exchange you should find a procedure for "active" email validation).

Now that you have your shining new code, you can check the address. The user fills out the form, you parse user input, no syntax errors, time to call your mycheckhost from the application. If the function is ok update your database, else load defaults into the form, display again, close the page, goodbye.

I've done something similar for MANY forms, some of them with very complex validation procedures, and I found that was too easy producing very bad and unreadable code (well, I actually realized that the first time I had to change some logic in data validation...).

tpl_form should provide a solid framework to build your forms with, and all the code will be self-contained and separated from main application logic. Hope you'll like it.

Time to see some code. First of all, class declaration, sub-classing tpl form:

```
class myform extends tpl_form {
 var $classname = "myform";
 function setup_fields() {
  $this->form_data->add_element(array(
  "name"=>"email",
  ..., // See previous code snippet
  $this->form_data->add_element(array(
   "name"=>"submit",
   "type"=>"submit",
   "value"=>"submit"
  ));
 function validate_input() {
  global $email;
  list($uname, $hostname) = split("@", $email);
  if (! mycheckhost($hostname)) {
   $this->error = sprintf("Sorry, unknown host %s, try again.", $hostname);
   return false;
  // Additional checks here...
  return true:
 }
```

You shuld provide a file myform.ihtml with HTML and php code to render the form. Minimalistic example:

```
<html>
<body>
<pphp
$this->form_data->start_form($this->classname, "POST", $sess->self_url(), "");
printf("%s<br/>his->error);
$this->form_data->show_element("email");
printf("<br/>his->form_data->show_element("submit");
$this->form_data->show_element("submit");
$this->form_data->finish();
?>
</body>
</html>
```

Your tpl_form class is complete, and will only need a little work on the artistic side... 8-) To use your brand new class, include class definition code into your application, then...

Hope this very little example does help in understanding the real power of tpl_form class, at least in terms of rapid designing and code partitioning.

55..66.. TTrreeee

The Tree class can render tree structures such as directory hierarchies and menu structures as HTML. The structure must be given to Tree as an nested array of arrays of arbitrary depth.

The idea of Tree is, that there are several mathematical models a tree could be viewed: One model is a data structure like nested arrays or a pointer structure from where you can print multidimensional graphics and can do other neat things like deleting one part of the tree or inserting a whole subtree. But you can also imagine a tree as a one dimensional string or as a sequence of function calls (which is nearly the same in the mathematical sense).

To generate HTML-code from a tree-structure it is like this: You need at the end a one-dimensional string, which tells the browser what to

do. The Tree class assists you in generating this string in this way, that it will go through the whole tree and call several functions on every stage trough the way. It will be your task to change the functions, so that a nice layout will be generated.

55..66..11.. IInnssttaannccee vvaarriiaabblleess

Accessible instance variables.

55..66..22.. Ilnnssttaanncee mmeetthhooddss

55..66..22..11.. AAcccceesssiibblee iinnssttaannccee mmeetthhooddss

bbuuiilldd ttrreeee(())

This function is completely user driven! You have to create an array with the structure described below. See the example for details.

Don't be shy to create some own functions which are called by build_tree() - e.g. for recursive calls.

ggoo__ttrroouugghh__ttrreeee((\$\$kkeeyy==

This is the most important function of this class. It will call the output functions in the right order with the correct parameters.

All variables are optional. The parameters are perhaps useful, if you want to display only partial trees, but this is not supported now.

ppaatthh__ttoo__iinnddeexx ((&&\$\$ppaatthh,,\$\$kkeeyy== This function is mostly used internally, but could be useful for you to generate \$this->tree. This function generates a PHP3 associate array-index string from a path, which is also a string but truncated by \$this->delimiter. If \$key is given, it will be added to \$path (minds empty path and so on).

Example:

```
$t->delimiter="/";
$path= "usr/local/lib";
## $path must be given as a var, because it is called by reference!
$bla = $t->path_to_index($path,"etc");

## $path is now "usr/local/lib/etc"
## $bla is now ["usr"]["local"]["lib"]["etc"]
```

ppaatthh_ttoo_ppaarreenntt ((&&\$\$ppaatthh))
This function isn't used internally, but could be

This function isn't used internally, but could be useful for you during generating the output tree. It will remove one from the depth of the path.

Example:

```
$t->delimiter="/";

$path= "usr/local/lib";

$bla = $t->path_to_parent($path);

## $path is now "usr/local"

## $bla is now ["usr"]["local"]
```

ppaatthh__aadddd ((\$\$ppaatthh,,\$\$kkeeyy))
This function is the 'non-call-by-reference-version' of path_to_index. It will add the \$key to the path and return it.

```
ppaatthh__ssuubb (($$ppaatthh))
 This function is the 'non-call-by-reference-version' of
 path_to_parent. It will find the parent of path and return it.
ppaatthh__iinnddeexx (($$ppaatthh))
  This function is the 'non-call-by-reference-version' of
 path to index(). It will return the associate key to the tree
 described by path.
ssttaarrttttrreeee (())
 This function is called by go_trough_tree() at the beginning of
 the output of a tree.
 All *tree-functions are called by go_trough_tree(), but it's
 your turn, to give them a nice layout. I think it is possible to
 generate nearly every kind of tree-layout with this. Have a look
 at the variables: E.g. $depth makes it possible to handle every
  "level" in another manner.
ggrroowwttrreeee (($$kkeeyy,,$$vvaalluuee,,$$ppaatthh,,$$ddeepptthh,,$$ccoouunntt,,$$ppccoouunntt))
  This function is called by go trough tree() at the beginning of
 the output of a tree.
 It is called every time, when go_trough_tree() will call itself
 recursively. You could also say it is called, when the current
 item has a successor.
lleeaaffttrreeee (($$kkeeyy,,$$vvaalluuee,,$$ppaatthh,,$$ddeepptthh,,$$ccoouunntt,,$$ppccoouunntt))
 This function is called, when the current item has _n_o successor.
sshhrriinnkkttrreeee (($$kkeeyy,,$$ddeepptthh))
 This function is the "opposite" of growtree(). It is called
 every time, when the current item was the last item in this sub-
 list.
eennddttrreeee(())
 Called when leaving tree.
```

55..66..33.. TThhee TTrreeee AArrrraayy

As said above, before you call go trough tree(), first \$tree must be generated.

\$tree consists of nested arrays of arbitrary depth. An example:

```
$t= new Tree:
t->tree = array(
         "usr" => array(
               => "allowed",
          "lib" => "forbidden",
          "local" => "allowed",
          "bin" => "forbidden",
          "etc" => array(
            0 =  "allowed",
           "hosts" => "forbidden".
           "mailcap"=> "allowed"
          "var" => "allowed".
          "tmp" => "allowed"
         "root" =>"forbidden"
        );
$t->go_through_tree();
print $t->outp;
```

This is a completely recursive structure and I think, it is clear, how to create it with a recursive call of a function. If not, see the example below.

One little quirk has to be explained, because it is a little bit confusing: the array name 0 (zero) is used for the value of the parent element. As shown in the example, an element with children (for example "etc") cannot have attributes (such as "allowed"). Instead the value of this element is stored in a pseudo-child named 0. If this element is not present, it will have the value "Array" (perhaps something that should be changed).

The output of this example if you don't change the output-functions will look like this:

```
^---- usr->'allowed': 'usr' (1) [1/2]

| ^---- lib->'forbidden': 'usr^lib' (2) [2/7]

| O---- local->'allowed': 'usr^local' (2) [3/7]

| O---- bin->'forbidden': 'usr^bin' (2) [4/7]

| O---- etc->'allowed': 'usr^etc' (2) [5/7]

| | ^---- hosts->'forbidden': 'usr^etc^hosts' (3) [2/3]

| | \--- mailcap->'allowed': 'usr^etc^mailcap' (3) [3/3]

| O---- var->'allowed': 'usr^var' (2) [6/7]

| \--- tmp->'allowed': 'usr^tmp' (2) [7/7]

\--- root->'forbidden': 'root' (1) [2/2]
```

Looks a bit confusing. From left to right the fields are

- · The _i_n_d_e_x_-_n_a_m_e of the current field
- · The _v_a_l_u_e of this field
- · The _f_u_l_l _p_a_t_h to this field (see path_to_*-functions)
- · The current $_d_e_p_t_h$ or $_l_e_v_e_l$
- The current _e_l_e_m_e_n_t _n_u_m_b_e_r. See below to understand, why it will begin sometimes with "2" in this example!
- · The _n_u_m_b_e_r _o_f _e_l_e_m_e_n_t_s in the subtree at this depth

55..66..44.. EExxaammppllee

My example is just going trough the directory structure of your hard disk.

The following code could read it:

```
class dir_Tree extends Tree {
   var $classname = "dir_Tree";
   var $delimiter="/";
   var $tdat;
   function build_tree ($path=".") {
     $this->tree=$this->recurs_dir($path,0);
   }
   ## This example code can read and output 1000 directory entries with
   ## many subdirs in about 20 seconds on my system (P200, 64 MB);
   ## 220 dir entries with a maximum depth of 4 are read in 2 seconds.
   ## This is ok. :)
   function recurs_dir ($path,$depth) {
   GLOBAL $flap_out;
     $d=opendir($path);
     while ( $name=readdir($d) ) {
        $pathname=$path . $this->delimiter . $name;
       if (is_dir($pathname) && !ereg("\.\.?",$pathname)) {
          if (isset($flap_out[$pathname])) {
            $array[$name]=$this->recurs_dir($pathname,$depth+1);
          # ATTENTION: It is IMPORTANT fill the [0] array
          # *after* filling the rest of the array!
          $array[$name][0]=$pathname;
        } else {
          $array[$name]=$pathname;
     closedir($d);
     return($array);
   }
```

```
## FLAPPING IN and OUT
  ## This is used to create an array which includes
  ## all sub-paths which should be showed
  function flapping ($path) {
  GLOBAL $flap_out;
    if ($path) {
       if (is_dir($path)) {
         if (isset($flap_out[$path])) {
           unset($flap_out[$path]);
         } else {
           $flap_out[$path]=$name;
    }
$t= new dir Tree;
$t->flapping($val); ## $val is given by GET-method, see *tree()-functions
$t->build tree();
$t->go_through_tree();
print $t->outp;
```

With this code it is very easy to flap in and out whole parts of the tree. Send the path via GET-method and put this path in flapping(). The whole \$flap_out-array must be persistent (e.g. via _s_e_s_s_i_o_n). Perhaps you can program a garbage collection, which will look into \$flap_out and check for paths that already exist?

55..66..55.. KKnnoowwnn BBuuggss // TTiippss

There is one known bug: If a name of a subpath contains the \$delimiter-string. This cannot be solved correctly and you have to look for it when you create the tree.

The same thing is with the value [0] (zero) of a sub-array. This element is always used as the attribute of the parent element.

A good tip: when you build your tree recursively then the [0]-index must be filled _a_f_t_e_r the subtree is returned from recursive call. See in the example above what I mean. I think it's a PHP3 specialty.

Also it is possible that not every name could be inserted into the associate index-field (Control-chars etc.), but this is untested.

55..77.. SSTTRRIINNGGSS22 ffuunnccttiioonn sseett

This is a set of functions, which are used very often by me.

They are so easy, that I now stop describing and simply insert the code. Perhaps the next revision of this set I will replace it with a better description:

```
<?php
##
## Strings2-Functions
##
## Copyright (c) 1998-2000 Alex 'SSilk' Aulbach
##
## These Functions are very practical and if I could program
## C a little bit better it will be placed directly in PHP3.
## But I can't...:-}
##</pre>
```

```
##
## Have you ever worried about such constructs like
    echo ($faxnumber) ? sprintf("Fax: %s",$faxnumber) : "";
##
## This functionset could help you to replace those constructs by
    p_iftrue($faxnumber,"Fax: %s");
## which is nearly the half of typing and looks more clear and solves
## an error if $faxnumber is unset.
##
function o iftrue ($val,$str) {
    if (isset($val) && $val) {
         return(sprintf($str,$val));
function p_iftrue ($val,$str) {
    print o_iftrue($val,$str);
}
##
## Output "One or More"
## This function is good if you want to differ a output by number:
## e.g. o_1or2(q-num_rows)
           "Found only one matching record",
##
           "Found %s records");
##
## Will output if num_rows() is 1: "Found only one matching record"
                    200: "Found 200 records"
##
##
## if $val is empty() or "" a blank string will be returned!
function o_1or2 ($val,$str1,$str2) {
    if (isset($val) && $val) {
         if (1==$val) {
              return(sprintf($str1,$val));
         } else {
              return(sprintf($str2,$val));
     } else {
         return(false);
function p_1or2 ($val,$str1,$str2) {
    print o_1or2 ($val,$str1,$str2);
## This is for the case, that you want to output something
## if $val is false e.g.
## p_0or1($faxnumber,"THERE IS NO FAXNUMBER","Faxnumber: %s");
##
function o_0or1 ($val,$str1,$str2) {
    if (empty($val) || !$val) {
```

```
if (isset($val)) {
               return(sprintf($str1,$val));
          } else {
               return($str1);
          }
     } else {
          return(sprintf($str2,$val));
function p 0or1 ($val,$str1,$str2) {
     print o_0or1 ($val,$str1,$str2);
}
##
## Replaces all blank-chars with
## This function is used, when you are not willing to let the browser
## break your lines an can be used instead of <NOBR>-Tag
## as very compatible replacement
##
## can also be replaced by a true whitespace which has in
## ISO-latin-1 the code 160
function o_nonbsp ($val) {
     return(ereg_replace("[[:blank:]\n\r]"," ",$val));
function p_nonbsp ($val) {
     print o_nonbsp($val);
?>
```

66.. AAcckknnoowwlleeddggmmeennttss

The initial idea on how to do serialization was contributed by KH Wild to the php3 mailing list. It was limited to serializing arrays of at most three dimensions, though. We worked on his solution, improving it to arrays of arbitrary depth and later rewrote the function from scratch, turning it upside down. Our new serialization code can now handle any first order data type available to PHP and is easily extensible. It is also encapsulated in a class, keeping the name space clean. While we were at it, we made session cookies more secure by not using uniquid() directly, but a md5() hash of uniqid().

Cameron Taggart and Guarneri Carmelo contributed ODBC support. Szandor van Verseveld contributed PostgreSQL support. Scott McMullan contributed some nice ideas for cleanup and is working on Sybase support. Sascha Schumann contributed much time developing and extending PHPLIB, including but not limited to mSQL/Sybase support, general storage container support, shared memory and LDAP support.

Alexander Aulbach submitted his Tree class. Jay Bloodworth contributed his excellent OOH Forms library for form generation and input validation.

A lot of people provided helpful hints and occasionally patches. Please see the file CREDITS for a complete list of contributors, testers and inspirations.



1. Quick Start

The Quick Start chapter tries to give you a ten-minute introduction to PHPLIB installation, outlines a few simple testing procedures and closes with an overview of PHPLIB features.

1.1 License

PHPLIB consists of the files in this directory and all its subdirectories. It is made available as free software under the LIBRARY GNU General Public license, as spelled out in the file COPYING in this directory. Also, it is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the license for more details.

1.2 Target Group and Prerequisites

PHPLIB targets the PHP application developer. You need to have good knowledge of the PHP language, at least basic SQL database knowhow and at least basic knowledge on how to operate your web server to be able to use the library.

The library will help you to write medium to large sized data-driven web applications. "Medium to large sized applications" are applications that consist of multiple database queries, have to generate tables from database data, need a user interface that generates SQL queries or need a comfortable and user-friendly way to protect pages or functionality on pages. "Data-driven" applications are applications that make use of a supported SQL-database to create HTML content and that use HTML forms to drive database transactions.

To make use of the library you obviously need access to a web server with a working installation of a current PHP interpreter (we recommend 3.0.12 or newer for this release of the library) and access to a supported SQL database (currently, PHPLIB supports MySQL, PostgreSQL, mSQL, Oracle 7 and Oracle 8, Sybase, Microsoft SQL Server and ODBC databases). You need to be able to create and drop database tables in that database and your web server must be able to execute SELECT, INSERT, UPDATE and DELETE statements on these tables.

Throughout this manual, we assume that you are using the MySQL database server. PHPLIB will run with any supported SQL server, but we are using MySQL in the development of PHPLIB.

PHPLIB can be used in conjunction with the CGI version of PHP and with mod_php, integrated into Apache. Usage of the CGI version has an impact on overall speed, because you cannot take advantage of persistent database connection. We recommend the Apache module over the CGI version, although we personally use the CGI version for various reasons (easier to update and can be run with Apache suexec).

PHP 4 is still in beta. We do not support deployment of this library with beta software.

1.3 Quick Guide to Installation

These instructions apply to PHPLIB running with CGI PHP. Most of them are valid for mod_php as well, though. *VERY IMPORTANT NOTE:* This is a quick installation guide to get you started if you have an installation where you control the web server, PHP interpreter and database server completely. They are not suitable for a web hosting setup where you have only limited to no control over the installation. Refer to Chapter 2 of this documentation for the complete installation instructions and troubleshooting information.

Before installing PHPLIB, get your web server up and running and have it executing files with the extension .php3. Check that with a simple <?php phpinfo() ?> script. Make sure the web server accepts index.php3 as well as

index.html as a default file for URLs ending in "/" (Apache: DirectoryIndex index.html index.php3).

Get your MySQL database server up an running. Create an empty database for your application and make sure the owner of your web server processes can access this database with SELECT, INSERT, UPDATE and DELETE access. Don't forget the mysqladmin reload after changing the user and db tables.

Step 1

Create an include directory named php parallel to your web servers document root directory. Do not put the include directory below your web servers document root.

Step 2

Unpack your PHPLIB distribution. Move the contents of the php distribution directory into the php directory you just created.

Step 3

Get to the php3.ini file for your web servers PHP interpreter and update the include_path statement so that it points to that php directory. Update the auto_prepend_file statement so that it points to the prepend.php3 file in that include directory.

If you do not have control over your php3.ini file, you did not read the VERY IMPORTANT NOTE above.

Step 4

Also check that track_vars are enabled and that you have enabled magic_quotes_gpc. While you are at it, you might want to check sendmail_path, if you plan to send mail from your application. It has to be set to /usr/lib/sendmail -t on most UNIX systems to work.

If you do not have control over your php3.ini file, you did not read the VERY IMPORTANT NOTE above.

Step 5

cd into the php include directory. Edit local.inc. In class DB_Example supply the appropriate parameters for your database connection.

Step 6

For this database, run create_database.mysql from the distribution to create active_sessions and auth_user. auth_user will be populated with a sample user named kris with a password test.

Step 7

Move the contents of the pages directory and all its subdirectories into your document root directory.

Step 8

Access the "/" URL of your web server with cookies enabled. If no index.html is present, index.php3 will be displayed. If you reload that page, the number shown must increment. Access your database with the mysql command client and select * from active_sessions. Check that there is a single session record for your browser and see how the text in val changes when you reload the page and select * from active_sessions again. If this works, the session class is functional with cookie mode.

Step 9

Now access showoff.php3. Try to login as kris, password test. Check active_sessions again. You now should have a Example_Session entry (see the name column) and a Example_User entry in your table. Both should increment on reload.

Step 10

Try again with cookies disabled. You should get a new session (the cookie is lost) and you should be able to see your session id as the get parameter part of your URL.

1.4 Using core features of PHPLIB

Many applications don't use PHPLIB's advanced features, but see PHPLIB as a convenient way to protect pages or functionality with passwords. This section covers such core functionality usage of PHPLIB.

Customizing the login screen

Edit loginform.ihtml in the include directory to suit your needs.

Customizing the permission levels

Edit local.inc and change the class <code>Example_Perm</code> to enumerate your permissions. Your users in <code>auth_user</code> must have one or more comma separated permission names from that list. Edit <code>perminvalid.ihtml</code> for a suitable error message.

Creating New Users

Use new_user.php3 from the pages/admin directory of the distribution. If you followed the installation instructions, it should be available under the /admin URL of your web server.

To manually create a user, run print md5(uniqid("some magic string") to get a user id. insert into auth_user values ("that userid", "username", "password", "permissions");.

Creating an unprotected session page

Begin that page with

```
<?php page_open(array("sess" => "Example_Session")); ?>
```

End that page with

```
<?php page_close(); ?>
```

Creating a protected session page

Begin that page with

```
<?php
  page_open(
    array("sess" => "Example_Session",
        "auth" => "Example_Auth",
        "perm" => "Example_Perm"));
  $perm->check("desired protection");
?>
```

and end that page with

```
<?php page_close(); ?>
```

Creating protected functionality

Begin that page with

```
<?php
  page_open(
    array("sess" => "Example_Session",
        "auth" => "Example_Auth",
        "perm" => "Example_Perm"));
?>
```

and end that page with

```
<?php page_close(); ?>
```

Enclose the protected functionality in

```
<?php
  if ($perm->have_perm("desired protection")):
?>
Put protected HTML or PHP here
<?php
  endif
?>
```

Note: desired protection is any combination of permissions from Example_Perm. Using the default values from Example_Perm, "user, author" or "admin" are all valid sample values. A user can access a page, if that user has all permissions that are being requested in a \$perm->check() or \$perm->have_perm() call.

Note: Users can have multiple permission in their perms column of auth_user. A user with perms "user, author, editor" can access all pages requesting any combination of these permissions.

Note: Don't use spaces. "user, author, editor" works. "user, author, editor" does not.

Note: If \$auth->auth["uid"] is set on a protected page and if (time < auth->auth["exp"]), then and only then the authentication is valid. You may then use \$auth->auth["uname"] as the user name, \$auth->auth["uid"] as a unique user id and \$auth->auth["perm"] for the current permissions of that user. Actually, you never want to touch \$auth->auth["perm"] manually, but use \$perm->have_perm("...") for that.

Getting a grip on PHPLIB

Read on. Then read the source. Read it again - Session->serialize() and Auth->start() are ugly. Get a CVS account. Contribute. Become famous. Buy a ferrari.

Note: You want to understand what registered variables are. You want to understand in what order form variables and session variables are imported into your page. You want to understand how to copy values from form values into session values without killing yourself. You do not want to make form variables persistent, ever. Then you will live happily thereafter...

1.5 Testing

These instructions apply to PHPLIB running with CGI PHP. Most of them is valid for mod_php as well, though. This section offers an incremental approach to find installation problems, should the above installation process fail.

We do have a support mailing list available under the address phplib@lists.netuse.de. To subscribe to the list, send the command subscribe to the address phplib-request@lists.netuse.de.

Checking that the web server is up and running

Make sure your web server is up and serving the virtual host you just set up. To do this, construct a small file test1.html in your DocumentRoot and access test1.html through your web server.

Checking that the web server is executing CGI programs

Make sure your web server is up and does run CGI. Check the current directory, the UID/GID it is running programs under and have a look at the environment variables. Install the shell script

```
#! /bin/sh --
echo "Content-Type: text/plain"
echo
id
echo
pwd
echo
echo
env | sort
echo
```

in your cgi directory under the name of cgi-test and in your document root under the name of cgi-test.cgi. Make it executable. Try to access /cgi/cgi-test?parl=one&par2=two and /cgi-test.cgi? parl=one&par2=two and check the output. What UID/GID are you running under, what is the output of pwd and what environment variables are set? What does QUERY_STRING look like? What does the PATH variable look like, what does the LD_LIBRARY_PATH variable look like and are all libraries needed by PHP accessible to PHP running in the CGI environment (Check by running the Unix 1dd command on PHP).

In particular, if you built Oracle support into PHP and linked libelntsh dynamically: Can it be loaded from the CGI environment? If not, PHP will not come up later in the next step.

Checking that the PHP interpreter is running (Assuming CGI PHP)

Copy your PHP binary into the cgi binary directory (which should NOT be below DocumentRoot!) and make it executable. Copy php3.ini into the same directory. In DocumentRoot, create a test2.php3 and put <?php phpinfo() ?> into it.

Are you running Apache? Add

```
Action php3-script /cgi/php
AddHandler php3-script .php3
DirectoryIndex index.php3 index.html index.htm
FancyIndexing on
```

to your config. This will map all requests to files ending in .php3 to the php3-script handler and define /cgi/php as the URL handling php3-script requests internally.

Request /test2.php3 and see that it is being executed. Make changes to your php3.ini (preferable some color

definitions) and reload. Are they reflected in the output of phpinfo()? If not, your php3.ini is not being found and your are having a problem. Recompile with proper settings.

Check the output of <code>phpinfo()</code> carefully! Is your PHP version current (We have tested and developed this release with PHP 3.0.12)? Are your database interfaces present in the output of <code>phpinfo()</code>? If not, recompile again.

Can you access /test2.php3 under the URL /cgi/php/test2.php3 as well? If so, you did not compile your PHP interpreter with --enable-force-cgi-redirect. PHPLIB will not work with this interpreter. Recompile with the switch being set.

PHP interpreter (Assuming mod_php)

Assuming your server is already correctly setup (don't forget to activate the PHP lines in srm.conf!), enter the following file and save it as test2.php3 under your DocumentRoot.

```
<? phpinfo() ?>
```

If you access this using a web browser now, it should spit out much info about PHP, Apache and its environment.

Checking PHPLIB inclusion

Does you PHP include PHPLIB properly? Check your php3.ini file. It must include the following settings:

```
include_path = pathname to directory with all the .inc files
auto_prepend_file = path to prepend.php3
track_vars = On
```

It should contain the following settings, too:

```
magic_quotes_gpc = On
```

If PHPLIB is included properly by your setup, the following page will execute without errors:

```
<?php
$db = new DB_Example;
print "It works without error messages.<br>\n";
?>
```

Checking database connectivity

PHPLIB installation requires that you adapt local.inc properly. Particularly, the provided class DB_Example must be customized for your database connection. Test that your web server can access the database with the following page:

```
<?php
include("table.inc"); // requires include_path to be functioning

$db = new DB_Example;
$db->query("select * from auth_user");

$t = new Table;
$t->heading = "on";
```

```
$t->show_result($db);
?>
```

When executing properly, this page will show you the user entry for kris, password test, permissions admin from the auth_user table. If this does not happen, your DB_Example definition in local.inc is broken.

Checking that sessions work

Access the page /index.php3 that has been provided with the distribution. This page will try to set a cookie in your browser. Allow that cookie to be set.

The page will display a headline with a counter. Reload that page. The counter must increment. If not, either your browser cannot deal properly with cookies or PHPLIB cannot properly read or write the table active_sessions in your database. Check that the cookie is being set by viewing the output of phpinfo() (the fourth table will report the cookie and other per-call data). Check your database permissions with your database command line interface.

Checking that Authentication works

Try loading /showoff.php3 that has been provided with the distribution. This page will require a login. Login as kris, using a password of test. If the login is successful, you will see the per-session counter and a per-user counter again. Reload that page: The counters must increment.

If you can't login, you probably have a problem with cookies. Check again that your browser accepts and sends session cookies. Another problem may be access to the auth_user table. You must be able to SELECT on that table and there must be at an entry for the user you are trying to login.

Next Previous Contents

2. Overview and Installation

The following sections discuss the installation, verification and layout of PHPLIB: How to install PHPLIB? Which functionality and class definitions are contained in which files? How do you layout a web server with PHPLIB installed? Which installation options are available and how do these affect performance?

2.1 Files, classes and functions

PHPLIB contains a set of core classes and functions that offer session tracking, per-session and per-user persistent variables, user authentication and permission checking. Building upon this core functionality, PHPLIB offers a set of commonly needed "background" classes and a set of "HTML widgets", classes that allow you to quickly generate HTML based user interfaces.

All PHPLIB definitions are designed that you don't need to change any of these files. Your customization of PHPLIB can be contained in two or three files, depending on the setup: local.inc, setup.inc and, in some cases, prepend.php3. You NEVER need to change any other file with PHPLIB. Details are outlined below.

Customization

The following three files are the only files from PHPLIB that require changes in normal PHPLIB applications.

Application configuration in local.inc:

Your application will almost certainly not work with the default values supplied by the above classes. You are supposed to extend the classes described below as you see fit.

In your subclasses, you only have to specify what is different in your application. These are things like database host names, database names, table names and username/password combinations. You need to provide login screen definitions (HTML) and user validation functions (SQL) to make the example work.

The distribution provides a local.inc to illustrate a typical setup. These definitions are also needed to get the administration and testing scripts provided with the distribution to run.

The file is required and you must change it for your setup.

Application setup in setup.inc:

The Session class provides the ability to execute initialization code at session setup. See the class description for instructions on how to set this up.

Per convention, we store such code in setup.inc in the include directory. The code is being executed whenever a new user connection to out application and a new session is started.

The file is optional. No default is provided.

Selection of automatically loaded classes in prepend.php3

The file prepend.php3 determines which code is being loaded for all PHP3 interpreted pages. Normally, we include the class definitions for all core classes in this file: db_mysql.inc, session.inc, auth.inc, perm.inc, user.inc, then your local customizations from local.inc and the page management functions from page.inc.

You must change prepend.php3 to reflect the database interface that you are using: Change the require statement for db_mysql.inc appropriately.

If you are not using some core features from PHPLIB in your application or if you want some other features to be present on all your pages, you can delete or add require statements for their respective include files here.

The file is required. You must change it for your setup, unless you are using MySQL.

Core functionality

The following files are included from prepend.php3 and provide definitions for the core classes of PHPLIB. We recommend that you always include all of them, as they are a tightly integrated set of classes with many dependencies among them.

Class DB_sql defined in exactly one of db_mysql.inc, db_msql.inc, db_pgsql.inc, db_odbc.inc, db_sybase.inc, db_mssql.inc, db_oracle.inc or db_oci8.inc:

A database access class for your database server. PHPLIB depends on the presence of a SQL database server. Depending on the type of your database server, you have to select the appropriate include file. The file contains the definition of a class <code>DB_Sql</code> suitable for your database server.

The class manages a database connection (connection setup is implicit) and result memory is managed automatically.

An independent class.

Class Session defined in session.inc:

Manages an arbitrary amount of arbitrarily named session variables of scalar, array and object types (Object support requires that you implement two instance variables in your classes). Tracks sessions via cookies or a get-variable appended to each URL.

Depends on DB_Sql.

Class Auth defined in auth.inc:

Manages session authentication. Sessions are authenticated against usernames and passwords in a database. Authentication can be time limited.

Depends on Session and DB_Sql.

Class Perm defined in perm.inc:

Manages permission checks on authenticated session pages. Protected pages are only accessible to users with the specified rights.

Depends on Auth, Session and DB_Sql.

Class User defined in user.inc:

Manages user dependent variables. Unlike session variables these are bound to a user id, not to a session id. They are persistent over multiple sessions, but are only available after a user has been authenticated.

Depends on Auth, Session and DB Sql, extension of Session.

functions page_open() and page_close() defined in page.inc:

Setup and Shutdown functions, must be present on any session page.

Depend on Session.

Extended functionality

The extended functionality classes offer GUI-less background features that are commonly needed in HTML-applications. They may make use of core functionality (indicated for each class below).

Cart in cart.inc:

Manages a simple shopping cart. Items can be put into the cart, taken out of the cart and the carts contents can be enumerated.

Depends on Session to be useful. Requires that you add the statement require ("cart.inc") to prepend.php3.

Template in template.inc:

Manages templates and variable replacement. Templates can be stored in files. They are loaded on demand and variables are replaced in these files.

An independent class. Requires that you add the statement require("template.inc") to prepend.php3 or that you include it manually on each page where you want to use it.

HTML widgets

HTML widgets are classes that generate some HTML-code (often forms or tables) to display GUI-elements. We try to provide functionality commonly used in applications, in a way that the actual look of the GUI-elements can be easily customized.

CSV_Table ${f in}$ csv_table.inc:

Creates a dump of a two dimensional array or a query result in CSV format, suitable for loading into a database or a spreadsheet program.

Depends on Table, extension of Table.

Sql_Query in sql_query.inc:

Create a selection widget that allows a user to choose arbitrary conditions on one or more table columns. SQL is being created from these selections that can be used in the where-clause of a larger SQL select statement.

Depends on Session and DB_Sql. Requires that you add the statement require("sqlquery.inc") to prepend.php3.

Table in table.inc:

Creates HTML tables from two dimensional arrays or from database query results. The class can either filter out the desired columns from an array or you can explicitly name which columns to show. A heading can be turned on if desired. All generated HTML elements are tagged with a classname you specify for stylesheet support, if needed. When used in a form tag, each table row can be prefixed with a checkbox input element to allow for row selection.

An independent class.

Form in oohforms.inc:

Creates HTML forms from feature->value arrays. This provides a single syntax for creating all of the different types of form elements. The class provides easy access to Javascript and server side validation, and supports 'freezing' some or all of the form elements to display static data. In addition, the library relies on object oriented implementations for the various form elements and these can easily be extended and customized.

An independent class.

2.2 Downloading and unpacking the distribution

The base library is supplied at the PHP Base Library download location. Two different formats are provided: A tar.gz version and a shar version.

If you are on a windows system, you can use phplib.tar.gz, if you have WinZIP installed. Current versions of WinZIP know how to handle compressed tar archives. The uncompressed files may be installed on your windows system or transferred to your Unix system.

If you can't handle binary files, you may download phplib.shar. This is a pure ASCII file containing a self extracting shell script. Just save the file, make it executable and feed it to your Unix shell (for example, by typing sh phplib.shar).

The <u>PHPLIB support mailing list</u> is available should you run into problems with the library. To subscribe send the command subscribe to the mailing list subscription address.

2.3 Requirements and things to check for

Interpreter requirements

The PHP base library requires a working web server with CGI capability and the CGI version of PHP 3.0.12 or higher installed. Alternatively mod_php can be used. Lower versions of PHP do not work at all: The session class uses the base64_encode() and base64_decode() functions which are known to be buggy in lower versions (up to 3.0.7) of the library. Also, the OOH Forms classes are using constructor syntax, which has been introduced into the PHP language in 3.0.5 and later versions. An issue with the \$PHP_SELF variable and CGI PHP has been resolved with version 3.0.5 and later. Perl regular expression functions are being used in the Template class and these are not really avilable up to 3.0.12.

Note: If you are using CGI PHP, it *must* have been compiled with the --enable-force-cgi-redirect switch for \$PHP_SELF to have the correct value.

Basically, if PHP_SELF is the exact local part of your \$URL, all is well. If it instead contains the modified URL with /your cgi-bin/php prefixed, you have a buggy version of CGI PHP. Either upgrade your version of PHP or replace all occurrences of \$PHP_SELF with \$PATH_INFO in PHPLIB.

Note: PHPLIB requires that you have track vars compiled in and enabled.

Note: PHPLIB does not require short_open_tag to be enabled. The library always uses <?php as the PHP command introducer.

Note: PHPLIB does not require magic_quotes_gpc to be enabled. The library always uses addslashes() when necessary.

Database requirements

The PHP base library requires a database connection in the default setup for storage of session variables, but this can be circumvented by selection another storage container type at installation time. Currently, storage containers are available for SQL databases (the default), SQL databases with limited string length (ct_split_sql.inc), System V shared memory (requires a PHP interpreter with SYSVSHM and SYSVSEM support), LDAP servers (requires a PHP interpreter with LDAP support), flat files, and DBM files.

Using SQL, currently MySQL is fully supported and PostgreSQL, mSQL, Sybase, Microsoft SQL Server, ODBC and Oracle have limited support (the limitation is only relevant if you intend to access metadata information, i.e. table definitions and the like). Database interfaces are not difficult to write and you can easily write your own interface.

You need a database server connection with select, insert, update and delete privileges from your CGI environment. You need create and drop privileges from an administrative account outside your CGI environment as well.

PHPLIB core functionality requires two tables as part of your application table name space: active_sessions (select, insert, update and delete privilege required for the application user) and auth_user (select privilege required for the application user insert, update and delete privilege required for the application user if user management is to be done from within the application).

Extended functionality may require additional tables.

Name space requirements

PHPLIB tries to be as name space neutral as possible with its core features. Is issues no HTML by default and it occupies only few names in the global name space. These are the class names for the classes defined: DB_Sql, DB_SAM, CT_Sql, Session, Auth, Perm, User. Additionally, the classnames DB_Example, Example_CT_Sql, Example_Session, Example_Auth, Example_Derm and Example_User are defined by the sample setup in local.inc, but these names can and shall be customized by the application developer. PHPLIB defines the function names page_open(), page_close, sess_load() and sess_save() for the page management functions. The global variable \$_PHPLIB (a hash) is taken. Only if page_open() is being used, globals are defined by the library by default, but one global for each "feature" requested in the page_open() statement is taken. These are at most \$sess, \$user, \$auth and \$perm.

Including extension functionality or HTML widgets may occupy additional classnames, function names or variables in the global name space.

Year 2000 compliance statement

PHPLIB uses date fields within the column changed in the table active_sessions in your database. The changed field is used in garbage collection, that is, to clean out abandoned sessions. The date field is a 14 character field of the format yyyymmddhhmmss, that is, the date field has four digit years and will cope properly with the new millennium.

PHPLIB sets cookies in the client browser. These cookies by default have session lifetime, that is, they do not expire but are not written to disk. Date calculations are not involved.

It is possible to have PHPLIB set cookies with a limited lifetime by defining the \$lifetime slot of the Session class. If this is done, it depends on the date handling of the client browser and client operating system, if the result is Y2K compliant. There are known issues with longterm cookies and any browser on MS-DOS/Windows 3.11 systems.

PHPLIB does some date arithmetic internally that involves mktime() and date() functions of the PHP3 language and Unix time_t data types. The signed 32 bit Unix time_t data type counts seconds since 01-Jan-1970 Midnight GMT and

will overflow sometime in the year 2038.

PHPLIB itself will function up to 2038 and longer, if the Unix time_t is being extended in time. PHPLIB does not protect you from date and Y2K issues in your PHPLIB application, the PHP3 interpreter, the server operating system oder server software, the client browser, the client operating system or other parts of your installation.

2.4 Installation procedure

mod_php note: The following instructions apply to the CGI version of PHP as well as to the module version. If you are working with mod_php, you must restart your web server to force a reload of the php3.ini file, though.

If you are using mod_php, you have additional configuration options: See the section below on using PHPLIB mit mod_php.

Library Setup

Create a directory php next to your cgi:

```
/home/www/servers/phplib.netuse.de/pages <- document root
cgi <- php binary
php <- includes and prepends
```

Make this php directory your php include directory: Put include_path =

/home/www/servers/phplib.netuse.de/php into cgi/php3.ini. If you already have an include path defined in your setup, add the PHPLIB include path to the existing include path using the separator character applicable for your operating system (":" on UNIX, ";" on Windows). Defining an include path will not actually include code on your pages; it only tells the PHP interpreter in which directories to look for files referenced in require() and include() statements.

Into the php directory go all the files from the php directory of the distribution. Into documentroot, put all the files and directories from the pages directory of the distribution.

Have documentation.txt handy and read it.

Database access with MySQL

The following information applies to MySQL only. No information is provided for other database servers as of now. You are encouraged to copy this section, adapt it for your database server and submit it to the authors. It will be included in further releases of PHPLIB.

Edit prepend.php3. Change the first require() statement to require("db_mysql.inc");. This will include the MySQL database interface (Interfaces for other databases are provided in db_<databasename>.inc. The require() statement has to be adapted to reflect this).

Assuming your database server is named database.netuse.de and your CGI user is webuser and you are accessing the database myapp, do

```
mysql -h database -u webuser myapp
```

If it does not work, connect your database as administrator and create the proper mysql access permissions. Adapt and run create_database.mysql from the stuff subdirectory of the distribution to create the databases active_sessions and auth_user as well as the sample user kris with password test. Try again to connect like shown above. Can you do select * from active sessions? and insert into active sessions

```
values ("1", "2", "3", "") as well as delete from active_sessions? Can you select * from auth_user?
```

Note: Additional database creation scripts are provided for several different databases in the stuff directory of the distribution.

Merging the library with your PHP files

Decide if you want to use include or auto_prepend_file. We do use auto_prepend_file here and we add the statement auto_prepend_file = /home/www/servers/phplib.netuse.de/php/prepend.php3 to our php3.ini.

Not all classes are included/required by prepend.php3, only core functionality files are: db_xxx.inc, ct_sql.inc, session.inc, auth.inc, perm.inc, user.inc, local.inc and page.inc. The library provides other, less essential classes that can be included manually on a page-by-page basis. Some classes make themselves persistent, if used. These classes require that you include their definitions in the prepend.php3 file where indicated to function correctly. See the usage instructions for these classes for details.

Having done this, access /index.php3. The counter should increment when that page is being reloaded. Also, checking active_sessions in the database should reflect that session.

Subscribe for support

Subscribe to the mailing list phplib@lists.netuse.de. Do so by sending a mail body of subscribe to phplib_request@lists.netuse.de and follow instructions. Share your experiences.

2.5 Using include() instead of auto_prepend_file=

If you do not want to use auto_prepend_file to load the PHPLIB core functionality, you can load the class definitions for the core manually on each page that requires them.

You will have to define a valid include_path=-statement in your php3.ini file as outlined previously to reflect the location of the *.inc files. Then, all core functionality can be loaded with include("prepend.php3") as the first statement at the top of each page.

To further optimize performance, you can minimize the contents of the prepend file, if you do not need all core functionality. You may leave out auth.inc, perm.inc and user.inc, if you do not require these features (note that there are dependencies among these classes!).

2.6 PHPLIB with mod_php (Apache module)

Installing PHPLIB onto a web server that has PHP3 as a module (actually Apache) mainly differs in where you can set up runtime settings for PHP3 itself. PHP3 can be compiled with a wealth of parameters (see the PHP section in phpinfo()), most of which can get overridden by the php3.ini file. The location of this file is shows as part of the output of phpinfo().

With PHP3 as a module you have a wider choice on placing these settings: they are overridden, in this order, by what is defined in httpd.conf and in your per-directory .htaccess file. Directives in these files are identical to their php3.ini brothers, but are prefixed with php_ to avoid clashes with Apache configuration keywords. Also, as they are Apache configuration keywords, they have no equals ("=") sign in them. If x=y is a configuration directive from php3.ini, you should be using php3_x y within the Apache configuration instead. That is, you should prepend php3_ to the keyword and omit the equals sign. If you misspell a configuration directive, you will get an error 500 from your webserver and find more details about the error in the logfile you configured with ErrorLog in your webserver setup.

Example: If below we talk about setting in your php3.ini the configuration

```
include_path = "/bla"
```

mod_php users may alternatively configure in their httpd.conf the following:

```
<Directory /home/www/servers/phplib.netuse.de/pages>
php3_include_path "/bla"
</Directory>
```

Of special interest to PHPLIB users are the following directives:

```
; Data Handling ;
; magic quotes for incoming
magic_quotes_gpc = Off
                            ; GET/POST/Cookie data
magic_quotes_runtime = Off; magic quotes for runtime-generated data,
                            ; e.g. data from SQL, from exec(), etc.
magic_quotes_sybase = Off ; Use Sybase-style magic quotes
                            ; (escape ' with '' instead of \'); enable $PHP_GET_VARS[], $PHP_POST_VARS[]; and $PHP_COOKIE_VARS[] arrays
track vars = On
; automatically add files before or after any PHP 3.0 document
auto_prepend_file = (add path to prepend.php3 here)
auto_append_file
; Paths and Directories ;
= (add path to the directory with all .inc files)
include_path
```

All of this comes very handy when you have multiple virtual hosts (e.g. you are an ISP). In this case you can comfortably place the php3 directives in the <VirtualHost> block or in an .htaccess file in the client directory.

Next Previous Contents

3. Core Functionality

Each class contains instance variables and instance methods. Some of these variables and methods are available for customization, some are internal to the classes themselves. All are documented, but tampering with internal variables and methods is not supported. Internal interfaces are subject to change without notice from one version of the library to another.

This section covers PHPLIB core functionality in reference form. Classes are presented in order of dependency, though, because the core structure is easier understood in this order. You will need to understand the complete core structure to successfully use all of PHPLIB's features.

3.1 DB_Sql

DB_Sql is used by CT_Sql and Auth to access a SQL database. You are encouraged to use it directly, too.

Instance variables

Accessible instance variables.

Host	Host where your SQL server is running on.	
Database	Name of database or instance to use on that server.	
User	The username used in connections attempts.	
Password	The password used in connection attempts.	
Row	Number of the current result row starting at 0.	
Errno	Integer: error number of last database operation.	
Error	String: error message of last database operation.	
Halt_On_Error	One of "yes", "no", "report". If set to "yes" (the default), the database interface will report any errors and haltthe program. If set to "report", the database interface will still reportany errors, but continue reporting "false" back to the application, withErrno and Error set appropriately. If set to "no", the database interfacewill not report any errors, but silently report "false" back to application, with Errno and Error set appropriately.	
Auto_Free	Boolean: In some DB interfaces a flag for earlyresult memory release.	
Debug	Boolean: If set, the database class will output all queriesand additional diagnostic output.	
type	Contant string: The name of the database interface, e.g. "mysql" or "oracle"	
revision	Contant version string: The version of the database API(e.g. 1.2), NOT the CVS revision of the file implementing the API.Sql_Table	string: The name of the table used by thenextid() API function.

Internal instance variables.

Record	In some DB interfaces a hash of the current table result row.
Link_ID	SQL Link ID.
Query_ID	SQL Result ID.

Instance methods

Accessible instance methods

DB Sql(\$query = "")

Constructor. When creating an instance, you may optionally supply a query string.

```
$db = new DB_Sql_Subclass("select * from mytable");)
```

query(\$query_string)

query_string is a SQL statement that is sent to the database. After sending the statement, Error and Errno are updated. If the query is syntactically incorrect (no valid result id is being produced), halt() is called with a meaningful error message.

If there is no active link to the database, a pronnect() is made using the information from the Host, Database, User and Password instance variables.

Returns the result of the query() statement, which is guaranteed to be a valid result id (or false, if Halt_On_Error isn't "yes").

next record()

next_record() advances the cursor through the current query result and updates the Record, Row, Errno and Error instance variables.

Returns true, if there is a new result record. Returns false, if done with the current result set. If Auto_Free is true, free_result() is called automatically before false is returned.

num_rows(), nf()

Returns the number of rows returned by the current SELECT query.

Note: This information is not available in all database interfaces. Some of the more advanced databases begin to return query results asynchronously while the backend is still appending result rows. In such environments the complete size of the result set is never known.

You should duplicate your WHERE clause of the query in such environments and ask for the COUNT(*). This will be less inefficient as it seems as the query path and query result have been cached by the database.

affected_rows()

Returns the number of rows affected by the current INSERT, UPDATE or DELETE query.

num fields()

Returns the number of columns returned by the current query.

np()

Prints the number of rows returned by the current query.

f(\$field)

Identical to accessing Record[\$field].

p(\$field)

Identical to printing Record[\$field].

haltmsg(\$msg)

This function is called by halt() and will actually print the database error message. You may override this method in your subclass of DB_Sql and format the error message to be consistent with the layout of the rest of your application. You may also add additional error handling such as informing the application operator by mail that a database error has occured.

seek(\$pos)

Positions the Row pointer within the result set. Useful for reading the same result set twice or otherwise jumping around within the result. \$pos is not checked in any way for validity.

Note: If Auto_Free is true, seek() may not be useable, because the result set has already been free'ed when next_record() when behind the last record of the result set.

Note: Not all database interfaces provide a cursor that is capable of seeking. This function will be unavailable in such environments.

link_id()

This function will return the current link ID, as returned by the pconnect() executed internally by the database class.

You should not need this information.

query_id()

This function will return the current result ID, as returned by the query() executed internally by the database class.

You should not need this information.

metadata(\$table = "", \$full = false)

\$table is a SQL table name in the current database. The function returns an array of hashes indexed on the (0 based) column number of \$table. Each hash is indexed by table (table of which this column is part of), name (name of this column), type (column data type), len (column width) and flags (database specific column flags, if applicable) with one row per table column. Each row describes a column in your table.

The data returned by metadata() is suitable for passing it to the Table class. If you specify the full parameter, an additional column meta is added, which is indexed by field name and returns the field number of that name. Also, a column num fields is added, containing the width of the table.

If \$table is omitted, the function returns metadata on the result of the last executed query. *Note:* This is currently implemented only for the MySQL interface. You are encouraged to implement this feature for other interfaces.

NOTE: At the moment, the PostgreSQL and ODBC interface only report the table, name and type data reliably. You are encouraged to fix this.

table_names()

Returns an array with table name and tablespace name.

```
table name : $return[$i]["table_name"]
tablespace_name : $return[$i]["tablespace_name"]
```

Tables are from \$i=0 to last table;

Implemented in db_oracle.inc,db_oci8.inc,db_mysql.inc,db_pgsql.inc

nextid(\$sequence_name)

This function will return a sequence number from the sequence named by \$sequence_name. This number is guaranteed to be obtained in an atomic manner and can be used as a primary key.

Internal instance methods

connect()

Used internally to generate a Link_ID, if necessary. Link creation is implicit, there is no need to call connect() manually, ever.

halt(\$msg)

Used by query() if the initial database connection cannot be made or the target database does not exist. Depending on the setting of Halt_On_Error, this method will call haltmsg() to report the error.

free()

Used internally by next_record() to free the result set, if so configured.

Example

Use a subclass to provide the appropriate parameters for a database connect. You may overwrite halt() to customize the error message, although a sensible default is provided.

```
printf("exact error message.<br>\n");
}
```

Use an instance of the subclass to manage your queries:

Additional information about database connections

PHP reuses connections, if possible. When a connection is being made to the same Host with the same Username and Password as an existing connection, no second connection is being made by PHP. Instead the existing connection is returned to the caller. This is true for both, the *_connect() and *_pconnect() calls of all PHP database interfaces.

This has implications for MySQL users: Never use the MySQL "use" command to change the current database. If you do, session management will fail to operate properly. Instead, create all PHPLIB tables as part of your application.

Some databases (for example Oracle) have very expensive connect() operations. For these databases, performance is dramatically improved if you switch from CGI PHP to mod_php. This is, because PHPLIB uses the "*_pconnect()" method to connect to your database. In mod_php, the database connection is kept around by the web server process after the page has been processed and is reused if a further connect requires a connection with the same Host/Username/Password pattern.

This means that there will be at most "number of web server processes" times "number of Host/Username/Password-combinations" many simultaneous connections to your database server. Keep that in mind when planning licenses and server load. Using CGI PHP will probably reduce the number of concurrent connects to your database server at the expense of connection setup time. For database servers where connection setup time is negligible (MySQL for example) this is a viable solution (don't try it with Oracle) though.

Using nextid()

The nextid() function can be used to obtain a sequence number which can be used as a primary key. The function manages an arbitrary number of named sequences, you have to provide the name of a sequence upon call.

3.2 Page Management

Accessible Functions

Page Management currently consists a collection of functions:

page_open(array("feature" => "classname"))

This function is to be called with an array of page features/classname pairs. Valid features are at the moment:

sess

This page makes use of session variables.

auth

This page uses session authentication. If you specify the auth feature, you MUST specify the sess feature, also.

perm

This page is protected by permissions and only accessible to authenticated users with matching rights. If you specify the perm feature, you MUST specify the auth and sess features, also.

user

This page makes use of user variables. If you specify the user feature, you MUST specify the auth and sess features, also.

Each feature specifies the name of the class that implements that feature, for example

```
page_open(array("sess" => "Shop_Session"));
```

The function creates an instance of <code>Shop_Session</code> as <code>\$sess</code> and initializes it. It also checks feature dependencies. Note that you are expected to provide an implementation of the class <code>Shop_Session</code>. This is usually done in <code>local.inc</code> and usually you do so by extending the provided <code>Session</code> class.

Examples on how to do this is given in the documentation below when the classes are introduced.

page_close()

At the end of your page (after all results have been calculated) you have to call <code>page_close()</code>. This will save all page state, session and user variables into database. Changes to session or user variables after <code>page_close()</code> has been called are not recorded. Currently it is allowed to call <code>page_close()</code> multiple times on a single page (not guaranteed for future versions!). Each time session state will be saved.

Note: This is going to change. When we introduce record locking, it is important that you call page_close() only once per page, because that will implicitly unlock your session record. Also, it is important that you call page_close() as early as possible on a page so that the locking time is kept minimal.

sess_load(array("var" => "classname")

Advanced feature. Some applications have need to manually load data belonging to one or multiple session classes. @@TODO

```
sess_save(array("var" => "classname"))
```

Advanced feature. @@TODO

Example

```
<?php
  page_open(array("sess" => "Shop_Session"));
  $sess->register("s"); // See "Session" below for explanation.
?>
<html>
<h1><?php print ++$s ?></h1>
</html>
<?php page_close(); ?>
```

The "cart" feature is gone

There used to be a feature "cart" for page_open() in versions of PHPLIB up to release-5. The cart has been removed from the core functionality of PHPLIB to keep the library small, maintainable and structured. Consequently the "cart" feature is gone.

The Cart class is still present and exists as an extended feature. You have to include and instantiate your cart manually on that pages that use it, though. See the Cart class for more information.

3.3 CT_Sql

The Session class used to contain a bit of SQL to read and write session data from and to a database. To make sessions database independent, this SQL has been isolated and put in a separate class, CT_Sql. Session now makes all storage accesses through a container class, which may or may not be an SQL container.

Instance variables

Accessible instance variables.

database_table	The name of the database table which should be used
database_class	A classname. CT_Sql uses this class to store and retrieve data

Example

Use a subclass to provide the appropriate parameters to your container. Usually your subclass looks like this:

```
class My_Sql extends CT_Sql {
    var $classname = "My_Sql";
    var $database_table = "active_sessions";
    var $database_class = "DB_Session";
}
```

You can then use My_Sql in class Session. Reference it by putting "My_Sql" in the "that_class" variable.

3.4 CT_Split_Sql

The session class used to contain a bit of SQL to read and write session data from and to a database. To make sessions database independent, Session now makes all storage accesses through a container class. The CT_split_sql container is very similar to CT_sql container, with the difference that if serialized data exceeds a specified amount of bytes, multiple rows will be used to memorized the entire field.

This class is NOT compatible with CT_Sql class, since table layout is different and column names are different in order to avoid reserved words in various database implementation. This uses a DB_Sql like class so you can access all supported databases with this container.

Instance variables

Accessible instance variables.

database_table	The name of the database table which should be used	
database_class	database_class A classname. CT_Sql uses this class to store and retrieve data	
split_length	A number. This specifies the maximum amount of bytessaved in each row of the table.	

Example

Use a subclass to provide the appropriate parameters to your container. Usually your subclass looks like this:

```
class My_Sql extends CT_Split_Sql {
    var $classname = "My_Sql";
    var $database_table = "active_sessions_split";
    var $database_class = "DB_Session";
    var $split_length = 4096;
}
```

You can then use My_Sql in class Session. Reference it by putting "My_Sql" in the "that_class" variable.

3.5 CT_Shm

The Session class used to contain a bit of SQL to read and write session data from and to a database. To make sessions database independent, Session now makes all storage accesses through a container class. To let Session use shared memory as container, you use CT_Shm.

Instance variables

Accessible instance variables.

max_sessions	The maximum number of concurrent sessions supported by this container.
shm_key	The unique (important!) key of the shared memory segment you want to use.
shm_size	The size of the shared memory segment. The size is set, when the segment is accessed for the first time. If you do not use too many session variables, the formula shm_size = max_sessions * 600 should be sufficient.

Example

Use a subclass to provide the appropriate parameters to your container. Usually your subclass looks like this:

```
class My_Shm extends CT_Shm {
    var $classname = "My_Shm";
    var $max_sessions = 500;
    var $shm_key = 0x1234232;
    var $shm_size = 64000;
}
```

You can then use My_Shm in class Session. Reference it by putting "My_Shm" in the "that_class" variable.

3.6 CT_Dbm

The Session class used to contain a bit of SQL to read and write session data from and to a database. To make sessions database independent, Session now makes all storage accesses through a container class. To let Session use a DBM database file as a container, you use CT_Dbm.

Instance variables

Accessible instance variables.

The path to the dbm file (must exist already AND must be writable by the server process)

Example

Use a subclass to provide the appropriate parameters to your container. Usually your subclass looks like this:

```
class My_Dbm extends CT_Dbm {
    var $dbm_file = "data/session.dbm";
}
```

You can then use My_Dbm in class Session. Reference it by putting "My_Dbm" in the "that_class" variable.

3.7 CT_Ldap

The Session class used to contain a bit of SQL to read and write session data from and to a database. To make sessions database independent, Session now makes all storage accesses through a container class. To let Session use a LDAP database as a container, you use CT_Ldap.

Instance variables

Accessible instance variables.

ldap_host	This is the hostname of the LDAP server to connect to	
ldap_port	And this is its port (LDAP default is 389)	
basedn	This is the basedn	
rootdn	This is the rootdn which is required to modify the database	
rootpw	The respective password for rootdn	
objclass	The objectclass for PHPLIB's data	

Example

Use a subclass to provide the appropriate parameters to your container. Usually your subclass looks like this:

```
class My_Ldap extends CT_Ldap {
    var $classname = "My_Ldap";
    var $ldap_host = "localhost";
    var $ldap_port = 389;
    var $basedn = "dc=your-domain, dc=com";
    var $rootdn = "cn=root, dc=your-domain, dc=com";
    var $rootpw = "secret";
    var $objclass = "phplibdata";
}
```

You can then use My_Ldap in class Session. Reference it by putting "My_Ldap" in the "that_class" variable.

3.8 Session

The session class keeps a list of global variable names and provides a set of functions to load and save these variables from and to a data storage container (we will call it container for shortness). The named variables may be scalar variables (strings, integers and floats) or arrays. Objects are handled as well, provided they implement two instance variables naming their class and enumerating their (persistent) slots.

Instance variables

Accessible instance variables.

classname	Serialization helper: The name of this class.
magic	A secret string used in ID creation. Change it!
mode	Mode of Session ID propagation. Either cookie or get.
fallback_mode	Mode of Session ID propagation should \$mode not work. Set \$mode to cookie and \$fallback_mode to get.
lifetime	Lifetime of the session cookie in minutes or 0 to use session cookies.
gc_time	Garbage collection tuning parameter, see below.
gc_probability	Garbage collection tuning parameter, see below.
Hallowcache I	Control caching of session pages. If set to passive (also the default), no cache-control headers are being sent. If set to no, the page is not cached under HTTP/1.1 or HTTP/1.0; if set to public, the page is publically cached under HTTP/1.1 and HTTP/1.0; if set to private, the page is privately cached under HTTP/1.1 and not cached under HTTP/1.0
allowcache_expire	When caching is allowed, the pages can be cached for this many minutes.
that_class	A classname. Session uses this class to store and retrieve data.
auto_init	The file to be loaded on session establishment.
secure_auto_init	Set to 0, if all pages always callpage_close() (This is never the case!).

Internal instance variables.

pt	Internal array of names of persistent variables.
	Flag: If set, auto_init has been executed.
name	A tag (name) for the session type.

id	Id of the current session.
that	Container object instance.

Instance methods

Accessible instance methods

register(\$varname)

Registers a global variable name as a session variable. The name may identify a scalar variable, an array or an object. If an object is to be made persistent, it must have two instance variables:

classname

A string with the name of the objects class.

persistent_slots

An array with the names of all object slots to save.

unregister(\$varname)

Unregisters a global variable name as a session variable. The variable is not deleted, but its value will be lost at the end of a page. It is no longer saved to the database.

is_registered(\$varname)

Returns true if the variable named \$varname is registered with the session, false otherwise.

delete()

Destroy the current session and put_id() the current session id.

After delete() has been executed, all session data has been removed from the database. Also, the session object is unusable on this page. Consequently, page_close() may not be called for this session. Session variables are still available on this page, even after the delete(), but will be lost on the following pages.

In cookie mode, it is possible to page_open() a new session after delete() has been called, if no HTML has been output so far so that the new cookie can be set. If you do this, you can also re-register some of the previous session variables and can call page_close() for the new session. This allows you to change the session on the fly and selectively carry over session data from the previous session.

url(\$url)

Return an URL referencing the current session. If in get mode, the current session id is attached to this URL, else the URL is returned unmodified.

purl(\$url)

A shorthand for print \$this->url(\$url);

self_url()

Return an URL referencing the current page, including PHP_SELF and QUERY_STRING information. If in get

mode, the session id is included.

pself_url()

A shorthand for print \$this->self_url().

hidden_session()

Adds a hidden form element containing the session name and id.

add_query(\$qarray)

Return string to be appended to the current URL for parameters in GET query format. Intended usage is like this:

```
<a href="<<?
$sess->pself_url().$sess->padd_query(array("again"=>"yes"))
?>"> Reload</a> and log in?
```

padd_query(\$qarray)

A shorthand for print \$this-> add query(\$garray).

reimport_get_vars()

When a FORM variable is made persistent, that form variable is imported into PHP, then page_open() is being called and the new variable value is overwritten from the database. The FORM value is lost.

If you had enabled track_vars and were accessing http_get_vars directly, which is recommended, this were not a problem. Some legacy scripts rely on persistent form input variables, though.

These scripts may call the appropriate reimport_x_vars() functions. These functions will re-read the tracked variable arrays and reinitialize the appropriate global variables after session variables have been restored.

Use of this function is discouraged.

reimport_post_vars()

```
See reimport_get_vars().
```

reimport_cookie_vars()

```
See reimport get vars().
```

set_container()

You shall not call this function directly. It is called back by the <code>start()</code> function of <code>Session()</code> during initialization. It is documented so that you can override its implementation in your subclass of <code>Session</code> if you know what you are doing.

This function creates and starts the container class used by this instance of session.

set_tokenname()

You shall not call this function directly. It is called back by the start() function of Session() during

initialization. It is documented so that you can override its implementation in your subclass of Session if you know what you are doing.

This function determines and sets the internal session name.

release_token()

You shall not call this function directly. It is called back by the <code>start()</code> function of <code>Session()</code> during initialization. It is documented so that you can override its implementation in your subclass of <code>Session</code> if you know what you are doing.

This function determines the current method of session propagation and determines if a new session token has to be generated.

put_headers()

You shall not call this function directly. It is called back by the <code>start()</code> function of <code>Session()</code> during initialization. It is documented so that you can override its implementation in your subclass of <code>Session</code> if you know what you are doing.

This function determines which header lines are to be generated by the session, including cache control headers.

Internal instance methods

get id()

See get_id().

get_id(\$id_to_use)

get_id() is used internally to determine a session identifier. Currently, a session identifier is a hex number of 32 characters (128 bits) and it is generated by md5(uniqid(\$this->magic)) to make it hard to guess.

get_id() may be called with an optional session id to use as a parameter. This is useful if you want to change a session id without breaking the session (taking over an old, left over session).

get_id() can be overwritten by a subclass, if you want a different system to create session ids. For example, some applications want to use a constant session id that is not propagated to the client to use a shared pool of persistent variables (a guestbook for example). These applications need locking (to be implemented soon).

put id()

put_id() is used internally to "unuse" a session it. At the moment it deletes the client side cookie and deletes \$HTTP_COOKIE_VAR[\$this->name] for that cookie. The variable \${\$this->name} is *not* deleted.

serialize(\$prefix, &\$str)

serialize() is used internally to append to str all PHP code needed to reconstruct the variable named in prefix.

freeze()

freeze() serializes all register()ed variables and writes the resulting code into the database, tagged with the current session id and the current session name.

thaw()

thaw() loads a set of freeze()ed variables for the current session id and session name out of the database and recreates them.

gc()

The active_sessions table contains one row for each session. That row is uniquely identified by the sid and name values (name is the name of the session class that has written the row). Each time that row is written, the column changed is updated with the current time.

The gc() function deletes all rows that are older than gc_time minutes and have a matching name field. For speed reasons, gc() is not not called every time an update to active_sessions is being made. Instead it is called randomly with a probability of gc_probability.

reimport_any_vars(\$arrayname)

Used to implement the three official reimport functions.

start()

Initialization function, to be called after object instantiation. Calls get_id() to get the current session id, creates a database connection, then calls thaw() to load all session variables. Randomly activates gc(). Checks allowcache to send proper headers to control browser caching.

Example

Use a subclass to provide the appropriate parameters to your session. Usually your subclass looks like this:

```
class My_Session extends Session {
  var $classname = "My_Session"; ## Persistence support

  var $mode = "cookie";
  var $lifetime = 0; ## use session cookies

  ## which container to use
  var $that_class = "Session_sql";
}
```

Remember that you have to provide a DB_Sq1 subclass with the parameters needed to access your database.

Use the page management functions (see above) to use your session subclass. The feature name for session management is sess; provide the name of your session subclass as a parameter to the sess feature:

```
page_open(array("sess" => "My_Session"));
```

Use the register() instance method to register variables as persistent. If \$sess is your session object, use

```
$sess->register("s");
```

to make the global variable \$s persistent. \$s may be a scalar value, an array or an object with persistence support slots.

Do not use the instance methods freeze() and thaw() directly, but use the page management functions instead.

To have some pages cached and others not cached, use multiple instances of the session object. For example, for those pages that should be cached, use a session object instance like

```
class My_Cached_Session extends My_Session {
  ## pages that use this session instance are cached.
  var $allowcache = "private";
}
```

Be careful when using the public cache option. Publically cached pages may be accessible to unauthenticated users. The private cache option prevents unauthenticated access, but is only functional in HTTP/1.1 browsers.

Using "auto_init"

You may define \$sess->auto_init to the name of an include file in your extension of session. Per convention, the name setup.inc is being used.

```
class My_Session extends Session {
  var $classname = "My_Session";
  var $magic = "Calvin+Hobbes";
  var $mode = "cookie";
  var $gc_probability = 5;

  var $auto_init = "setup.inc"; // name of auto_init file.
}
```

Whenever a new session is established, that is, a user without a session id connects to your application, the auto_init file is included and executed exactly once. The file is executed from within the context of the page_open() function, that is, *not* within a global context. To define or access global variables from the auto_init file, you have to global them.

When auto_init is being executed, all features of your page already exist and are available globally. That is, you can safely rely on the existence of the \$sess, \$auth, \$perm and \$user variables, if your application specifies them. *Note* that you cannot in general know which particular page triggered the execution of auto_init, though. If you have some pages that request authentication and others that don't, you cannot rely on the presence of the \$auth object in general, but have to test for it with is_object(\$auth) before accessing it.

The auto_init file is the appropriate place to initialize and register all your session variables. A sample setup.inc may look like this:

Note: If you don't use a fallback_mode and you get users that turn off cookies, these users will force a new session each time they hit any page of your application. Of course this will force inclusion and execution of setup.inc for each page they visit, too. Nothing can be done about this.

Unregistering variables and deleting sessions

To get rid of a persistent variable, call \$sess->unregister() with the name of that variable. The value of the

formerly registered variable is still available after the call to unregister, but the variable is no longer persistent and will be lost at the end of the current page.

To get rid of all session related data including the session record in the database, the current session id and the session cookie in the users browser, call \$sess->delete(). In shopping applications this is commonly done when the user commits his order to get rid of the current shopping cart and everything else. You may want to remember selected information about that user, though, as shown below.

```
<?php
  page_open(array("sess" => "Shop_Session"));

// send order as mail
  mail_order($shopowner, $user, $cart);

// delete the current session
  $sess->delete();

// now get a new session id, but retain the users
  // address and name:
  page_open(array("sess" => "Shop_Session")); // will force auto_init again!
  $sess->register("user"); // could be done in auto_init as well

?>
```

Reading and understanding session data for debugging

When debugging PHPLIB applications, it is often useful to be able to read and understand the contents of the active_sessions table. Each session is represented by a single line in this table. The primary key to this table is the pair name and sid. name is the content of \$this->name and is usually the classname of your session class. sid is the content of \$this->id and is usually the MD5 hash of a uniqid and some magic string.

By choosing a pair, it is possible for PHPLIB to have more than one session type (for example, session and user data, see the <code>user</code> class below) per application and store all this data in a single table. If you are debugging a session class, for example <code>_session</code>, only records where <code>name = "Example_Session"</code> are of interest to you. Determine the current session id of your <code>Example_Session</code> by printing <code>\$sess->id</code> and select the record with that <code>name</code> and <code>sid</code> from the database.

The changed field indicates when this record has been updated the last time. It is a 14 character (Y2K compliant) string of the format YYYYMMDDhhmmss. Ordering by changed desc will show you the most current session records first (the MySQL "limit" clause may come in handy here).

The val column of a session record contains a PHP program that can be safely fed to stripslashes() first and eval() after that. The PHP program consists entirely of assignments and contains all instructions necessary to recreate the persistent variables. The structure and order of instructions within this program is always the same.

First item is always an assignment to \$this->in. If set to 1, auto_init has been executed by this session. If *not* set to 1, auto_init has not been executed, yet. This may be because no auto_init file is defined for that session.

After that comes code like this: \$this->pt = array(); followed by a bunch of assignments like \$this->pt["somestring"] = 1;. Each somestring is the name of a registered variable. Variable registrations are persistent themselves and are saved with the \$this->pt array. Even if the variable in question is not set, it may be registered and stays so until it is unregistered or the session is deleted. Check the contents of the pt array is you want to see which variables are currently registered with your session.

Finally, the actual contents of your variables are saved. This is always done by accessing the \$GLOBALS array and always by enumerating the scalar values that make up the persistent variable. For a scalar, you will see code like \$GLOBALS[somevar] = "value";

For an array, first \$GLOBALS[someary] = array(); is generated. Then the scalars that make up the array, if any, are written out, generating code that looks like \$GLOBALS[someary][index] = "value".

And for objects, code to create an object instance is saved: \$GLOBALS[someobj] = new Classname; ."Classname" is taken from the objects \$classname slot, which *must* be present and accurate. Then the scalars that are to be saved are written out, according to the contents of the objects persistent_slots array: \$GLOBALS[someobj]->slot = "value"; is written.

If you want to see what values have been saved to the database, you just have to look at the \$GLOBALS assignments for that session.

How "serialize()" operates

The following information is applicable only to library developers, that is, programmers that want to change the internal workings of PHPLIB. You may safely skip this section; some information here requires advanced understanding of the PHP language.

The heart of the session class is the serialize() internal function. This function takes an expression called prefix and generates PHP code that will assign the value of that expression to the expression when executed. For example, if the expression is \$GLOBALS["a"] and the global variable \$a\$ has the value 17, then serialize will create the PHP program \$GLOBALS["a"] = "17"; . To save memory, serialize() operates on a reference parameter \$str, where is will append the code generated.

First thing serialize() does is to determine the type of the current expression using the PHP <code>gettype()</code> function. The current type is stored in <code>\$t</code>. The type of the expression may indicate either a scalar value (integer number, float number or string), an array or an object.

Scalar values are the easiest to handle: serialize() just evaluates the current expression and remembers the result value in \$1. An assignment is generated that will assign the current value to the current expression. Since the current value may be a string and that string may contain bad characters (any of backslash, double quotes or dollar sign), these characters are backslashed. We are done, serialize() ends here for scalars.

In the case of \$t indicating an array, code is generated to create an empty array (expression = array();). Then the keys of current expression are enumerated and for each key serialize() is called recursively with the current key appended to the expression. That will append code for each array slot.

Should \$t indicate an object, code is generated to create that object (expression = new Classname;). Since one cannot find out the name of the class of an object for arbitrary objects in PHP, objects handled by serialize() must have a slot named classname. The object handler will then enumerate the contents of the objects slot persistent_slots and call serialize() recursively for each of these slots with the appropriate prefix.

Since many of the expressions used in serialize() require variable variable names or even variable code, eval() is used liberally. Unfortunately, this makes the code hard to read.

3.9 Auth

Authentication management can be used to authenticate a session, that is, to identify the user at the client side of the session.

Authentication is done inline, with HTML forms, *not* with HTTP authentication (that's the browser popup you get when you hit a page protected with htaccess). Inline authentication has several advantages over HTTP authentication:

- It can be undone: A session can be un-authenticated, the user can "log out".
- It can expire: A session can automatically be un-authenticated after a given idle time.

- It can be customized: You are not limited to user/password pairs. Instead you could use a customer number, operator id and a password to log in. Also, you have full control over the login screen, which is a normal HTML page with logos, help and forms as you see fit.
- It is database based. Authentication is being done against a database of your design, not a htpasswd text file.
- It is per page. You decide on a per-page basis which pages are authenticated and which aren't.
- It can be user authenticating and optionally self registering. In *registration* mode, a user without a valid login is encouraged to register and an account is created for this user.
- It works with CGI PHP. HTTP authentication is available only in mod_php.
- It is integrated with a permission checking scheme.

Instance variables

Accessible instance variables.

classname	Serialization helper: The name of this class.
persistent_slots	Serialization helper: The names of all persistent slots.
IIIIIeiime	Maximum allowed idle time before the authentication expires. If set to 0, The authentication never expires (as long as the session remains active).
refresh	Maximum allowed time before the authentication info (perms and alike) are re-read from the database calling auth_refreshlogin() method. If set to 0 authentication info are read only at the login stage.
mode	Authentication mode: log or reg (see below).
database_class	A classname. Auth uses this class to make a database connection.
database_table	Database table used to keep the session variables.
magic	An arbitrary value used in uniqid generation.
nobody	Flag: If true, we use default authentication.
cancel login	The name of a button that can be used to cancel a login form

Internal instance variables.

db	Internal: The database connection object instance.
auth	Internal: User authentication information, see below.
in	Internal: Used in default authentication mode.

Instance methods

Accessible instance methods

url()

A function that can be used in auth_loginform()a and auth_registerform. It returns the appropriate "action=" attribute to the form tag.

purl()

A function that can be used in auth_loginform() a and auth_registerform. It prints the appropriate "action=" attribute to the form tag.

login_if(\$t)

A function that can be used to change the current user identity. See the section and example on using default authentication below.

unauth(\$nobody = false)

This function destroys the authentication information in \$this->auth, forcing the user to relogin the next time a protected page is being loaded.

\$this->auth["uname"] is being kept, so that the correct username is available as a default.

Since V6: To give the user the credentials of `nobody', pass true as the first parameter to unauth. This will also change \$this->auth["uname"].

Since V7.2: Passing \$nobody to this method is deprecated.

logout(\$nobody = \$this->nobody)

This function destroy all authentication information in \$this->auth, forcing the user to relogin the next time a protected page is being loaded.

Most applications want to use \$this->unauth() instead.

Since V6: To give the user the credentials of `nobody', pass true as the first parameter to logout. This defaults to the value you set in the class definition (\$nobody). logout() will call unauth() (passing \$nobody), so the behaviour is identical (except logout() will always clear \$this->auth["uname"] and unregister the auth class).

Since V7.2: Passing \$nobody to this method is deprecated.

is_authenticated()

Will return false, if the current authentication is invalid or expired. Will return the authenticated uid otherwise.

auth_preauth()

This function can be overridden in a subclass to Auth. It is being called as the very first step in the authentication process and has the opportunity to authenticate the user without a loginform being displayed (by deriving all necessary information telepathically, or by using cookies, or divining the user identities from the incestines of a dead squirrel).

If it returns a UID value, the user is authenticated and neither auth_loginform() nor auth_validatelogin() are called. If it returns false, all goes on as usual.

auth loginform()

This function must be overridden by a subclass to Auth. It should output HTML that creates a login screen for the user. We recommend that you use an <code>include()</code> statement to include your HTML file.

auth validatelogin()

This function is called when the user submits the login form created by auth_loginform(). It must validate the user input.

If the user authenticated successfully, it must set up several fields within the \$auth[] instance variable:

"uid"

must contain the user id associated with that user.

"uname"

must contain the user name as entered by the user.

"exp"

must not be tampered with (field is maintained by start(), contains the time when the login expires).

"perm"

if you want to use the permission feature, you must store the permissions of the validated user here. (Hint: due to a name conflict with sybase, "perm" is called "perms" in all the databases tables. Look for this small difference!)

See the example below for more information.

auth_refreshlogin()

This function is called every refresh minutes. It must refresh the authentication informations stored in auth array by auth_validatelogin() method. It is not called if the user is logged in as nobody.

It must return true on success, false otherwise (i.e.: the userid is no longer valid).

auth_registerform()

See auth_doregister().

auth doregister()

These functions mirror auth_loginform() and auth_validatelogin() in registration mode.

Internal instance methods

start()

Initialization function, does the authentication. If we are in log (login) mode, auth_loginform() is called to draw a login screen. When the login screen is submitted back, auth_validatelogin() is called to validate the login. If the validation was successful, the actual page content is shown, otherwise we're back at auth_loginform().

In reg mode, auth_registerform() is called to draw a registration form. When the registration form is submitted back, auth_doregister() is called to register the user and to validate the session. If registration was successful, the actual page content is shown, otherwise we're back at auth_registerform().

Example

Use a subclass of Auth to provide parameters for your authentication class and to implement your own auth_* functions.

```
class My_Auth extends Auth {
  var $classname = "My_Auth"; # Object serialization support
```

```
var $lifetime
                            15;
  ## DB_Sql subclass and database table to use
  var $database class = "DB Session";
  var $database table = "auth user";
  ## Some magic value to make our uids harder to guess.
  var $magic = "Abracadabra";
  ## Use an own login form
  function auth_loginform()
    qlobal $sess;
    include("loginform.ihtml");
  function auth_validatelogin() {
                                      ## form variables from loginform.ihtml
    global $username, $password;
    ## If authentication fails, loginform.html will
## find $this->auth["uname"] set and use it.
    $this->auth["uname"]=$username;
    ## Value to return in case auth fails.
    $uid = false;
    ## Check the database for this user and password pair.
    $query = sprintf(
      "select * from %s where username = '%s' and password = '%s'",
      $this->database_table,
      addslashes($username),
      addslashes($password)
    $this->db->query($query);
    ## If we found a matching user, grab the uid and permissions...
    while($this->db->next_record())
      ## Required.
      ## Optional, for the perm feature.
$this->auth["perm"] = $this->db->f("perms");
      ## if you use perm feature be aware, that the db-field in our
      ## example table is called "perms" due to a name conflict with sybase
    return $uid;
}
```

Your loginform.ihtml contains HTML and PHP code to draw a login form. \$this->auth["uname"] will be empty on the first login attempt and set on all further login attempts. You can use this to detect repeated login attempts and display an appropriate error message. You must print the result of \$this->url() to create your forms action attribute.

See the provided loginform intml for an example.

Use the page management functions (see above) to use your authentication subclass. The feature name for authentication management is auth; provide the name of your Auth subclass as a parameter to the auth feature. The auth feature requires the sess feature:

```
page_open(array("sess" => "My_Session", "auth" => "My_Auth"));
```

Using default authentication

Many applications want to use \$auth and \$perm objects to protect functionality on a page, but do want to make the unprotected part of this page available to users with no account. This presents a kind of dilemma, because you need \$auth and \$perm objects to protect functionality on a page, but you don't want a login screen to appear by default.

Default authentication solves this dilemma by providing a special uid and uname "nobody", which is guaranteed to fail every permission check. If you set the nobody flag, sauth will not create a login screen to force a user to authenticate, but will authenticate the user silently as nobody. The application must offer a login button or other facility for users with accounts to change from that id to their real user id.

To use default authentication, create a subclass of My_Auth as shown above with the nobody flag set (*Note:* No need to extend in two steps. The only important thing here is that the nobody flag is set.)

```
class My_Default_Auth extends My_Auth {
  var $classname = "My_Default_Auth";

  var $nobody = true;
}
```

To create a page that uses default authentication, use the page management functions. Check for relogin requests with the <code>login_if()</code> function. Create a relogin link on your page.

```
<?php
  // using Default Authentication
  page_open(array("sess" => "My_Session", "auth" => "My_Default_Auth"));
  $auth->login_if($again);

  if ($auth->auth["uid"] == "nobody"):
?>
    <A HREF="<?php $sess->purl("$PHP_SELF?again=yes") ?>">Relogin</A>
    to this page.
<?php endif ?>
```

Using Challenge-Response Authentication

As distributed, local.inc contains an example class named Example_Challenge_Auth, which uses a Challenge-Response authentication scheme. If the client browser supports Javascript, this login screen does not transmit passwords in clear over the network. If the client does not support Javascript, login is still possible, but passwords are transmitted in clear, as regular Example_Auth always does.

Example_Challenge_Auth is there to demonstrate advanced usage of PHP and Javascript and to show off the flexibility of the library base classes: The Challenge-Response authentication scheme has been implemented completely and naturally in local.inc by subclassing Auth with no alteration of library code.

Example_Challenge_Auth includes crloginform.ihtml. It also requires that the file md5.js is present in the document root directory of your web server. That file contains an implementation of the MD5 message digest algorithm done by Henri Torgemane. The basic idea behind this authentication scheme is simple: \$auth->auth_loginform() creates a challenge value which is incorporated into this form. When the user tries to submit the form, MD5("username:password:challenge") is calculated and filled into the reply field. The password field is erased. The server can calculate the expected reply from the username received, the password in the database and the challenge, which it knows. It can compare the expected reply to the actual reply value. If they match, the user is authenticated.

If the reply field is empty and password is set, the server knows that the client cannot do Javascript. The user can still be authenticated, but the password is visible on the network.

The class is a dropin-replacement for Example_Auth.

The complete guide to authentication and user variables

This feature has originally been written for the PHPLIB mailing list by Kristian Köhntopp and was included into the

documentation later.

How is the Auth class used usually?

Usually, you write code like this into the top of the page you want to protect:

```
<?php
page_open(array(
    "sess" => "My_Session",
    "auth" => "My_Auth"));
?>
<!-- your code here -->
<?php
page_close()
?>
```

How does sauth work internally?

When you access this page, the call to page_open() is being made as the first thing on that page. page_open() creates an instance of My_Auth named \$auth and starts it. \$auth then detects that you are not authenticated (how it does, I will explain below) and displays loginform.ihtml. \$auth then exits the interpreter, so that <!-- your code here --> is never being executed or displayed.

The user now sits in front of a loginform.ihtml screen, which is shown under the URL of the page the user originally tried to access. The loginform has an action URL, which just points back to itself.

When the user filled out the loginform and submits it, the very same URL is requested and the above page_open() is reexecuted, but this time a username and a password are submitted. When the \$auth object is created and started, it detects these parameters and validates them, resulting in either a NULL value or a valid user id. If the validation failed, creating an empty user id, the loginform is displayed again and the interpreter exits. Again <!-- your code here --> is not executed.

If a UID is returned, that UID and a timestamp are being made persistent in that session and \$auth returns control to page_open(). When page_open() finishes, which it may or may not do, depending on the presence and result of an optional \$perm check, <!-- your code here --> is being executed or shown.

Later calls to other pages or the same page check for the presence of the UID and the timestamp in the sessions data. If the UID is present and the timestamp is still valid, the UID is retained and the timestamp is refreshed. On page_close() both are written back to the user database (Note: Authenticated pages REQUIRE that you page_close() them, even when you access them read-only or the timestamp will not be refreshed).

If the UID is not present (\$auth->logout() or \$auth->unauth() have been called, for example) or the timestamp has expired, \$auth will again intercept page display and draw the loginform.

The only way to get into a page with an sauth object on it is to have a UID and a valid timestamp in your session data (Note: This is true even for default authentication. These create a dummy UID and timestamp in your session data).

How do \$sess and \$auth interact?

Your browser has a session cookie, named after your session class. This is the only thing that is ever shipped between your browser and PHPLIB, as far as core functionality is concerned. The session cookie value is used as a reference into active_sessions, to retrieve PHPLIB generated PHP code, which is then eval()ed and recreates your session variables within page_open().

Part of the \$auth object now makes itself persistent and is retrieved when the \$sess part of page_open() is being executed. This is just before the \$auth part of page_open() gets its turn, so that \$auth can rely on its persistent data being present when it is being called.

From the PHPLIB source you all know that \$auth has only one persistent slot, called \$auth->auth[], of type hash. This hash contains the slots uid, exp and uname. \$auth->auth["uid"] is the currently authenticated user id, \$auth->auth["exp"] is the currently active expiration timestamp (Unix time_t format) for that uid. \$auth->auth["uname"] is completely irrelevant as far as the regular PHPLIB Auth class is concerned. It is relevant in the context of the supplied default Auth subclass Example_Auth, though.

So a session is authenticated, if it contains \$auth->auth["uid"] != false and time() < \$auth->auth["exp"].

Where is the beef?

The original Auth class as included in PHPLIB makes no assumptions at all on how a loginform looks or how and where uids come from. There is no code at all in Auth that ever checks anything but the above two conditions. It is your responsibility to modify a subclass of Auth in a way that these conditions can ever be met.

Auth helps you in doing this by calling its own function \$auth->auth_loginform() when it wants to draw a loginform. Unfortunately this function is empty in Auth itself, so you have to provide an implementation for that. The suggested standard implementation in local.incs Auth subclass Example_Auth is

```
function auth_loginform() {
  include("loginform.ihtml");
}
```

and you put your code into that file. We also provide sample code for that file, but you are not limited to that code and may write a loginform.ihtml as it meets your needs.

When the loginform has been filled in and submitted back by the user, Auth calls <code>\$auth->auth_validatelogin()</code>. Again, this function is empty in Auth itself and so Auth by itself will never function correctly. You have to subclass Auth and provide your own implementation of <code>\$auth->auth_validatelogin()</code> in <code>local.inc</code> to make it work.

What you actually do in that function is completely irrelevant to Auth itself. It only exspects that you either return false, if the user-supplied authentication data was invalid, or a user id, if the user could be validated. Auth then takes care to create the appropriate entries (\$auth->auth["uid"] and \$auth->auth["exp"]) in the session record.

I still do not understand! What am I supposed to code?

You write your code into local.inc, after you have removed the classes Example_Auth, Example_Default_Auth and Example_Challenge_Auth from that file (keep a copy around, just for reference).

You code a class called My_Auth and you use that name later in your calls to page_open as an argument to the auth feature, as show at the start of this message. Follow the standard rules for deriving persistent classes in PHPLIB when you create your code, that is, do it like this:

```
class My_Auth extends Auth {
var $classname = "My_Auth";
// we inherit $persistent_slots and do not need to modify it.

// later code is inserted here
}
```

Now configure the lifetime of the authentication, that is, how many minutes in the future shall the current value of <code>\$auth->auth["exp"]</code> be? Also, name a database connector class and name the table that you will be using to check usernames and passwords.

```
// insert this code as indicated above.
var $lifetime = 15;
var $database_class = "DB_Example";
var $database_table = "my_special_user_table";
// later code is inserted here
```

Okay, now we have a basic implementation of My_Auth that is only lacking the required functions auth_loginform() and auth_validatelogin(). Our implementation of auth_loginform() will have access to all \$sess variables by globaling \$sess into our context (because these can come in handy) and to all \$auth variables (via \$this).

```
function auth_loginform() {
   global $sess;
   include("loginform.ihtml");
}
```

The loginform is free to do whatever it damn well pleases to create a form for the user to supply the needed values for authentication. It has access to anything \$sess and anything \$this related.

The loginform will display some input fields for the user, for example a given name, a surname and a password. When the form is submitted back, auth_validatelogin() is being called. The form values are global variables (or \$\text{HTTP_x_VARS[]}\$) and must be imported into \$\text{auth->auth_validatelogin()}\$. Then, \$\text{auth->auth_validatelogin()}\$ is free to do whatever it must do to produce a unique identifier for that user (or return false).

Suppose you created input fields named given_name, surname and password. So go ahead, global \$given_name, \$surname and \$password and set \$uid to false. Then create the SQL needed to access you user table and retrieve the user record from your database as indicated by \$given_name and \$password.

The query may succeed, if a record with matching \$given_name, \$surname and \$password is present. In that case return the uid, which uniquely identifies exactly that (given_name, surname) pair. Else return false.

In code:

```
function auth validatelogin() {
  // import authentication data
 global $given_name, $surname, $password;
 $uid = false;
 $query = sprintf("select uid
                      from %s
                     where given_name = '%s'
  and surname = '%s'
                       and password = '%s'",
             $this->database_table,
             $given_name, $surname, $password);
  // we really should use addslashes() here,
 // or have magic_quotes active.
  // $auth->db is our DB_Example database connection
 $this->db->query($query);
  // now check for any results
 while($this->db->next_record()) {
```

```
// either $uid is false now (no results)
// or set to the last retrieved value from the uid
// column.

// Anyway we are set now and can return control
return $uid;
}
```

Okay, that's all and useable now. There is room for some improvements, though: First we did not retrieve permission data, so this will not work, if we want to use the perm feature as well.

This is easily changed: Modify the query to select uid, perms instead of select uid alone. Of course, you may call your perm column whatever you like, just adapt the SQL accordingly. Also, add a line after the \$uid assignment so that the code looks like this:

```
$uid = $this->db->f("uid");
$this->auth["perm"] = $this->db->f("perms");
```

This will store the retrived perms value under the key perm within the <code>\$auth->auth[]</code> array. It will be kept around in that place in case <code>\$perm</code> is called and starts looking for the current permissions of that user.

Another possible improvement becomes apparent when you try to login and fail to do so correctly: auth_validatelogin() returns false and you hit the loginform again. Empty loginform that is, because we did not remember what you typed into the given_name and surname fields before. If we remembered what you typed, we could easily supply these values back to you so that you can correct them. We would also be able to detect if this is a second, third, ... attempt to login and display an appropriate error message somewhere in that loginform to inform the user of his or her typo. A convenient place to store these values is the \$auth->auth array, which is persistent anyway.

Standard Example_Auth uses the field <code>\$auth->auth["uname"]</code> to store that value, but you may use any field and as many fields as you like as long as you make sure not to clash with any of the three officially used fields, uid, exp, and perm.

Do not try to turn the global variables \$given_name and \$surname into persistent variables by calling \$sess->register("given_name") and \$sess->register("surname")! Remember: These are form variables! Never ever make form variables persistent and never ever trust unvalidated user supplied from the Internet!

So add the folling code just below the "global" line:

```
$this->auth["gname"] = $given_name;
$this->auth["sname"] = $surname;
```

and check for these two variables in loginform.ihtml at the appropriate places.

Ok, I did that and it works. I even understood it. Now, what exactly is that uid used for?

It is simply a token to indicate that the user is authenticated. We use a different token for each user, so that we can decide which user we are currently dealing with. You can think of the uid as a primary key for your auth_user table (or whatever it is being called in your current application). The (given_name, surname) tuple would also be a possible primary key, albeit a compound one. It is the external, human-readable (and probably sometimes very long) representation of the internal uid. The password field is functionally dependent on either of both key candidates.

The internal user id should never be presented to the user; the (given_name, surname) pair is much more natural to handle for the user and easier to remember (A user who does not remember his or her name would probably not be in a

state of mind to operate the rest of the application anyway:-).

The internal user id should always be used to identify a user internally within an application, though. That is, because the uid is of a fixed length and has a known form and structure, so you can make assumptions. A given_name or surname may be of any length and may contain about any character, so you probably do not want to use this as a user-reference internally.

But is the uid used internally by PHPLIB?

Yes, if you make use of the user feature of page_open(), that is, if you create user variables.

The User class is actually a subclass of Session. That is, user variables are just like session variables. They are even stored in active_sessions. The only difference is that the session has a different name (it is called Example_User instead of Example_Session, if you use the classes and names supplied in local.inc).

And in Example_User, the user id of the authenticated user becomes the session id in the active_sessions table. That is the reason why we recommend md5(uniqid("abracadabra")) style uids.

3.10 Perm

Permission management relies on an authenticated session. It associates a set of required permissions with a page. The actual page content is only visible to users with ALL matching permissions; all other users are shown a screen of your design.

Instance variables

Accessible instance variables.

classname	Serialization helper: The name of this class.
permissions	A hash of (name, permission bit) pairs.

Instance methods

Accessible instance methods

check(\$required)

Checks that the currently authenticated user has all the rights that are specified in required. If not, perm invalid() is called.

If one or more of the required rights or user rights are invalid (not to be found in the permissions hash), perm_invalid() is called as well.

have_perm(\$required)

Similar to check() in usage, only that it doesn't halt the session if the user doesn't have the appropriate rights: This function returns true, if the user has the required rights, false otherwise.

perm_sel(\$name, \$current = "", \$class = "")

This function returns a SELECT-tag with the given name. Within this tag, all available permission values from \$perm->permissions are contained as OPTION tags.

If you supply a value for current, the permission value that matches current is SELECTED. If you supply a value for class, the tags are marked with that CSS stylesheet class.

Internal instance methods

permsum(\$rights)

Logically or's all the rights and returns a pair (valid, or_result). If valid is true, an or_result is provided. If valid is false, the or_result is undefined and one or more of the rights do not exist at all. This is a severe error and the application should be halted at once.

perm_invalid(\$does_have, \$must_have)

Called in case of an access violation. does_have is a string listing the rights the user actually has. must_have are the rights the page requires.

Example

Use a subclass of Perm to provide parameters for your permission class and to implement your own perm_invalid function.

Use the page management functions (see above) to use your permission subclass. The feature name for permission management is perm; provide the name of your perm subclass as a parameter to the perm feature. The perm feature requires the sess feature and the auth feature:

```
page_open(array("sess" => "My_Session", "auth" => "My_Auth", "perm" => "My_Perm"));
```

Use the check() instance method to protect your page:

```
$perm->check("admin"); ## This page is for users with admin rights only.
```

Use have_perm() to create protected functionality on a page:

```
<?php
  if ($perm->have_perm("admin")):
  ?>
  <h1>Admin only functionality</h1>
```

```
<?php
endif;
?>
```

How permissions work

Your subclass of Perm defines an array \$permissions, which translates permission names into bit patterns. For example, the definition of Example_Perm in the distributed local.inc defines the names user, author, editor, supervisor and admin, all of which translate into a bit pattern with a single bit set.

A user may be assigned any number of permissions as a comma separated list of permission names (no spaces!) in the perms column of the auth_user table. The effective permissions of the user are determined by logically OR'ing the bit patterns of these permissions.

A page may require any permissions as a comma separated list of permission names (again no spaces!) with the \$perm->check() function. The required permissions are again determined by logically OR'ing the bit patterns of these permissions. Similarly, a page function may be protected by requiring permissions with \$perm->check().

Access is granted to a protected page or a protected page function, if the effective permissions of the authenticated user have all the required bits set, that is: If the effective permissions of the user logically AND'ed with the required permissions are equal to the required permissions.

With the permission names as defined in Example_Perm from the distribution, a user kris may be defined with admin permission in the auth_user table. A page that requires admin, user permission with \$perm->check("user,admin") is inaccessible to this user. This is how it is calculated:

```
Effective Permissions of User: admin translates into: 16

Required Permissions of Page: user,admin translates into: 1 OR 16 == 17

Permission Check: Effective Permissions 16

AND Required Permissions 17

ARE 16 & 17 = 16

MUST BE Required Permissions 17 -> access denied
```

The example permissions as defined in Example_Perm from the distribution are called *atomic* permissions, because each of them has only a single bit set. Atomic permissions are the simplest of all schemes, because they allow for easy permission checks: To access a page protected with user, admin, you need to have at least user, admin rights in your auth_user table.

Another common scheme used in permission definitions are inclusive permissions. In this scheme, each permission definition has all bits of its predecessor set plus one addition bit. For example

defines a set of inclusive permissions. In this example, a user kris with admin permissions can easily access a page protected with editor permissions. This is how it is calculated:

```
Effective Permissions of User: admin translates into: 31

Required Permissions of Page: editor translates into: 7

Permission Check: Effective Permissions 31

AND Required Permissions 7

ARE 31 & 7 = 7

MUST BE Required Permissions 7 -> access granted
```

Inclusive Permissions are easy to deal with, too, because a user with a *higher* access level may access all pages or page functions with a *lower* access level.

Due to limitations of your machines integer size you can only define up to 31 permission levels.

3.11 User

The user class is an extension (a subclass) of the Session class. It keeps a list of global variable names and provides a set of functions to load and save these variables from and to a database. The same restrictions as for session variables apply to user variables.

Unlike session variables, user variables are not lost when the user stops and restarts the browser or moves to a different workplace (the session id is then lost and consequently all session variables are lost, since they are bound to the session id).

User variables require that the user logs in, because they depend on the availability of a User id to bind variables to this id. Thus, User is dependent on Auth.

The User class is an extension of the Session class. It has all instance variables and instance methods of Session, only that some are implemented different. This documentation only describes these differences.

Note that Session and User can successfully share a single active_sessions table in a database due to the different values in the name column.

Instance variables

Accessible instance variables.

classname	Serialization helper: The name of this class.magic	Not meaningful for User.
mode	Not meaningful for User.	
fallback_mode	Not meaningful for User.	
lifetime	Not meaningful for User; see authentication lifetime in Auth instead.	
gc_time	Functional, but probably not useful in User.	
gc_probability	Functional, but should be set to 0 in User.	
that_class	A classname. User uses this class to store and retrieve data.	
auto_init	Not meaningful for User.	
secure_auto_init	Not meaningful for User.	

Internal instance variables.

pt	Internal array of names of persistent variables.
name	A tag (name) for the session type.
id	Id of the current session.
that	Container object instance.

Instance methods

Accessible instance methods

register(\$varname)

Works as expected.

unregister(\$varname)

Works as expected.

delete()

Works as expected.

url(\$url)

Not useful with User.

purl(\$url)

Not useful with User.

self_url()

Not useful with User.

pself_url()

Not useful with User.

reimport_get_vars()

Works as expected.

reimport_post_vars()

Works as expected.

reimport_cookie_vars()

Works as expected.

Internal instance methods

get_id()

This is only a stub implementation that depends on the user id provided by the page management functions. The page management functions will use \$auth->auth["uid"], which is set up by Auth.

put_id()

Empty. Not useful with User.

serialize(\$prefix, &\$str)

Works as expected.

freeze()

Works as expected.

thaw()

Works as expected.

gc()

Works as expected. You do not want to use it, though.

reimport_any_vars(\$arrayname)

Works as expected.

start()

Initialization function, to be called after object instantiation. Calls get_id() to get the current session id, creates a database connection, then calls thaw() to load all session variables. *Note:* gc() activation is commented out! Remove the comments if you really want gc with User variables.

Example

Use a subclass to provide the appropriate parameters to your user variables. Usually your subclass looks like this:

```
class My_User extends User {
  var $classname = "My_User"; ## Persistence support
  var $that_class = "CT_Sql";
}
```

Remember that you have to provide a DB_Sql subclass with the parameters needed to access your database.

Use the page management functions (see above) to use your User subclass. The feature name for user variables is user; provide the name of your User subclass as a parameter to the user feature:

```
page_open(array("sess" => "My_Session", "auth" => "My_Auth", "user" => "My_User"));
```

Use the register() instance method to register variables as persistent. If \$user is your user object, use

PHP Base Library Documentation, Release phplib_7_2: Core Functionality

```
$user->register("u");
```

to make the global variable \$u persistent. \$u may be a scalar value, an array or an object with persistence support slots.

Do not use the instance methods freeze() and thaw() directly, but use the page management functions instead.

Note: Using default authentication and user variables is going to be a problem, because currently User does not do any locking. This is, because the DB_Sql has currently no portable locking mechanism.

Next Previous Contents

4. Extended functionality

The section on extended functionality covers non-GUI classes that provide often needed application functions without a user interface. Some extended classes depend on core functionality, some contain independent classes.

Extended classes are treated differently from core classes in that their code is not automatically included by prepend.php3. You have to include the class definition manually where needed or you modify prepend.php3.

4.1 Cart

The Cart class is programmatically independent, but makes sense only if its instances are made persistent in some way. The Cart class automatically registers itself as a session variable in its start() function.

Cart implements a shopping cart. At the moment, items within the shopping cart are independent of each other; the cart can only hold simple things. Support for compound articles that require other articles to function and provide a base for dependent articles is to be added at a future time.

An example of a simple article is any article with no options, for example an apple or a book. Common examples for compound articles are a pizza (which requires a foundation in either American or Italian style, a selection of toppings, and cheese, to function correctly) and a computer system (which requires a housing, a motherboard, RAM, a video card, etc to function correctly).

Note: Cart was a core class up to *release-5*. If your applications uses the Cart class, you *must* manually add the statement include("cart.inc") to your prepend.php3 file where indicated in that file.

Note: The page management functions do no longer support the feature cart to set up and start the cart class. It is recommended that you use Session's auto_init feature instead to start your cart automatically or that you manually set up your cart.

Instance variables

Accessible instance variables.

classname	Serialization helper: The name of this class.
persistent_slots	Serialization helper: The names of all persistent slots.
item Multidimensional array of items in the cart.	
currentItem	A counter for item positions.

Instance methods

Accessible instance methods

check(\$art)

Checks that an item with the given article number \$art is in the cart. Returns an array of a boolean value and an integer number. If the boolean is true, there are number many articles of that article number in the cart.

reset()

Deletes all items in current cart, resetting \$this->currentItem to 1. Always returns true.

num items()

Returns the number of articles in the current shopping cart, or false if cart is empty. For compatibility reasons, this function is available as tot_arts as well (but will print a warning if used by this name).

add_item(\$art, \$num)

Add \$num many articles of article number \$art to the current shopping cart. Returns the position number of \$art in the shopping cart.

remove item

Remove \$num many articles of article number \$art from the shopping cart, if there are at least that many articles in the cart. Returns the position number of \$art in the shopping cart or false, if there weren't enough \$art to remove them from the cart. If the function does return false, the cart has not been modified.

set item

Set the quantity of article number \$art in the shopping cart to exactly \$num. If \$num is set to zero, article is removed from cart. Returns the position number of \$art in the shopping cart.

show_all()

If the shopping cart is empty, it will call <code>show_empty_cart()</code> once and then return.

Calls <code>show_item_open()</code> once at the beginning of a shopping cart listing. Then calls <code>show_item()</code> once for each item in the shopping cart. Calls <code>show_item_close()</code> once at the end of a shopping cart listing.

show_item(\$art, \$num)

This function should be provided by the user. It renders the HTML to display a single item from the cart. \$art is the article number of the item and there are \$num of these in the cart.

show_cart_open()

This function should be provided by the user. It renders the prologue HTML to display a shopping cart listing.

show_cart_close()

This function should be provided by the user. It renders the epilogue HTML to display a shopping cart listing.

show empty cart

This function should be provided by the user. It should render an appropriate message to symolize an empty cart.

Example

Use a subclass of Cart to provide an implementation of show_item().

```
class My_Cart extends Cart {
  var $classname = "My_Cart";
```

```
// Look up article numbers..
 var $database_class = "DB_Article";
 var $database_table = "articles";
 var $db;
 var \$sum = 0;
 function show cart open()
   printf("\n");
   this->sum = 0;
 function show_cart_close() {
   printf(" \ \ ");
   printf("That's a total of %s.\n", $this->sum);
 function show_item($art, $num) {
  if (!is_object($this->db)) {
     $class = $this->database_class;
     $this->db = new $class;
   $query = sprintf("select * from %s where artid = '%s'",
     $this->database_table,
     $art);
   $this->db->query($query);
   while($this->db->next_record()) {
     printf(" \n
                                 %s\n",
       $this->db->Record["name"]);
     printf(" %s\n",
       $this->db->Record["price"]);
     printf(" %s\n",
       $num);
     $rowsum = $num * $this->db->Record["price"];
     $this->sum += $rowsum;
     printf("
             %s\n",
       $rowsum);
    printf(" \n");
 }
}
```

To use a cart, create an instance of your cart subclass and call start(). This will automatically register cart.

It is recommended that you set in your Session subclass in local.inc the slot \$auto_init to the value setup.inc and create an include file of that name which contains the following code:

Use add_item() and remove_item to work with your Cart:

```
$cart->add_item("101", 2);  ## Add two pieces of "101"
$cart->remove_item("101", 1); ## Drop one piece of "101"
```

Use show_all() to display the contents of your cart.

```
$cart->show_all(); ## What's in a cart, anyway?
```

On using Cart

To make use of the Cart class, you need to define a new table in your database that lists all articles you shop should

sell. With PHPLIB and MySQL we recommend that you create a new instance of PHPLIB for each virtual web server and a new database for each customer. This database should hold the active_sessions and auth_user tables as well as all application specific tables like for example the article list. In other words, with MySQL we strongly discourage that you use PHPLIB and the MySQL directive use database_name together. There is no support if you do (there is no support if you do as we say, too, because PHPLIB is an open source product you are using on your own risk, but ...).

So let us assume you define a very simple new table articles with a structure like this:

```
#
# Table structure for table 'articles'
#
CREATE TABLE articles (
   name text,
   price float(8,2),
   artid int(11) DEFAULT '0' NOT NULL auto_increment,
   PRIMARY KEY (artid)
);
```

This table has an article number called artid, and for each artid there is an article description name and a price. You may extend this minimal definition for your purposes by adding article groups, BLOBs with article images and more, but this will suffice for our example purposes.

Populate this table with some products that suit your taste.

The next step is to teach PHPLIB about the cart class. Three steps are necessary to do so:

- the Cart class has to be included on every page. Even on that pages that do not make use of the Cart class. On that pages that use Cart, a cart subclass is instantiated and saved. On all subsequent pages, that Cart object is recreated and to be able to recreate the Cart object, PHP must know what a Cart object is. Since you cannot know which pages a user loads after he has put the first item into the Cart, we need to have a definition for Cart on all pages. The proper place to include the Cart definition from cart.inc is consequently prepend.php3. Edit prepend.php3 and require("cart.inc") as indicated by the comments in that file.
- a subclass of Cart has to be created to suit your tastes. Your subclass of Cart will be called Example_Cart in this example. You may actually name it as you like, but you have to be consistent. The definition of Example_Cart goes into local.inc anywhere below your definition for Example_Session. It looks like this

```
class Example_Cart extends Cart {
  var $classname = "Example_Cart";
}
```

and we will add additional code later in this example. That additional code will teach your shopping cart about the database table that holds your articles and so on.

• finally, you need to create an instance of your shopping cart class so that you have an object that actually holds the articles selected by the user. We will use a very nifty feature of PHPLIB to create that object instance: If you set up PHPLIB properly, it is able to load and execute an include file every time a session is being created. We call this feature auto_init, after the instance variable of Session that controls it. Go into local.inc and edit your subclass of Session. You will have some code like

```
class Example_Session extends Session {
  var $classname = "Example_Session";
...
}
```

in your local.inc. Add a line like

```
var $auto_init = "setup.inc",
```

to your definition of Example_Session and create a file setup.inc in the same directory that holds your local.inc. Whatever code is in this file will be executed every time we create a new session. The code is being executed after your \$sess, \$auth and \$perm objects are loaded and initialized, but does run from within a function context. You have to global everything you define to export it from that function context. In setup.inc, create a global instance of Example_Cart named \$cart and register that variable with PHPLIB:

```
<?php
  global $cart;
  $cart = new Example_Cart;

// $sess is already global
  $sess->register("cart");
?>
```

Now you have a \$cart object available by default on every page that uses PHPLIB. That object is created automatically at session startup, is carried from page to page by PHPLIBs session management and is destroyed by the garbage collection that reaps session records. You do not have to worry anymore about that cart, but simply use it anytime between page_open() and page_close(). PHPLIB does the rest for you.

The cart class is actually dead stupid. It maintains an array \$cart->item[] that holds records about what the user bought. Each \$cart->item[\$x] consists of a \$cart->item[\$x]["art"], which is the article number of an item the user wants to buy and of a \$cart->item[\$x]["num"], which is the # of items with that article number that are wanted. \$cart->currentItem is the next \$x\$ to use for articles added to \$cart->item[].

You add articles to the shopping cart with

```
$x = $cart->add_item($art, $num)
```

This will add \$num items with the article number \$art to your cart contents. If you already have an item with that article number in your cart, the count for that article is increased by \$num. Otherwise a new article entry is being created and set to \$num. The function does return the \$x index into the \$cart->item[] array for that article.

To remove an item from the shopping cart, code

```
$x = $cart->remove_item($art, $num)
```

This will remove \$num items with the article number \$art from your cart, if there are that many items in your shopping cart. If you do not have the \$art in your cart or there are not \$num many \$art in your cart, the function will return false and not remove anything from the cart. Otherwise, \$num articles with article number \$art are taken out of the cart and if the count for that article drops to zero while doing this, we even unset the array element.

You may check how many articles with a given article number are in the cart:

```
list($have, $num) = $cart->check($art)
```

The check function does return a two-element array. The first element \$have is true, if we have the wanted article in the cart. If \$have is true, \$num holds the number of articles with that number in the cart, otherwise \$num is undefined (actually, it is 0, but you must not rely on that).

Finally, we have a function

```
$cart->show_all()
```

which you may call to walk your shopping cart and have Example_Cart to generate a list of articles in your cart. That function will first call <code>\$cart->show_cart_open()</code>, for which you may provide code in your subclass. It will then call <code>\$cart->show_item(\$art, \$num)</code> for each item in the cart. We have a stupid default implementation for that function in Cart, but you may provide more sophisticated code in <code>Example_Cart</code> for that, too. Finally, at the end of your cart listing, <code>\$cart->show_cart_close()</code> is being called, which again may be code of yours.

The example in the previous section shows a more sophisticated implementation of a Cart subclass. That implementation uses show_cart_open() to create an opening table tag (formatted with a CSS class) and sets a counter scart->sum to zero.

In show_cart_close(), the table is being closed and the \$cart->sum counter is printed.

As you might have guessed, <code>show_item(\$art, \$num)</code> queries the database for each article number, retrieves the article description and prices and finally sums up all prices, taking the number of articles per article into consideration. It also generates table rows, printing a nice receipt for the customer.

4.2 Template

Note: If you think that this is like FastTemplates, read carefully. It isn't.

The template class allows you to keep your HTML code in some external files which are completely free of PHP code, but contain replacement fields. The class provides you with functions which can fill in the replacement fields with arbitrary strings. These strings can become very large, e.g. entire tables.

Instance variables

Accessible instance variables.

classname	String. Serialization helper: The name of this class.
debug	Boolean: if set to true, the class will emitdebugging output.
unknowns	One of "keep", "comment", "remove" (Default).Determines how to handle unresolved variable names intemplates upon output. If set to "keep", those are leftuntouched. If set to "comment", unresolved variable names are transformed into HTML comments reporting the error. If set to "remove", unresolved variable names are silently removed (thedefault).
halt_on_error = "yes"	One of "yes" (Default), "report", "no". Determines how Template handleserror conditions. If set to "yes" (the Default), the error isreported, then execution is halted. If set to "report", theerror is reported, then execution continues by returning "false". If set to "no", errors are silently ignored, and execution resumes reporting "false".
last_error =	The last error message iskept in this variable.

Internal instance variables.

file	Hash of strings. A translation table whichtranslates variable names into filenames.
root	String (Pathname). The base directory from whichtemplate files are being loaded.

varkeys	Hash of strings. A translation table whichtranslates variable names into regular expressions forthemselves.
varvals	Hash of strings. A translation table whichtranslates variable names into replacement values for their respective varkeys.

Instance methods

Accessible instance methods

Template(\$root = ".", \$unknowns = "remove")

Constructor. May be called with two optional parameters. The first parameter sets the template directory (see set_root(), the second parameter sets the policy regarding handling of unknown variables.

set_root(\$root)

The function checks that \$root is a valid directory and sets this directory as the base directory where templates are being stored.

set_unknowns(\$unknowns = "keep")

The function sets the policy for dealing with unresolved variable names. Must be either "remove", "comment" or "keep". If set to "keep", those are left untouched. If set to "comment", unresolved variable names are transformed into HTML comments reporting the error. If set to "remove", unresolved variable names are silently removed (the default).

set_file(\$handle, \$filename = """)

The function defines a filename for the initial value of a variable. It may be called with either a \$handle/\$filename pair or with a hash of \$handle/\$filename pairs. The files are not referenced yet, but only when needed.

set_block(\$parent, \$handle, \$name = "")

A variable \$parent may contain a variable block named by \$handle. The function removes that block from \$parent and replaces it with a variable reference named \$name. If \$name is omitted, it is assumed to be the same as \$handle.

set_var(\$varname, \$value = "")

The functions sets the inital value of a variable. It may be called with either a \$varname/\$value pair or with a hash of \$varname/\$value pairs.

subst(\$handle)

The function returns the value of the variable named \$\text{handle}, with all defined variable values filled in. The resulting string is not "finished", that is, the unresolved variable name policy has not been applied yet.

psubst(\$handle)

This is a shorthand for print \$this->subst(\$handle).

parse(\$target, \$handle, \$append = false)

The function substitutes the values of all defined variables in the variable named \$\text{handle} and stores or appends

the result in the variable named \$target.

If \$handle is an array of variable names, \$append is ignored. The variables named by \$handle are being sequentially substituted and the result of each substitution step is stored in \$target. The resulting substitution is available in the variable named by \$target, as is each intermediate step for the next \$handle in sequence.

pparse(\$target, \$handle, \$append = false)

A shorthand for print \$this->parse(...).

get_vars()

Returns a hash of all defined values, keyed by their names.

get_var(\$varname)

Returns the value of the variable named by \$varname. If \$varname references a file and that file has not been loaded, yet, the variable will be reported as empty.

When called with an array of variable names, an hash of values, keyed by their names, will be returned.

get_undefined(\$handle)

The function will return a hash of unresolved variable names in \$handle, keyed by their names (that is, the hash has the form \$a[\$name] = \$name).

finish(\$str)

The function will returned the finished version of \$str, that is, the policy regarding unresolved variable names will be applied to \$str.

p(\$varname)

The function will print the finished version of the value of the variable named by \$varname.

get(\$varname)

The function will return the finished version of the value of the variable named by \$varname.

haltmsg(\$msg)

This function can be overridden by your subclass of Template. It will be called with an error message to print.

Internal instance methods

filename(\$filename)

When called with a relative pathname, this function will return the pathname with \$this->root prepended. Absolute pathnames are taken unchanged.

The resulting filename must exist, or an error is generated.

varname(\$varname)

The function will construct a variable name regexp for a given variable name.

loadfile(\$handle)

If a variable is undefined or empty and is backed by a filename, the backing file will be loaded and the files contents will be assigned as the variables value.

halt(\$msg)

This function is called whenever an error occurs and will handle the error according to the policy defined in \$this->halt_on_error.

Example

The class manages a set of variables which are text strings. These strings may contain references to other variables in the form of "{variable}". When parsed or substituted, a variable reference is being replaced by the value of that variable.

A variable value may be defined manually by calling set_var("name", "value"); or it may be defined from a file by calling set_file("name", "filename.ihtml");. In the latter case, the contents of the file are being loaded when needed (as late as possible) and set as the value of that variable.

A third way to define a variable value is to call <code>set_block("parent", "block", "name");</code>. In this case, the variable named <code>parent</code> is being searched for a block that starts with <!-- BEGIN block --> and ends with <!-- END block -->. This string is removed from the variable <code>parent</code> and assigned to the variable named <code>block</code>. In <code>parent</code>, a variable reference to <code>name</code> is placed instead. If the optional parameter <code>"name"</code> is left out, <code>"block"</code> is being used instead.

Use Template directy or define a subclass of Template as needed.

Define a template file named page.ihtml as follows:

This file contains a reference to the variable pagetitle and a reference to the variable named out. Another template file, named box.ihtml, contains a block named row with three variable references {TITLE}, {NUM} and {BIGNUM}:

The following php3 file demonstrates how to use these templates:

```
<?php
  include("./template.inc");
  # create Template instance called $t
 $t = new Template("/page/to/webserver/template", "keep");
  # define variables named page and box, referencing files
 $t->set_file(array(
     "page" => "page.ihtml",
"box" => "box.ihtml"));
 # extract the block named "row" from "box", creating a
 # reference to {rows} in "box".
$t->set_block("box", "row", "rows");
  # define the variables TITLE and PAGETITLE
 n = i;
   $nn = $i*10;
   $t->set_var(array("NUM" => $n, "BIGNUM" => $nn));
$t->parse("rows", "row", true);
  # build out from box, then build out from page...
 $t->parse("OUT", array("box", "page"));
  # finish out and print it.
 $t->p("OUT");
?>
<hr>
<?php
 # report leftover variables, if any.
 print implode(", ", $t->get_undefined("rows"));
 2 >
```

Next Previous Contents

5. HTML Widgets Classes

5.1 Sql_Query

Sql_Query will generate a query form for simple table queries: A list of field names, comparision operators and input fields is presented. The user may search for any values in any of the presented columns using SQL standard operators. Multiple query conditions are possible and these conditions can be joined using AND and OR operations.

The number of query conditions can be made variable. If so, the user may shrink and grow the query widget using the appropriate buttons.

All button labels and other messages of the interface are variable and held in language dictionaries. Currently, *de* and *en* dictionaries are provided.

Instance variables

Accessible instance variables.

classname	Serialization helper: The name of this class.
persistent_slots	Serialization helper: Names of all persistent slots
conditions	Number of query conditions
input_size	Visible width of input fields
input_max	Useable width of input fields
method	Form method to GET or POST the form
lang	Language dictionary to use
translate	Flag: translate column names
container	Flag: create a master container
variable	Flag: create resize buttons

Internal instance variables.

dict	The GUI language dictionary.
compare	SQL comparison function dictionary.

Instance methods

Accessible instance methods

start()

Initialization function. Currently empty.

form(\$base, \$option, \$class, \$target)

The function will generate and return HTML for the SQL Query selection form. All variables in the form will start with the prefix \$base and have numeric indices appended after an underline character. It is possible to have multiple Sql_Query instances on a single page, if they use different base characters.

The function must know the field names of the SQL table that is to be queried. \$option can be either a simple array of these field names (\$translate set empty) or a hash field name to long name (\$translate set to on).

All tags in the generated form are tagged with a CSS stylesheet class, if \$class is set to a CSS classname. \$class is optional and if it is left empty, no class attributes are generated. \$target is the URL of the SQL Query form target. It is optional and if it is left empty, a self referencing form is generated (recommended).

The function returns a string containing the HTML to render the SQL Query selection form.

where(\$base, \$incr)

When the form() generated page is submitted, a lot of parameters have to be evaluated and transformed into a SQL where condition matching the user selections. The where() function takes care of all this; it just needs to be told which \$base prefix has been used in the form() call.

The \$incr parameter is optional and determines how many query condition rows are added or subtracted when the "More" and "Fewer" buttons are used. The default value is 1.

The function returns a string which can be successfully used behind a "where" keyword in a SQL query.

Internal instance methods

plain_where(\$base)

This function does all the work for where (), but does not resize the query condition window.

Example

The sql_Query class can be used directly. It is more useful when made persistent, so it is recommended that you add the line require("sqlquery.inc") to your prepend.php3 file where indicated in that file.

See the Table class in this section for a nice method to display and format query results. See the DB_Sql class (a core class) for a nice method to connect to databases.

The following code fragment is quite large, but contains a complete and working example using the Sql_Query, DB_Sql and Table classes to query a database table.

```
// We require() sqlquery.inc and table.inc in prepend.inc
  // to make this example work!
  page_open(array("sess" => "Example_Session"));
  $db = new DB_Example;
                              // That's a DB_Sql subclass.
  $t = new Table; // For formatting results
$t->heading = "on"; // We want table headings..
?>
<html>
<head><title>Testseite</title>
<style type="text/css"><!--
                font-family: arial, helvetica, sans-serif; color: #d33e30 }
h1
table.test
                background-color: #eeeeee
                font-family: arial, helvetica, sans-serif }
font-family: arial, helvetica, sans-serif }
th.test
td.test
table.query
              { background-color: #cccccc }
td.query
               font-face: arial, helvetica, sans-serif }
--></style>
```

```
</head>
<body bgcolor="#ffffff">
<h1>Testpage</h1>
  // the following fields are selectable
  $field = array(
                => "Login Name",
    "username"
                  => "Password",
    "password"
                 => "Permissions"
     'perms"
  );
  // When we hit this page the first time,
  // there is no $q.
  if (!isset($q))
    $q = new Sql_Query;
                               // We make one
    $q->conditions = 1;
                             // ... with a single condition (at first)
// ... column names are to be translated
    $q->translate = "on";
$q->container = "on";
                    = "on"; // ... with a nice container table
= "on"; // ... # of conditions is variable
= "en"; // ... in English, please
    $q->variable
    $q->lang
                               // and don't forget this!
    $sess->register("q");
  // When we hit that page a second time, the array named
  // by $base will be set and we must generate the $query.
  // Ah, and don't set $base to "q" when $q is your Sql_Query
  // object... :-
  if (isset($x))
    query = q->where(x, 1);
  // In any case we must display that form now. Note that the
  // "x" here and in the call to q->where must match.
  // Tag everything as a CSS "query" class.
  printf($q->form("x", $field, "query));
  printf("<hr>");
  // Do we have a valid query string?
  if ($query) {
    // Show that condition
    printf("Query Condition = %s<br>\n", $query);
    // Do that query
    $db->query("select * from auth user where ". $query);
    // Dump the results (tagged as CSS class test)
    printf("Query Results = %s<br>\n", $db->num_rows());
    $t->show_result($db, "test");
 page_close();
?>
</body>
</html>
```

5.2 Table and CSV_Table

The Table class is a neat way to format two-dimensional associative arrays of data or the results of a database query into a table. Table and its subclasses allow you to simply pass them either an array or a query result and they spit out the proper HTML for a table containing all the values. Table has some primitive filtering capabilities making it useful even without subclassing, but for the full power of Table you have to write your own subclass.

When used with the check option, it is assumed that the table is part of a HTML FORM element. Code is generated to create an INPUT TYPE=CHECKBOX element before each table row. The checkboxes will form an array indexed by row number. The name of the array will whatever you set the check instance variable to.

Exactly one of two types of possible column filtering take place when each table row is generated. If the fields instance variable is set, only the columns keyed by the named fields in that array are shown in that order. That is, if you fill in the fields instance variable with array("a", "c", "e"), only the columns a, c and e become part of the generated table.

If fields has not been set, all data columns are traversed with each() and all columns whose names match the regexp in filter are shown in the table. By default, this regular expression lets through all column names that start with an alphabetic character and continue with either alphanumeric characters or "_" (underscore). This default has been chosen, because the DB_Sql database class uses mysql_fetch_array() internally to get data from the database and this function returns all columns twice with both a numeric index and proper column names. The default filter will make all data show up only once and with proper column names.

Additionally, the map_cols instance variable provides column name remapping. If map_cols is set, it will remap the name of the found column with a new name.

For instance, a table with the following columns, fname, lname, and mydate can be remapped to First Name, Last Name, and Date using the following code (where \$t is your instantiated Table class object):

The map_cols instance variable also allows you to map column names to different languages using this same technique.

For derived classes, the instance variable add_extra has been added. If this variable is set, then the functions table_heading_row_add_extra() and table_row_add_extra() are called. In the Table class, these functions do nothing, but in derived classes override these functions to provide additional functionality that may be needed. For instance, hyperlinks to provide edit, delete, or view capabilities for that row could be easily added into these functions (in your derived Table class) allowing greater customization.

A subclass of Table, CSV_Table, is being provided to allow to create CSV representations of your data with minimal effort. CSV (comma separated values) can be imported by MySQL's LOAD DATA INFILE statement and many spreadsheet import functions.

The Table class now provides both high-level, mid-level and low-level functions through modularization. This allows programmers to use either the simplified high-level functionality or, depending on the degree of complexity needed, the power of the mid- or low-level functions. Every effort to maintain backwards compatibility has been applied. However, it would be a good idea to become familiar with the new functions if you use the Table class extensively. Typically, the high- and mid-level support functions begin with <code>show_</code> while the low-level functions do not.

Instance variables

Accessible instance variables.

classname	Serialization helper: The name of this class.
check	If set, the check option is active.
filter	A regular expression selecting the columns that are shown.
fields	A list of colum names that are shown.
heading	A flag; if set, a heading is being created.
map_cols	A list of colum names that remap (replace) the static column names.
add_extra	A flag; if set, the the extra functions for heading and rows are called.

Instance methods

High-level instance methods

```
show($ary, $class = '''')
```

Will format and print the two dimensional array (or hash) \$ary as a table according to the filtering rules explained above. If \$class is set, each HTML element will be tagged as belonging to the named class; this is useful with cascading style sheets.

```
show_page($ary, $start, $num, $class = "")
```

Just as show(), but will show only num elements starting at start.

```
show_result($db, $class = """)
```

Will format and print the result set of \$db. \$db is exspected to be a subclass of DB_Sql that has just been sent a query. Table will grab all available results from the result set of that query by calling \$db->next_record() repeatedly and format them into a table.

```
show_result_page($db, $start, $num, $class = "")
```

Just as show result(), but will show only num elements starting at start.

Mid-level instance methods

```
show table rows($ary, $class="")
```

Walks the passed array displaying each row of data as an HTML table row.

```
show table rows result($db, $class="")
```

Walks the passed database object displaying each record as an HTML table row.

```
show_table_page_rows($ary, $start, $num, $class="")
```

Walks the passed array displaying each row of data as an HTML table row. However, data does not start displaying until \$start element and end after \$num rows.

```
show_table_page_rows_result($db, $start, $num, $class="")
```

Walks the passed database object displaying each record as an HTML table row. However, data does not start displaying until \$start record and ends after \$num records have been displayed.

```
show table heading row($ary, $class="")
```

Uses the passed array to create an HTML header row.

```
show_table_heading_row_result($db, $class=""")
```

Uses the passed database object to create an HTML header row.

```
show table heading cells($data, $class="")
```

Walks the passed array and displays each item in an HTML table header cell.

```
show_table_cells($row, $row_key, $data, $class="")
```

Walks the passed array and displays each item in an HTML table cell.

Low-level instance methods

table_open(\$class = """)

This function can be overridden by a subclass of Table. It is called as the very first step in table creation and should output HTML that opens a table (for example printf("<table%s>\n", \$class?" class=\$class":"");).

table close()

This function can be overridden by a subclass of Table. It is called as the very last step in table creation and should output HTML that closes a table (for example printf("\n");/).

select_colnames(\$data)

Internal function to generate a list of column names.

table_heading_row(\$data, \$class = "")

Internal driver function to generate a table heading row.

table_heading_cell(\$col, \$val, \$class)

This function can be overridden by a subclass of Table. It is called each time a table heading cell is to be generated.

\$col is the current column number, \$val is the name of the column. \$class is the HTML CSS class of the element that is to be generated.

table heading cell open(\$class="")

Starts a header cell.

table heading cell close(\$class="")

Ends a header cell.

table_heading_row_add_extra(\$data, \$class="")

Virtual function for derived classes. This function is called after all header cells have been created. It allows the programmer to add additional HTML code to the header row before it is closed.

table row(\$data, \$class = "")

Internal driver function to generate a table row.

table_row_open(\$row, \$data, \$class = "")

This function can be overridden by a subclass of Table. It is called as the very first step in row creation and should output HTML that opens a table row.

\$row is the current row number. \$data is a hash of column name/value pairs for that row and \$class is an optional HTML CSS class name for all generated elements.

```
table_row_close()
```

This function can be overridden by a subclass of Table. It is called as the very last step in row creation and should output HTML that closes a table row.

```
table_cell($row, $cell, $key, $val, $class)
```

This function can be overridden by a subclass of Table. It is called each time a table cell is to be generated.

\$row is the current row number, \$cell is the current cell number. \$key is the current column name, \$val is the value of the cell. \$class is the HTML CSS class of the element that is to be generated.

```
table_cell_open($class="")
```

Starts a cell.

```
table_cell_close($class=""")
```

Ends a cell.

```
set checkbox heading($class="")
```

This function creates an empty header cell to coincide with the checkbox option for that column.

```
table_checkbox_cell($row, $row_key, $data, $class="")
```

Outputs HTML code to display a checkbox. This function runs if the member variable \$check has been set. \$check should be set to some key within the \$data array (ex: if \$data["myKey"], then set \$check="myKey").

```
set_checkbox($row, $row_key, $data, $class="")
```

Creates an HTML checkbox based on the passed data, only if the instance variable \$check is set.

Example

Table is not automatically included or prepended into each page. Include the table class into the pages that are to use Table. Then create an instance of Table:

```
<?php
  // Include Table
  require("table.inc");

// make a Table instance
$t = new Table;

// We want table headings to be printed.
$t->heading = "on";
```

Now create a two dimensional array or prepare a database query and have table print it.

```
// Create a database object
$db = new DB_Session;

// create a twodim array called $tab
$tab = $db->metadata("active_sessions");

// print that array
$t->show($tab, "metadata");
```

```
// prepare a database query
$db->query("select * from active_sessions");
// print that result
$t->show_result($db, "data");
```

5.3 Menu

Menu will generate a hierarchical menu of clickable items suitable as a navigation bar. Menu takes a tree definition of items as the basis for this navigation bar and knows which subtrees to fold, depending on the current position in the menu tree. Menu uses the current URL as presented in PHP_SELF to determine the current position in the menu tree automatically.

Menu does not depend on a hierarchical organisation of files in URL-space to generate a menu hierarchy. The organisation of menu items and the organisation of files in URL-space are in fact completely independent and Menu uses a mapping hash to derive a menu position from an URL. In the following class documentation we'll say URL when we mean files the latter and menustring when we mean the former. In the context of the Menu class, URLs are always relative URLs starting at the root of the local servers URL space, as we'll see them in PHP_SELF. They may look like /menu/index.php3. A menustring is usually numeric and all components have the same length, if necessary with leading zeroes. It may look like /001/007, denoting an item in main menu 1, submenu 7.

Instance variables

Accessible instance variables.

	1 1000 SS202 INSUMINO WILWOOD!
\$urlmap	Hash. Maps a relative URL as seenin PHP_SELF to a menustring.
\$map	Menustring. Current position in menu.
\$item	Hash of Hashes. The item hash is indexed bymenustring. For each menustring, an arbitrary number of itemattributes is stored. Menu does not use the item array,only Menu::get_cell() does, which you implement. Eachmenu item will automatically get an attribute url throughurlmap inversion from the class constructor and the name ofthe menu item should be stored as the title attribute. Youmay add other attributes such as textcolor and bgcolorand use them in your Menu::get_cell()implementation.
∥xn∩main ∣	Boolean. If set, the first menu item will neverbe shown. This menu item is always the menu root, adding anextra level of indentation.
Stitle	String. After calling <code>get_title()</code> , this variable contains the title of the page based on it's location in the menu hierarchy.
\$title_delim	String. Used to delimit (i.e., separate) components in the page title built by <code>get_title()</code> . Default is ":

Internal instance variables.

	Hash of arrays. The children is indexed bymenustring. For each menustring, an array of
\$children	childrenmenustrings is stored. This array is generated from urlmapin the class constructor and is used to determine which submenusare visible from the current map position.
	determine which submenusare visible from the current map position.
	Array of menustrings. The visible arraycontains the menustrings visible from the current mapposition. It
	is calculated by the Menu::get() and Menu::show() functions.

Instance methods

Accessible instance methods

Menu()

Constructor. Calls Menu::setup() internally.

show()

A shorthand notation for print \$this->get().

get()

This function will calculate the menu items visible from the current map position. The menu will be constructed by calling Menu::start_menu() first. For each visible menu item, Menu will check the current indentation level and the indentation level of the current menu cell. If the indentation level increases,

Menu::shift_in(\$oldlevel, \$level) is called once, if it decreases, Menu:shift_out(\$oldlevel, \$level) is called once.

After that, Menu::get_cell(\$number, \$level) is called once. The number is an index into the visible array.

After all menu cells have been drawn, Menu::end_menu() will be called once.

get cell(\$n, \$level)

You are expected to implement this function yourself. It should render a single menu item. You may use the visible and item arrays for that purpose: \$m = \$this->visible[\$n] will return a menu string and \$attr = \$this->item[\$m] is a hash of attributes for that menu string. \$hilite = (\$this->visible[\$n] == \$this->map) is true for the current menu item, which should be rendered in way to stand out from the rest of the menu items.

get_title()

This function will calculate the title of the current page based on the position of the current page in the menu hierarchy. This function uses \$this->title_delim to separate the components of the title. This function sets \$this>title to the calculated title and returns the title as a string.

setup()

This function initializes the internal arrays of Menu and should be called once from the constructor. It actually is the constructor, but language stupidity has it that PHP3 constructor names vary with class names, which means that you have to write a new constructor for each subclass and call this function manually.

Internal instance methods

normalize_pos(\$pos)

This function looks at the current URL in \$PHP_SELF and tried to translate this into a menustring. If the URL matches a menustring directly, this is easy.

If not, the current URL will be sequentially shortened by applying the dirname PHP function to it until it matches. This allows you to create a single menu item for all files in a directory.

split path(\$p)

This function is used in the construction of the set of visible menu items. Given a menustring or a pathname, it constructs a series of pathnames which converge elementwise against the given pathname. That is, given the

menustring /3/2, this function will return an array with the elements " " (the empty string), /3 and /3/2.

find_visible(\$r)

This function calculates the actual set of visible URLs given a series of converging pathnames. It will include the set of children of each of these pathnames in the visible set, then sort this set numerically.

Example

To use Menu, you must enable the require statement for menu.inc in prepend.php3. To use Menu_Button, you must enable the require statement for menu.inc and menu_button.inc in prepend.php3.

Use a subclass of either Menu or Menu_Button to create a menu. Define a class Example_Menu in your local.inc file with a number of menu items in it. Do not forget to build a constructor.

```
class Example_Menu extends Menu
  # Map of PHP_SELF URL strings to menu positions
  var $urlmap = array(
    "/menu/index.php3"
                          => " "
                          => "/1"
    "/menu/item1.php3"
                          => "/1/1",
    "/menu/item11.php3"
    "/menu/item12.php3"
                          => "/1/2"
                          => "/1/3",
    "/menu/item13.php3"
    "/menu/item2.php3"
                          => "/2"
                          => "/2/1"
    "/menu/item21.php3"
                          => "/2/2"
    "/menu/item22.php3"
    "/menu/item221.php3" => "/2/2/1"
    "/menu/item222.php3" => "/2/2/2",
    "/menu/item23.php3"
                          => "/2/3"
                          => "/2/4"
    "/menu/item24.php3"
  # Information about each menu item
 var $item = array(
          => array("title" => "Main"),
            => array("title" => "Text 1")
    "/1"
    "/1/1" => array("title" => "Text 1.1"),
    "/1/2" => array("title" => "Text 1.2"),
           => array("title" => "Text 1.3"),
    "/1/3"
           => array("title" => "Text 2"),
=> array("title" => "Text 2.1"),
    "/2"
    "/2/1"
    "/2/2"
            => array("title" => "Text 2.2")
    "/2/2/1"=> array("title" => "Text 2.2"),

"/2/2/1"=> array("title" => "Text 2.2.1"),
    "/2/2/2"=> array("title" => "Text 2.2.2"),
    "/2/3" => array("title" => "Text 2.3"),
            => array("title" => "Text 2.4")
    "/2/4"
  );
  function Example_Menu() {
    $this->setup();
}
```

In each of your files mentioned in the above urlmap, create an instance of Example_Menu and call the show() method of that instance.

5.4 Form

The form class (sometimes called OOH Forms) is a convenience library for dealing with html forms. It provides Javascript and server-side form validation, and is customizable and extensible.

Using OOH Forms

The OOH Forms library consists of five files: oohforms.inc of_checkbox.inc of_radio.inc of_select.inc of_text.inc of_text.inc of_text.inc of_text.inc of_text.inc of_text.inc of_text.inc oohforms.inc automatically includes the others. You may wish to modify this so you can manually include the files for just the form elements you use. Or you may wish to cut and paste the contents of the element files into oohforms.inc to save the overhead of multiple includes. Determining the appropriate configuration of the files for your site is left an exercise for the reader; for most purposes require("oohforms.inc") will suffice.

In general, the structure of a page that uses oohforms is as follows:

```
require("oohforms.inc");
                                   // include the library
$f = new form;
                                    // create a form object
                            // set up form elements
$f->add_element(...);
$f->add element(...);
$f->add_element(...);
if ($submitname)
                                   // Is there data to process?
                                   // Is the data valid?
// No; Display error
  if (\$err = \$f->validate()) {
    echo $err;
    $f->load_defaults(); // Load form with submitted data
  else {
    /* Process data */
                                   // Data ok; Do something with it
$f->start(...);
                                   // Start displaying form
$f->show_element(...);
                            // Show elements
$f->show_element(...);
$f->show_element(...);
$->finish();
                                    // Finish form
```

There are obviously many variations on this theme, but that covers the basics. Specific methods are documented below.

start(\$jvsname,\$method,\$action, \$target)

Outputs the initial <form> tag and sets up some initial state needed by the class. All of the arguments are optional, though you'll usually want to use at least one in order to get Javascript validation. \$jvsname is an arbitrary string used to link the Javascript to the form; if it is empty (the default), no Javascript validation is provided. \$method is the HTTP method used to submit the form; default is "POST". \$action is the URL that receives the form submission; default is \$PHP_SELF. \$target is the frame target for the form results; default is _self.

finish(\$after,\$before)

Outputs the any hidden fields that were added to the form, the final </form> tag, then the Javascript validation function (if necessary). \$after and \$before are both optional; if either is a nonempty string it is output as additional Javascript to be run on submission of the form, either before or after the validation code. Though the order of the arguments may seem counterintuitive, it makes the common case easier to type; in general you'll want to wait until after the validation has taken place to do anything fancy with Javascript. Note that unlike with validation, OOH Forms has no way of giving you server side functionality equivalent to the Javascript you use here.

add_element(\$element)

add_element is used to define the attributes of a particular form element so that the other class methods can use and manipulate it properly. add_element takes exactly one argument: an associate array whose key value pairs are used to define the form element type and it's various attributes. Some of these attributes correspond to html attributes, while others are needed for the value added features of oohforms. The attributes and the syntax and semantics of the values they take are documented below; note that not all element types use all of the attributes

type

```
The type of element this is; can be "submit", "hidden", "text", "textarea", "select", "radio", "checkbox", or "file".
```

name

A string naming this element. This name will be used as an argument to other methods and will be the name="" value in the generated html (and hence the variable name in PHP). **Do not** append [] to the name if you want an array valued element; set the multiple attribute instead.

value

The default value of this form element. If the form element has the multiple attribute set, value can be an array. If the this is a select element, value can refer to either the textual name (label in the options array) or the submission value (value in options).

multiple

A flag to tell oohforms to assume this element is array valued. The use of this flag is most straightforward with select elements, but it can be use with text and checkbox elements as well. See the show_element documentation for more information about how oohforms deals with such elements.

extrahtml

Extra html code that is inserted verbatim into the opening form tag. For select elements, the extra html goes into the select tag, not the option tags.

size

For text elements, used to set the html size attribute that gives the width in characters of the text entry box. For select elements, the size (number of options visible at once) of the selection box. Validation is only performed on select elements if size is set to 1, since select validation doesn't make much sense if you can see multiple options at once. For file elements, the maximum size file upload to accept.

pass

If set for a text element, renders the html as a password element, i.e. entry displays as asterisks.

src

If set for a submit element, convert to an image element and use the value of src as the source URL for the image.

maxlength

Used verbatim as the maxlength html attribute in text elements.

minlength

If length_e is set, this is the minimum length text element entry accepted by the validator.

length_e

If set, validate the text element to assure it has at least minlength characters. The value of length_e is the error string to be used in the event of failed validation.

valid e

If set, perform validation on a text, radio, or select element. For a text element, validation assures a match with valid_regex. radio element validation assures that one of the options in the group has been chosen. select validation only works for select elements with multiple unset and size equal to 1; the validator will not accept the first option of accept menu, assuming that it is some sort of prompt (e.g. "Please select an item"). In all cases, the value of valid_e is the error string used for failed validations.

valid regex

Regular expression used to validate entry into a test field if valid_e is set. Note that if you must use ^...\$ if you want the regex to match the whole entry.

icase

If set, regex matching is case insensitive.

checked

Only used for a checkbox element that does not have multiple set. If checked is set, the element will display as checked.

rows

Used verbatim as the rows= element in a textarea element.

cols

Used verbatim as the cols= element in a textarea element.

wrap

Used verbatim as the wrap= element in a textarea element.

options

Array of options to be displayed in a select element. If the elements of the array are simple values (strings or numbers), they are simply displayed verbatim and used as the value for that particular option.

The elements may themselves be associate arrays with keys "label" and "value". In that case, the value of "label" is displayed and the value of "value" is used on submission.

Examples:

```
$f->add element(array("type"=>"text",
                                  "name"=>"foo",
                                  "valid_regex"=>"^[a-z]*$",
                                  "valid_e"=>"Letters only",
                                  "icase"=>1,
                                  "value"=>"bar"));
$f->add_element(array("type"=>"checkbox",
                                  "name"=>"compress",
                                  "multiple"=>1));
$f->add_element(array("type"=>"textarea",
                                  "name"=>"comment",
                                  "rows"=>6,
                                  "cols"=>40
                                  "value"=>""));
$0 = array(array("label"=>"Please Select","value"=>0),
            array("label"=>"Apple", "value"=>1),
array("label"=>"Orange", "value"=>2),
            array("label"=>"Pear", "value"=>3),
array("label"=>"Grape", "value"=>4));
$f->add_element(array("type"=>"select",
                                  "name"=>"menu",
                                  "options"=>$o,
                                   "size"=>1,
                                  "valid_e"=>"Please select a fruit",
                                  "value"=>"apple"));
```

show_element(\$name,\$value)

Outputs the form element named \$name. Usually, the second argument is not used. It is always necessary for radio elements and checkbox elements with the multiple attribute set, since many of these may have the same name. It also must be used for submit elements to label the submission button; the value attribute is not used for submit elements. For other elements that may be array valued (notably text elements) multiple calls to show_element will show successive values.

load_defaults(\$element_list)

Sets the default value of form elements to the value of the PHP variables with the same name. This is generally used to redisplay a form with the same values which were submitted. The argument is optional; if given it is an array of element names; only these elements are affected.

validate(\$result,\$element list)

Validates a form submission. If all of the elements are valid, return \$result, otherwise return the relevant error message (the valid_e or length_e attribute of some form element). \$result defaults to false. The second argument is also optional; it is an array of names of elements to validate.

freeze(\$element_list)

Freezes the form elements whose names are given in the array passed as the argument. If no argument is given, freeze all of the elements. Frozen elements are rendered as plain, static html rather than form widgets. This static rendering is accompanied by appropriate hidden elements to simulate the affect of using the unfrozen version of the element.

Customizing OOH Forms

Since OOH Forms is object oriented, it can be easily customized by extending the classes that define the element types.

In general, you must make sure your derived class has a constructor and you may override any of the self_* functions of of_element. The source for the existing elements is the best documentation for how to do this properly, but a few general notes follow.

self_show(\$val,\$which)

Display an instance of this element unfrozen. \$val is the \$value argument of show_element if there was one; \$which can be used as an index for array valued elements; it is equal to the number of times show_element has been called for this element previously. This function must return the number of hidden tags output.

self_show_frozen(\$val,\$which)

Display an instance of this element frozen. In addition to the html to show the frozen element, you must output tags for hidden fields to duplicate the effect of submitting an unfrozen element of this type. The function must return the number of hidden tags output;

self_validate(\$val)

Validate \$val for this element. If it is valid, return false, otherwise return the relevant error string.

self_print_js(\$ndx_array)

Print out Javascript code to validate this element. \$ndx_array is an array of the indices of the elements to be validated as used in the Javascript form.element[] array. This is needed since the extra [] in array valued element names confuses Javascript.

self_load_defaults(\$val)

Set the default value for this element to \$val. Usually the default definition of this function, which just copies \$val to \$this->value is all that is needed, but there may be special cases that need to do something else. See the implementation of the checkbox element for an example.

5.5 tpl_form

The tpl_form class is intended to provide a general framework for HTML form deployment. It heavily depends on OOH Forms library, so it is required that you read and understand the relative documentation.

The main idea is that you create a black box by sub-classing tpl_form, provide some HTML code mixed with OOH Forms calls to actually render the form itself. Your application will then use that black box to obtain some input from the user. Your application doesn't have to know how to handle user input, nor how to validate input data, since internal methods will take care of that.

This approach is very similar (I think) to OOH Forms one, only at a higher level. OOH Forms elements have no way to communicate with each other, and are only able to perform "simple" checks on data integrity, while tpl_form adds a consistent interface for complex data evaluation and processing.

Furthermore, the get_default_values and set_default_values methods can be used to maintain user input between sessions, without worrying about serialization of form variables (a BAD THING(tm)), using an hash array containing field names and values.

You'll note that an array is used to share data with the application. You may object this is kinda futile, since all user input data may be found in global vars and http_post or http_get global hashes. This is true, and in the general case you'll pass back and forth an empty array. The values variable is intended for use in very complex data-entry setup, where a form behaviour may depend on previous data entered by the user. In this case, if all forms cooperate reading and writing to values hash array, final result may be constructed step by step across multiple HTML pages.

Instance variables

Internal instance variables.

	Name of the class. Used for serialization AND in display to determine the filename of template containing HTML/php code needed to actually render the form.	
error	Contains error messages generated by validate and validate_input methods.values	This is a sort of "shared memory area" between the form and the application. Is read in init method and passed back in get_values method.
form data	Holds form info (Form object).	
has defaults	Flag, form default values were passed via set_default_values method. Should not be tampered with by the user.	

Instance methods

Accessible instance methods

init(\$values)

This is a sort of a constructor for the class. \$values is an hash array intended to store form values to be passed back to the application via get_values method.

get_default_values()

Returns an array containing all data submitted by the user for the form. This array is intended to be passed to set_defaults_values some time later.

set_default_values(\$fv)

Restore form defaults from an array as returned by get_default_values.

display()

Actually display form fields. This method should not be overridden in descendants. User should instead provide a file named as the derived class and with ".ihtml" extension which will be automatically included.

get_values()

This method should not be overridden. It is intended as the main interface between the application and the form. Once the form has been properly derived to suit designer's needs, application calls <code>get_values</code> and receives back the array passed to <code>init</code>, eventually modified by <code>process_input</code> method, or <code>false</code> if user input is invalid. In that latter case, the application should call <code>display</code> to (re)present the form to the user, eventually filled with proper default values.

clear()

Sort of a "destructor". There should no real need to call it, except maybe freeing some memory. May be called from the application, otherwise is not executed. Returns true.

Internal instance methods

setup()

Init the Form object, which will contain all fields info. The hidden field form_name, automatically added by this routine, is used by other methods to determine if form has already been submitted by the user. You shouldn't override this in descendants, use setup_fields instead. Returns true.

setup_fields()

Override this method in order to provide form fields definition that suit your needs.

validate()

Validates user input. This method should not be overridden in descendants. See validate_input instead. Returns false on error and sets error variable accordingly.

validate_input()

This method should be overridden in descendants, in order to provided complex validation methods (i.e. field2 should not be empty IF field1 == "other"). Should return false on error and set error variable with a sensible error message.

process()

Process user data. This method should not be overridden by descendants. See process_input and process default instead. Returns true on success, false otherwise.

process_input()

This method should be overridden in descendants. It is executed after validation has taken place. The data passed to the form could be used to fill values array.

process default()

This method should be overridden in descendants. It is executed when form validation fails or before presenting the form for the first time. Should be used to bypass form displaying if data can be extracted from previous actions, divination, penguin fly watching or whatever.

Example

Suppose you have a form that the user should fill with her (eheh) name and e-mail. You want to check wether this e-mail is valid, or your blind date setup is lost. A... er... simple regular expression for validating syntactically the e-mail is presented in the example code below.

```
$this->form_data->add_element(array(
  "type"=>"text",
  "name"=>"email",
  "valid_e"=>"Syntax error in E-Mail address.",
  "valid_regex"=>"^([-a-zA-Z0-9.]+@[-a-zA-Z0-9]+(\.[-a-zA-Z0-9]+)+)*$"
));
```

Now, this piece of code should do the job, but since you're feeling very paranoid today, you'd also like to validate the host name part of the address with DNS. So, you put together some code which takes an hostname in input and reports true on valid hostname, false otherwise (HINT: on PHP Code Exchange you should find a procedure for "active"

email validation).

Now that you have your shining new code, you can check the address. The user fills out the form, you parse user input, no syntax errors, time to call your mycheckhost from the application. If the function is ok update your database, else load defaults into the form, display again, close the page, goodbye.

I've done something similar for MANY forms, some of them with very complex validation procedures, and I found that was too easy producing very bad and unreadable code (well, I actually realized that the first time I had to change some logic in data validation...).

tpl_form should provide a solid framework to build your forms with, and all the code will be self-contained and separated from main application logic. Hope you'll like it.

Time to see some code. First of all, class declaration, sub-classing tpl_form:

```
class myform extends tpl_form {
 var $classname = "myform";
  function setup_fields()
    $this->form_data->add_element(array(
     "name"=>"email",
     ..., // See previous code snippet
    $this->form_data->add_element(array(
     "name"=>"submit",
     "type"=>"submit"
     "value"=>"submit"
    ));
  function validate_input() {
    global $email;
    list($uname, $hostname) = split("@", $email);
    if (! mycheckhost($hostname)) {
      $this->error = sprintf("Sorry, unknown host %s, try again.", $hostname);
      return false;
    // Additional checks here...
    return true;
}
```

You shuld provide a file myform.ihtml with HTML and php code to render the form. Minimalistic example:

```
<html>
<body>
<?php
$this->form_data->start_form($this->classname, "POST", $sess->self_url(), "");
printf("%s<br/>br>\n", $this->error);
$this->form_data->show_element("email");
printf("<br>\n");
$this->form_data->show_element("submit");
$this->form_data->show_element("submit");
$this->form_data->finish();
?>
</body>
</html>
```

Your tpl_form class is complete, and will only need a little work on the artistic side... 8-) To use your brand new class, include class definition code into your application, then...

```
// some memory...
global $email;
// process your data at will
} else {
  $mf->display();
}
```

Hope this very little example does help in understanding the real power of tpl_form class, at least in terms of rapid designing and code partitioning.

5.6 Tree

The Tree class can render tree structures such as directory hierarchies and menu structures as HTML. The structure must be given to Tree as an nested array of arrays of arbitrary depth.

The idea of Tree is, that there are several mathematical models a tree could be viewed: One model is a data structure like nested arrays or a pointer structure from where you can print multidimensional graphics and can do other neat things like deleting one part of the tree or inserting a whole subtree. But you can also imagine a tree as a one dimensional string or as a sequence of function calls (which is nearly the same in the mathematical sense).

To generate HTML-code from a tree-structure it is like this: You need at the end a one-dimensional string, which tells the browser what to do. The Tree class assists you in generating this string in this way, that it will go through the whole tree and call several functions on every stage trough the way. It will be your task to change the functions, so that a nice layout will be generated.

Instance variables

	• .	
Accessible	inctance	Wartablee
	HISTAIRCE	variannes.

classname	Serialization helper: The name of this class.
delimiter	a char for truncating the "path"
tree	an array of an array of an array
outp	the "output"
prfx, sufx, flag	internal - some helpers to create outp

Instance methods

Accessible instance methods

build_tree()

This function is completely user driven! You have to create an array with the structure described below. See the example for details.

Don't be shy to create some own functions which are called by build_tree() - e.g. for recursive calls.

go_trough_tree(\$key="",\$path="",\$depth=0,\$lcount=0,\$pcount=0)

This is the most important function of this class. It will call the output functions in the right order with the correct parameters.

All variables are optional. The parameters are perhaps useful, if you want to display only partial trees, but this is

not supported now.

path_to_index (&\$path,\$key=""")

This function is mostly used internally, but could be useful for you to generate \$this->tree. This function generates a PHP3 associate array-index string from a path, which is also a string but truncated by \$this->delimiter. If \$key is given, it will be added to \$path (minds empty path and so on).

Example:

```
$t->delimiter="/";
$path= "usr/local/lib";
## $path must be given as a var, because it is called by reference!
$bla = $t->path_to_index($path,"etc");
## $path is now "usr/local/lib/etc"
## $bla is now ["usr"]["local"]["lib"]["etc"]
```

path_to_parent (&\$path)

This function isn't used internally, but could be useful for you during generating the output tree. It will remove one from the depth of the path.

Example:

```
$t->delimiter="/";
$path= "usr/local/lib";
$bla = $t->path_to_parent($path);

## $path is now "usr/local"
## $bla is now ["usr"]["local"]
```

path_add (\$path,\$key)

This function is the 'non-call-by-reference-version' of path_to_index. It will add the \$key to the path and return it.

path_sub (\$path)

This function is the 'non-call-by-reference-version' of path_to_parent. It will find the parent of path and return it.

path_index (\$path)

This function is the 'non-call-by-reference-version' of path_to_index(). It will return the associate key to the tree described by path.

starttree ()

This function is called by go_trough_tree() at the beginning of the output of a tree.

All *tree-functions are called by go_trough_tree(), but it's your turn, to give them a nice layout. I think it is possible to generate nearly every kind of tree-layout with this. Have a look at the variables: E.g. \$depth makes it possible to handle every "level" in another manner.

growtree (\$key,\$value,\$path,\$depth,\$count,\$pcount)

This function is called by go_trough_tree() at the beginning of the output of a tree.

It is called every time, when <code>go_trough_tree()</code> will call itself recursively. You could also say it is called, when the current item has a successor.

leaftree (\$key,\$value,\$path,\$depth,\$count,\$pcount)

This function is called, when the current item has *no* successor.

shrinktree (\$key,\$depth)

This function is the "opposite" of growtree(). It is called every time, when the current item was the last item in this sub-list.

endtree()

Called when leaving tree.

The Tree Array

As said above, before you call <code>go_trough_tree()</code>, first <code>\$tree</code> must be generated.

\$tree consists of nested arrays of arbitrary depth. An example:

```
$t= new Tree;
$t->tree = array(
                 "usr" => array(
                          => "allowed",
                   0
                   "lib" => "forbidden",
                   "local" => "allowed",
                          => "forbidden",
                   "bin"
                   "etc"
                           => array(
                     0
                              => "allowed".
                     "hosts" => "forbidden",
                     "mailcap"=> "allowed"
                   "var"
                           => "allowed"
                           => "allowed"
                   "tmp"
                 root" =>"forbidden"
               );
$t->go through tree();
print $t->outp;
```

This is a completely recursive structure and I think, it is clear, how to create it with a recursive call of a function. If not, see the example below.

One little quirk has to be explained, because it is a little bit confusing: the array name 0 (zero) is used for the value of the parent element. As shown in the example, an element with children (for example "etc") cannot have attributes (such as "allowed"). Instead the value of this element is stored in a pseudo-child named 0. If this element is not present, it will have the value "Array" (perhaps something that should be changed).

The output of this example if you don't change the output-functions will look like this:

```
| \--- mailcap->'allowed' : 'usr^etc^mailcap' (3) [3/3]
| O--- var->'allowed' : 'usr^var' (2) [6/7]
| \--- tmp->'allowed' : 'usr^tmp' (2) [7/7]
| \--- root->'forbidden' : 'root' (1) [2/2]
```

Looks a bit confusing. From left to right the fields are

- The *index-name* of the current field
- The value of this field
- The *full path* to this field (see path_to_*-functions)
- The current depth or level
- The current *element number*. See below to understand, why it will begin sometimes with "2" in this example!
- The *number of elements* in the subtree at this depth

Example

My example is just going trough the directory structure of your hard disk.

The following code could read it:

```
class dir Tree extends Tree {
    var $classname = "dir_Tree";
    var $delimiter="/";
    var $tdat;
    function build_tree ($path=".")
        $this->tree=$this->recurs_dir($path,0);
    ## This example code can read and output 1000 directory entries with
    ## many subdirs in about 20 seconds on my system (P200, 64 MB);
    ## 220 dir entries with a maximum depth of 4 are read in 2 seconds.
    ## This is ok. :)
    function recurs_dir ($path,$depth) {
    GLOBAL $flap_out;
        $d=opendir($path);
        while ( $name=readdir($d) ) {
             $pathname=$path . $this->delimiter . $name;
             if (is_dir($pathname) && !ereg("\.\.?",$pathname)) {
                if (isset($flap_out[$pathname])) {
                     $array[$name]=$this->recurs_dir($pathname,$depth+1);
                 \sharp ATTENTION: It is IMPORTANT fill the [0] array
                 # *after* filling the rest of the array!
                 $array[$name][0]=$pathname;
             } else {
                 $array[$name]=$pathname;
        closedir($d);
        return($array);
    ## FLAPPING IN and OUT
    ## This is used to create an array which includes
    ## all sub-paths which should be showed
    function flapping ($path) {
    GLOBAL $flap out;
        if ($path)
             if (is_dir($path)) {
                if (isset($flap_out[$path])) {
                    unset($flap_out[$path]);
                 } else {
```

```
$flap_out[$path]=$name;
}

}

$t= new dir_Tree;
$t->flapping($val); ## $val is given by GET-method, see *tree()-functions
$t->build_tree();
$t->go_through_tree();
print $t->outp;
```

With this code it is very easy to flap in and out whole parts of the tree. Send the path via GET-method and put this path in flapping(). The whole \$flap_out-array must be persistent (e.g. via session). Perhaps you can program a garbage collection, which will look into \$flap_out and check for paths that already exist?

Known Bugs / Tips

There is one known bug: If a name of a subpath contains the \$delimiter-string. This cannot be solved correctly and you have to look for it when you create the tree.

The same thing is with the value [0] (zero) of a sub-array. This element is always used as the attribute of the parent element.

A good tip: when you build your tree recursively then the [0]-index must be filled *after* the subtree is returned from recursive call. See in the example above what I mean. I think it's a PHP3 specialty.

Also it is possible that not every name could be inserted into the associate index-field (Control-chars etc.), but this is untested.

5.7 STRINGS2 function set

This is a set of functions, which are used very often by me.

They are so easy, that I now stop describing and simply insert the code. Perhaps the next revision of this set I will replace it with a better description:

```
<?php
## Strings2-Functions
## Copyright (c) 1998-2000 Alex 'SSilk' Aulbach
## These Functions are very practical and if I could program ## C a little bit better it will be placed directly in PHP3.
## But I can't... :-}
##
## Have you ever worried about such constructs like
       echo ($faxnumber) ? sprintf("Fax: %s",$faxnumber) : "";
##
##
## This functionset could help you to replace those constructs by
## p_iftrue($faxnumber,"Fax: %s");
## which is nearly the half of typing and looks more clear and solves
## an error if $faxnumber is unset.
##
function o_iftrue ($val,$str)
         if (isset($val) && $val)
                   return(sprintf($str,$val));
function p iftrue ($val,$str) {
```

```
print o_iftrue($val,$str);
##
## Output "One or More"
##
## This function is good if you want to differ a output by number:
##
    e.g. o_1or2($q->num_rows(),
                   Found only one matching record",
##
                   "Found %s records");
                                       "Found only one matching record"
## Will output if num_rows() is 1:
                                       "Found 200 records"
                                 200:
##
## if $val is empty() or "" a blank string will be returned!
function o_lor2 ($val,$str1,$str2) {
         if (isset($val) && $val) {
                 if (1==$val)
                          return(sprintf($str1,$val));
                  } else {
                          return(sprintf($str2,$val));
          else
                 return(false);
function p_lor2 ($val,$str1,$str2) {
        print o_lor2 ($val,$str1,$str2);
## This is for the case, that you want to output something
## if $val is false e.g.
## p_0or1($faxnumber, "THERE IS NO FAXNUMBER", "Faxnumber: %s");
function o_0or1 ($val,$str1,$str2)
          if (empty($val) || !$val)
                  if (isset($val)) {
                          return(sprintf($str1,$val));
                  } else {
                          return($str1);
          else
                 return(sprintf($str2,$val));
function p_0or1 ($val,$str1,$str2)
        print o_0or1 ($val,$str1,$str2);
## Replaces all blank-chars with
## This function is used, when you are not willing to let the browser ## break your lines an can be used instead of <NOBR>-Tag
## as very compatible replacement
     can also be replaced by a true whitespace which has in
## ISO-latin-1 the code 160
function o nonbsp ($val)
        return(ereg_replace("[[:blank:]\n\r]"," ",$val));
function p_nonbsp ($val) {
        print o_nonbsp($val);
?>
```

Next Previous Contents

Next Previous Contents

6. Acknowledgments

The initial idea on how to do serialization was contributed by KH Wild to the php3 mailing list. It was limited to serializing arrays of at most three dimensions, though. We worked on his solution, improving it to arrays of arbitrary depth and later rewrote the function from scratch, turning it upside down. Our new serialization code can now handle any first order data type available to PHP and is easily extensible. It is also encapsulated in a class, keeping the name space clean. While we were at it, we made session cookies more secure by not using uniquid() directly, but a md5() hash of uniqid().

Cameron Taggart and Guarneri Carmelo contributed ODBC support. Szandor van Verseveld contributed PostgreSQL support. Scott McMullan contributed some nice ideas for cleanup and is working on Sybase support.

Sascha Schumann contributed much time developing and extending PHPLIB, including but not limited to mSQL/Sybase support, general storage container support, shared memory and LDAP support.

Alexander Aulbach submitted his Tree class. Jay Bloodworth contributed his excellent OOH Forms library for form generation and input validation.

A lot of people provided helpful hints and occasionally patches. Please see the file CREDITS for a complete list of contributors, testers and inspirations.

Next Previous Contents



EasyBMP Cross-Platform Windows Bitmap Library: Change Log Library Author(s): Paul Macklin Library License: BSD (revised). See the BSD (revised) license.txt file for further information. Copyright: 2005-6 by the EasyBMP Project Email: macklin01@users.sourceforge.net Support: http://easybmp.sourceforge.net All changes by Paul Macklin unless otherwise noted. *_____* Version: 0.50 Date: 1-31-2005 None! (first release) *_____* Version: 0.51 Date: 2-14-2005 Added full 32-bit BMP file support Took out annoying "colors: " message from BMP8 initialization from scratch Added more license and copyright info to each file Added change log to library To do next: Should update the error messages for the initializations Should simplify the reading and writing code *_____* Version: 0.52 Date: 2-19-2005 Fixed a minor bug in the MakeGreyscalePalette function where the 0 color turned out to be (255,255,255), rather than (0,0,0)Updated standard colors for 4-bit, 8-bit, and 24-bit

Version: 0.53 Date: 2-27-2005

Fixed unsigned / signed problem that VS.net shows

Tried fix of line 186 in EasyBMP_BMP4.h file. If it works, I'll apply it consistently. I think that VS.net wants us to clear char* blah, then char = new blah [size], just like the old days for g++.

Removed EasyBMP_StandardColors.h from standard package

Version: 0.54 Date: 2-27-2005

The fix of line 186 in EasyBMP_BMP4.h file appears to have worked. I applied it through the remainder of the code. Hopefully, this should ensure Visual Studio.Net compatibility.

Fixed some typos in the comment lines

Version: 0.55 Date: 5-2-2005

Introduced RGBApixel struct.

Introduced BMFH, BMIH, and BMP classes.

Deprecated all old code to *_legacy.h.

Rewrote EasyBMP_VariousBMPutilities.h to use the new BMP class.

Version: 0.56 Date: 5-4-2005

Made Width, Height, and BitDepth private members and added functions for accessing them.

Made a new function, SetBitDepth, as the only means to change the bit depth. It will create/resize a palette as necessary. This simplifies the WriteToFile code, as well as any palette altering algorithms. (All algorithms can now assume that a properly-sized palette exists.) This will help improve code stability greatly.

Made a new function, SetSize, as the only way to change the width and height of the image.

Eliminated useless HasPalette and NumberOfColors members, and added TellNumberOfColors() function.

Updated EasyBMP_VariousBMPutilities.h to respect privacy of data members. Version: 0.57 Date: 5-8-2005 Removed fclose(fp) lines from EasyBMP_BMP.h and EasyBMP_VariousBMPutilities.h whenever (!fp) occurs, to avoid a crash when trying to close a non-existant file. Added a line to set bmfh.bfType = 0; to getBMFH() routine in the case where (!fp) occurs, so that a nonexistant file doesn't falsely show up as a bitmap file. Made error messages in BMP::ReadFromFile(char*) more meaningful, since Argh! doesn't help much. :-) Made ReadFromFile operations safer: can deal more effectively with corrupted and/or truncated files by adding the new SafeFread() wrapper function. Moved all change-log entries to the change log to make the source file tidier. Removed all references to Palettes: renamed them to ColorTables. *_____* Version: 0.58 Date: 5-13-2005 Rewrote ReadFromFile() to fix program crashes on reading 4-bit files. (*grumble* I can't believe there was such a bug in such a late version! */grumble*) Added support to ReadFromFile() for reading 1-bit files. Rewrote ReadFromFile() to avoid trying to read bitmap files of depths other than 1, 4, 8, 24, and 32 bits. Tested reading 4-bit files of width 0,1,2, and 3 (modulo 4), and 1-bit files of width 0,1,2,3,4,5,6, and 7 (modulo 8) *_____* Version: 0.59 Date: 5-15-2005

Made ReadFromFile() more robust. Evidently, reading to the same temp variable all the time made it unstable when reading

many files. I would never have guessed. I instead declare BMIH and BMFH objects and read directly to their members. This appears to be more stable when dealing with many ReadFromFile() calls.

On a related note, made sure to not call SetSize(Width,Height), which is a bit recursive, as well as SetBitDepth(BitDepth). This appears to help stability, since these two functions were create precisely for the purpose of setting those variables values safely.

Made use of the boolean return value in SafeFread() to detect when files are obviously corrupted. Used this to have an early catch in ReadFromFile() and set it to a 1x1 1-bit image and exit.

Made ReadFromFile() stricter, in that it only reads recognized bit depths (1,4,8,24,32). Any other bit depth will prompt the routine to terminate and set it to a 1x1 1-bit file.

Added write support for 1-bit files.

Rewrote WriteToFile() for 4,8-bit files to match methods used for reading them.

Revised CreateStandardColorTable() and CreateGreyscaleColorTable() to add support for 1-bit files.

Rewrote WriteToFile() to be stricter in only writing known bit depths (1,4,8,24,32) and ignoring all others.

Version: 0.60 Date: 5-21-2005

Deprecated *_legacy.h files.

Tested library extensivey in linux with good results.

Made CreateGreyscaleColorTable() stricter, in that it exits if supplied a bit depth other than 1, 4, or 8.

Made cosmetic changes in EasyBMP_DataStructures.h to improve readability.

Made SetBitDepth() stricter, in that it will never allow a bitmap to be set to an unsupported bit depth. Only bit depths of 1, 4, 8, 24, or 32 are accepted.

Made SetSize() stricter, in that it will not allow negative widths or heights.

Made cosmetic changes in EasyBMP_BMP.h to improve readability.

Added a check in ReadFromFile() to see if the requested width or height is negative, a good sign of file corruption. In such a case, the file is set to a blank 1x1 1-bit file.

Added code to ReadFromFile() to set size to 1x1 and bit depth to 1-bit if the file was not found.

Version: 0.61 Date: 5-22-2005

Fixed awIndex typo in WriteToFile().

Replaced double BestDistance comparisons in WriteToFile() with int BestDistances (so as to do integer operations, rather than double operations). This gave a roughly 100% speedup in 8-bit, 4-bit, and 1-bit write operations on unoptimized (no compiler flags) code and a 30% speedup on optimized code.

Removed checks like if (BestDistance < 1) { k=256; } .. from WriteToFile(), as they give more overhead than savings in my testing. For 8-bit files, there was a slight gain by putting it back in with another method: while(k < 256 && BestDistance > 0).

Redefined StepSize in CreateGreyscaleColorTable() to give a better range of greys in 4-bit mode. As it was, white was not in the color table. (Colors were spaced by 256/16 = 16). Now, colors are spaced by (255-1)/(16-1) = 17, which gives the full range.

Version: 0.62 Date: 5-25-2005

Added endianess check function IsBigEndian() to EasyBMP_DataStructures.h file.

Added functions to swap bytes in WORD and DWORD multibyte variables to EasyBMP_DataStructures.h file for future big-endian support.

Added functions to switch endianess to BMFH and BMIH objects to EasyBMP_DataStructures.h file.

Added endianess checks to ReadFromFile() and WriteToFile() functions in EasyBMP_BMP.h file, along with endianess conversions where necessary.

Added endianess checks and conversions to GetBMFH() and GetBMIH() functions in EasyBMP VariousBMPutilities.h file.

Rewrote GetBitmapInfo() function to use GetBMFH() and GetBMIH() functions instead. (In EasyBMP_VariousBMPutilities.h.) This cuts down on the redundancy in the code.

Renamed GetBitmapInfo() to DisplayBitmapInfo() in the EasyBMP_VariousBMPutilities.h file.

With these changes, big-endian architectures should be supported, including IBM PowerPC, Sun Sparc, Motorola 86k, etc., and including Mac OSX.

Version: 0.63 Date: 7-20-2005

Added IntPow(int,int) function to help compiling with std namespace. Besides, integer operations are faster and more accurate.

Moved Square(double), IntSquare(int), and IntPow(int,int) to EasyBMP_DataStructures.h

Simplified and cleaned up code in Create4bitColorTable(RGBApixel**).

Changed safety check in BMP.ReadFromFile(char*) to set size to 1 x 1 if width or height is non-positive, rather than simply negative.

Added bounds checking to BMP.operator()(int,int) to automatically truncate requested pixel if out of bounds. Also added a warning to cue the user in. :-)

Made error messages more consistent in format.

Simplified and cleaned up code in Create4bitColorTable(RGBApixel**).

Added #include <iostream.h> to EasyBMP.h, since EasyBMP uses cout, etc.

Simplified and cleaned up code in Create1bitColorTable(RGBApixel**).

Changed BMP.SetSize(int,int) to disallow non-positive widths and heights, rather than simply negative widths and heights. Such function calls are now ignored.

Version: 0.64 Date: 8-2-2005 Changed "include <iostream.h>" to "include <iostream>" for ANSI-C++ compliance, as well as for better compatibility with the std namespace and VC++. (Thanks, Tommy Li!)

Added some #ifndef pragmas to each header so that it should be fine to incluce EasyBMP.h in multiple files in larger projects.

Added "using namespace std" inside any function that used C++ math or I/O operations. I avoided putting "using namespace std" anywhere with global scope for maximum compatibility with C++ software in the wild.

Added includes for <cmath> and <cstdio> to EasyBMP.h

Removed unused temporary variables (TempWORD and TempDWORD) from EasyBMP_BMP.h for cleaner compiling. If I see any more such unused variables, I'll remove them, too.

Version: 0.65 Date: 8-13-2005

Moved implementations of BMP::BMP(), BMP::~BMP(), and BMP::operator()(int,int) outside of the class. This should help for eventually moving everything into a separate cpp file.

Made RGBApixel** Pixels a private data member of the class BMP.

Added function void BMP::SetColor(int,RGBApixel) to BMP class to allow safe method of changing a color in the color table.

Added function RGBApixel BMP::GetColor(int) to BMP class to allow safe method of retrieving a color in the color table.

Cleaned up error messages in EasyBMP BMP.h

Cleaned up error messages in EasyBMP VariousBMPutilities.h

Version: 0.66 Date: 8-18-2005

EasyBMP_StandardColorTables.h was removed from the library.

CreateStandardColorTable(RGBApixel**,int) was changed to CreateStandardColorTable() and made a member function of BMP. All other CreateStandardColorTable functions are now unnecessary and have been removed.

CreateGreyscaleColorTable(RGBApixel**,int) was changed to CreateStandardColorTable(BMP&) and moved to EasyBMP_VariousBMPutilities.h. RGBApixel* Colors was made a private data member of the BMP class. CreateGreyscaleColorTable(BMP&) was renamed to CreateGrayscaleColorTable(BMP&). *_____* Version: 0.67 Date: 9-14-2005 Made the EasyBMP custom math functions in EasyBMP DataStructures.h inline. (Square,IntSquare,IntPow). This should make those function calls faster while improving compatibility with compiling DLL's. Separated the code from SafeFread() in EasyBMP BMP.h to improve compatibility with compiling DLL's. Removed #define _WINGDI_H from EasyBMP_DataStructures.h to improve compatibility with win32 applications. Instead, there's an extra #ifndef _SELF_DEFINED_BMP_DATA_TYPES conditional added. _SELF_DEFINED_BMP_DATA_TYPES renamed to _SELF_DEFINED_WINGDI in EasyBMP DataStructures.h. All bit-flipping functions (IsBigEndian, FlipWORD, FlipDWORD) in EasyBMP_DataStructures.h were made inline to improve execution speed and improve compatibility with compiling DLL's.

All code was separated from function declarations in EasyBMP_VariousBMPutilities.h to improve compatibility with compiling DLL's.

Updated and cleaned up layout of EasyBMP ChangeLog.txt.

Updated contact and support information in library files.

Corrected the LGPL license version.

Version: 0.68 Date: 10-9-2005

Changed references to FILE to std::FILE in the SafeFread function in EasyBMP_BMP.h to improve compatibility with Borland's compiler.

Removed a few assignments in EasyBMP_BMP.h that weren't used to improve efficiency and reduce Borland warnings.

Changed calls like NotCorrupted = SafeFread() to NotCorrupted &= SafeFread() in BMP::ReadFromFile() in EasyBMP_BMP.h to improve robustness. Now, if the NotCorrupted bit is ever set to false, it stays false, meaning that the function won't "forget" that it encountered file corruption.

<_____*

Version: 0.69

Date: 10-19-2005

Changed BMP::WriteToFile(char*) to BMP::WriteToFile(const char*) in EasyBMP_BMP.h to respond to a feature request.

Changed BMP::ReadFromFile(char*) to BMP::ReadToFile(const char*) in EasyBMP_BMP.h to respond to a feature request.

Made BMP::ReadFromFile() and BMP::WriteToFile() in EasyBMP_BMP.h return true/false to indicate success/failure in the operations. These functions previously returned void.

Made BMP::SetSize() and BMP::SetBitDepth() in EasyBMP_BMP.h return true/false to indicate success/failure in the operations. These functions previously returned void.

Made BMP::SetColor() and BMP::CreateStandardColorTable() in EasyBMP_BMP.h return true/false to indicate success/failure in the operations. These functions previously returned void.

Made CreateGrayscaleColorTable() in EasyBMP_VariousBMPutilities.h return true/false to indicate success/failure in the operations. This function previously returned void.

Changed the char* argument GetBMFH(char*), GetBMIH(char*), DisplayBitmapInfo(char*), and GetBitmapColorDepth(char*) in EasyBMP_VariousBMPutilities.h to const char* for cleaner, safer programming.

Version: 0.70

Date: 10-19-2005

Found and fixed error in BMP::ReadFromFile() in the check for only reading support bit depths.

Changed license from LGPL to BSD (revised/modified) to simplify licensing issues and resolve any lingering licensing questions.

Fixed compiler error when using MSVC++.

Improved fix to allow compiling with Borland without breaking Borland support.

Added a few lines to EasyBMP.h to make it easier to tailor code to specific compilers. (For future use as needed.)

Added a few lines to EasyBMP_BMP.h (in BMP::ReadFromFile(), BMP::WriteToFile(), and BMP::SetBitDepth()) to eventually add support for 16-bit files.

Version: 0.71

Date: 11-01-2005

Cleaned up comments in BMP::ReadFromFile() in EasyBMP_BMP.h

Added endian-safe read support for 16-bit files that are in the standard 5-5-5 format (not specified in bit fields)

Added endian-safe read support for 16-bit files that use bit fields, including 5-6-5 files.

Added endian-safe write support for 16-bit files. Uses the 5-6-5 encoding scheme to maximize the utility of the bits used.

Added a check for compression in BMP::ReadFromFile(). Because file compression is beyond the scope of EasyBMP, such files are not supported, and EasyBMP now properly detects these situations and exits with an error.

Added a check for files that attempt to use bit fields but are not 16-bit files to BMP::ReadFromFile(). Such files are not supported.

Added a check to BMP::ReadFromFile() for files that use unknown values of bmih.biCompression, such as old OS2 bitmaps. Such files are not supported.

Removed "switching endianness" messages from EasyBMP_BMP.h

Added support for indexed (1, 4, and 8-bit) files that don't specify all the colors.

Added support for reading files that include extra meta data before the pixels. This data is skipped.

Added enclosing #ifndef EasyBMP ... lines to EasyBMP.h as a further safeguard when EasyBMP is included in multiple cpp files.

Version: 1.00

Date: 02-06-2006

First Production/Stable release.

Corrected typographical errors in the comment sections of all files.

Updated copyright on all files.

Removed extraneous comment in BMIH::BMIH() function in EasyBMP_DataStructures.h file.

Replaced instances of \n with the more modern endl in EasyBMP_DataStructures.h, EasyBMP_BMP.h, and EasyBMP_VariousBMPutilities.h.

Added placeholder MetaData1 and MetaData2 data members to the BMP class for potential future use.

Removed extraneous comments from EasyBMP BMP.h.

Removed warning messages for switching endianness from EasyBMP_VariousBMPutilities.h.

Updated copyright in EasyBMP_ChangeLog.txt file.

Fixed formatting issues in EasyBMP_ChangeLog.txt file.

Added DefaultXpelsPerMeter and DefaultYpelsPerMeter in EasyBMP.h. These will default to 96 dpi.

Changed BMP::WriteToFile() to use DefaultXpelsPerMeter and DefaultYpelsPerMeter when writing the BMIH structure.

Added XpelsPerMeter and YpelsPerMeter data members to BMP class so that horizontal and vertical resolution are handled properly. Currently, upon reading a file, the stated resolutions are preserved, and upon writing, if no resolutions are given, the defaults (of 96 DPI) are used.

Added function void BMP::SetDPI(int,int) to set the horizontal and vertical resolutions.

Removed some unnecessary code from GetBitmapColorDepth() in EasyBMP_VariousBMPutilities.h.

Fixed a bug in RangedPixelToPixelCopyTransparent() and RangedPixelToPixelCopy() in EasyBMP_VariousBMPutilities.h which caused copies to be truncated by an extra row or column in certain circumstances.

Fixed a bug in RangedPixelToPixelCopyTransparent() and RangedPixelToPixelCopy() in EasyBMP_VariousBMPutilities.h which checked the wrong variable (FromT instead of FromB) to see if it was out of range.

Added extra checks to RangedPixelToPixelCopyTransparent() and RangedPixelToPixelCopy() in EasyBMP_VariousBMPutilities.h to prevent attempted access of out-of-range pixels.

Version: 1.01

Date: 03-31-2006

Made only the short functions Square, IntSquare, IsBigEndian, FlipWORD, and FlipDWORD inline functions in EasyBMP_DataStructures.h.

Moved all code (other than inline functions) to EasyBMP.cpp.

Changed DefaultXPelsPerMeter and DefaultYPelsPerMeter to #define lines in EasyBMP.h to make the library compatible with with the header-code split.

Removed memory hole in ~BMP() where "delete Colors;" was used instead of "delete [] Colors;". Likewise with MetaData1 and MetaData2.

Fixed memory leak in BMP::SetBitDepth() by changing to delete [] Colors;

Removed potential memory leak in BMP::WriteToFile() in 24- and 32-bit writing where szTemp wasn't delete at the end of a row.

Fixed bug where XPelsPerMeter and YPelsPerMeter weren't properly initialized in the BMP::BMP() constructor, leading to strange horizontal and vertical resolutions.

Fixed memory leak in BMP::ReadFromFile() where TempSkipBYTE wasn't deleted.

Fixed memory leak in BMP::ReadFromFile() where szTemp wasn't deleted.

Added BMP::TellVerticalDPI() and BMP::TellHorizontalDPI() functions to give this information. If those values have not yet been set, then they are first set to the EasyBMP defaults of 96 dpi.

Set uninitialized RGBApixel values to white (255,255,255,0) in a few functions for the BMP class.

Added a sample cpp application and makefile.

Version: 1.02

Date: 05-29-2006

Inserted a line into EasyBMP.h to suppress the Visual Studio warnings. We'll keep using the C++ standard fopen for now until fopen_s becomes a real standard.

Moved the code sample and makefile to a subdirectory, so that unzipping EasyBMP#_##.zip into a project directory doesn't overwrite any crucial makefiles.

Improved SafeFread() to check if the proper amount of data could be read.

Dramatically cleaned up ReadFromFile() code for 1 and 4 bpp files.

Fixed a typo (draw.o) in the sample makefile.

Modified ReadFromFile() to use buffering when reading the pixel data. This should substantially improve disk access performance. Only 16 bpp files are read in the old, slower way.

Changed DWORD from unsigned long to unsigned int. This should fix the issue where 64-bit machines see DWORD as an 8-byte data type, rather than 4 bytes. (Thank you to Bas Wegh!)

Renamed BYTE, WORD, and DWORD data types to ebmpBYTE, ebmpWORD, and ebmpDWORD to eliminate the possibility of conflict with windows applications, particularly with 64-bit windows, which likely uses 8 byte DWORDS.

Modified WriteToFile() to use buffering when reading the pixel data. This should substantially improve disk access performance. Only 16 bpp files are read in the old, slower way.

Added new function, EasyBMPcheckDataSize(), to check that the ebmpBYTE, ebmpWORD, and ebmpDWORD types have the correct type.

Added some new macros of the EasyBMP version number for easier version checking. New versions include _EasyBMP_Version_ (a double), _EasyBMP_Version_String_ (a char* version), and _EasyBMP_Version_Integer_ (an integer version, e.g., 102).

Version: 1.03

Date: 06-20-2006

Inserted a line into EasyBMP.h to suppress the Visual Studio

Added a check to BMP.SetColor() to ensure that the color table is defined before attempting to set a color entry.

Added a check to BMP.GetColor() to ensure that the color table

is defined before attempting to retrieve a color entry.

Simplified the conditional in BMP.WriteToFile() from if(BitDepth == $1 \parallel$ BitDepth == $4 \parallel$...) to the simpler if(BitDepth != 16).

Removed the old, unused code for writing 1- and 4-bit files from BMP.WriteToFile().

Removed the line Colors = new RGBApixel [NumberOfColors]; in BMP.ReadFromFile(). This operation is already covered by the earlier SetBitDepth() call, and may contribute to a memory leak. Furthermore, for files that had fewer than expected number of colors (e.g., an 8-bit file with 236 colors), it lead to memory access errors in BMP.GetColor() and BMP.SetColor(). (In fact, this is the main motivation for release 1.03.)

Added a warning when BMP.ReadFromFile() encounters an underspecified color table, and code to pad the table with white entries.

Added screen output on EasyBMP version and project website to the code sample.

Version: 1.04

Date: 07-22-2006

Removed the assignment to the integer i in IntPow() to eliminate a Borland compiler warning.

Removed the assignment to the integer i in the Read##bitRow() functions to eliminate Borland compiler warnings.

Removed the assignment to ZeroWORD in line 478 of EasyBMP.cpp in BMP::WriteToFile() to eliminate Borland compiler warnings.

Removed the assignment to ZeroWORD in line 825 of EasyBMP.cpp in BMP::ReadFromFile() to eliminate Borland compiler warnings.

The Borland warnings about conditions always being false are incorrect. (Lines 1587, 1594, and 1601.) Likewise, the Borland warnings about unreachable code (lines 1589, 1596, and 1603) are incorrect. This code serves as a protection on unexpected hardware where the data types may not be of the correct size, and helps to future-proof EasyBMP. The first time this type of error was encountered was on 64-bit CPUs, where the size of the DWORD was larger than assumed when writing EasyBMP. Therefore, we will not "correct" these "errors" detected by Borland. If they bother you, compile with the -w-8008 and -w-8066 options.

Borland issues warnings about argc and argv being unused in the sample project. These are silly warnings and will be ignored. If

this warning bothers you, compile with the -w-8057 option.

Modified the sample makefile so that EasyBMP.o depends upon EasyBMP.cpp and EasyBMP*.h in the current working directory, rather than the parent directory.

Added a global EasyBMPwarnings boolean variable, and functions SetEasyBMPwarningsOn() and SetEasyBMPwarningsOff() to enable and disable EasyBMP warnings and errors. Note that this will not disable error checking or any other EasyBMP behavior, other than cout output of the warning and error messages.

Added the function GetEasyBMPwarningState() to query the EasyBMP warning state. (Either warnings are enabled or disabled.)

Removed old commented code (Write1bitRow()) from EasyBMP.cpp.

Replaced the 24-bit EasyBMPbackground.bmp image in the code sample with a dithered 8-bit version to reduce the download size of the core library.

Version: 1.05

Date: 11-01-2006

Renamed BytesRead to ItemsRead in the SafeFread() function in EasyBMP.cpp for greater clarity.

Added a copy constructor to the BMP class. However, note that passing by value is not recommended practice. (Passing by reference is much faster, and consumes less memory.)

Added a new function:

bool Rescale(BMP& InputImage, char mode, int NewDimension); to resize an image. The mode variables are as follows:

'P': resizes the image to a new percentage of the old size,

e.g., 42%, 13%, etc.

example: Rescale(SomeImage, 'p', 42);

'W': resizes the image such that the new width is as specified.

example: Rescale(SomeImage, 'W', 100);

'H': resizes the image such that the new height is as specified.

example: Rescale(SomeImage, 'H', 100);

'F': resizes the image to fit in a square of specified size.

example: Rescale(SomeImage, 'F', 100); // fits in 100x100

// box

All rescaling is done with bilinear interpolation.

Version: 1.06

Date: 12-01-2006

Added includes for <cctype> and <cstring> to EasyBMP.h. These are

used and should have been included all along. This should help with Intel icc compiling.

Fixed the && bug in the copy constructor. (Thank you to user fcnature!)

Added image scaling to the supplied code sample.

Added GetPixle() and SetPixel() functions for future use. These will be added to enable more careful use of the const keyword.



Easy&MP makes life pretty! This cheesy message is brought to you by the EasyBMP project. http://easybmp.sourceforge.net

phplib FAQ

 ${\rm Ted}\ {\rm Rolle}^1$

March 23, 1999

 $^{^{1}{}m ted.rolle@usa.net}$

Contents

Preface	i
1 FAQ	1

Preface

This is the phplib Frequently-Asked Questions document. It is maintained by Ted Rolle <code>;ted.rolle@usa.net;</code>

Chapter 1

FAQ

THE MOST-FREQUENTLY-ASKED QUESTION

setcookie() oops The answer is that something - text or html - has been
written to the page before the call to page_open().

Even newlines or spaces written after local.inc or prepend.php3 - but before page_open() - can do it.

CompTorrent: File Index

Main Page	Classes	Files
File List		

File List

Here is a list of all documented files with brief descriptions:

abstractpeer.h [code]		
comptorrentparser.h [code]		
comptorrentpeer.h [code]		
controller.h [code]		
known_peer.h [code]		
listener.h [code]		
messagecollection.h [code]		
network_socket.h [code]		
orig_data_chunk.h [code]		
processor.h [code]		
router.h [code]		
SP2PCommon.h [code]		
trackerparser.h [code]		
worker_thread.h [code]		

Generated on Sat Mar 27 12:27:46 2010 for CompTorrent by 1.6.1



















....









Here is a list of all documented class members with links to the class documentation for each member:

• xml_to_map() : Controller

Generated on Sat Mar 27 12:27:34 2010 for CompTorrent by



xml_to_map Controller



• xml_to_map() : Controller



Main Page Classes Files

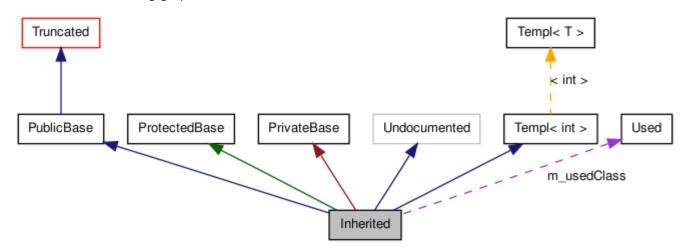
Graph Legend

This page explains how to interpret the graphs that are generated by doxygen.

Consider the following example:

```
/*! Invisible class because of truncation */
class Invisible { };
/*! Truncated class, inheritance relation is hidden */
class Truncated : public Invisible { };
/* Class not documented with doxygen comments */
class Undocumented { };
/*! Class that is inherited using public inheritance */
class PublicBase : public Truncated { };
/*! A template class */
template<class T> class Templ { };
/*! Class that is inherited using protected inheritance */
class ProtectedBase { };
/*! Class that is inherited using private inheritance */
class PrivateBase { };
/*! Class that is used by the Inherited class */
class Used { };
/*! Super class that inherits a number of other classes */
class Inherited : public PublicBase,
                   protected ProtectedBase,
                   private PrivateBase,
                   public Undocumented,
                   public Templ<int>
  private:
    Used *m_usedClass;
```

This will result in the following graph:



The boxes in the above graph have the following meaning:

- A filled gray box represents the struct or class for which the graph is generated.
- A box with a black border denotes a documented struct or class.
- A box with a grey border denotes an undocumented struct or class.
- A box with a red border denotes a documented struct or class forwhich not all inheritance/containment relations are shown. A graph is truncated if it does not fit within the specified boundaries.

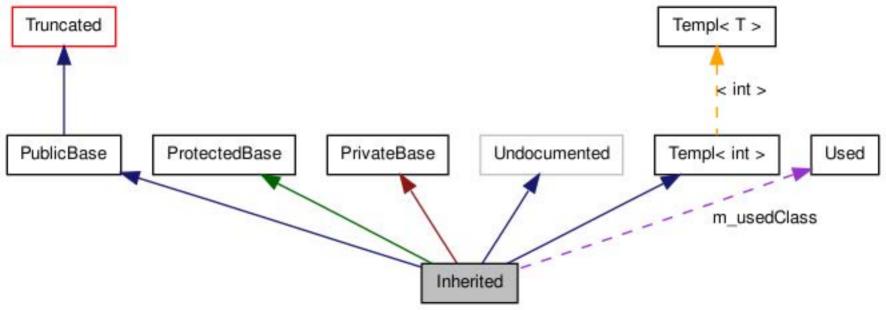
The arrows have the following meaning:

A dark blue arrow is used to visualize a public inheritance relation between two classes.

- A dark green arrow is used for protected inheritance.
- A dark red arrow is used for private inheritance.
- A purple dashed arrow is used if a class is contained or used by another class. The arrow is labeled with the variable(s) through which the pointed class or struct is accessible.
- A yellow dashed arrow denotes a relation between a template instance and the template class it was instantiated from. The arrow is labeled with the template parameters of the instance.

Generated on Sat Mar 27 12:27:46 2010 for CompTorrent by





.







known_peer.h

```
00001 #ifndef _known_peer_h_
00002 #define _known_peer_h_
00003
00004 class peer {
00005
00006 public:
00007
00008
               peer() {
00009
                        ip = port = string("");
                        distorrent = NULL;
00010
00011
00012
               peer(string ip_, string port_, void* distorrent_) {
00013
00014
                        ip = ip_;
00015
                        port = port_;
00016
                        distorrent = distorrent_;
00017
00018
00019
               peer& operator=(const peer &rhs)
                                         ip = rhs.ip;
00020
00021
                                         port = rhs.port;
00022
                                         distorrent = rhs.distorrent;
                                         return *this;
00023
00024
00025
00026
                                string ip, port;
               void* distorrent;
00027
00028 };
00029
00030 typedef list<peer> peer_type;
00031 typedef peer_type::iterator peer_type_iter;
00032
00033 class file_chunk {
00034
00035 public:
00036
               file_chunk(string name_, string type_) {
00037
00038
                        name = name_;
00039
                        type = type_;
00040
00041
               file_chunk() {
00042
                        name = "";
00043
                        type = "";
00044
00045
00046
00047
               string name;
00048
               string type;
00049
00050
               file_chunk& operator=(const file_chunk &rhs) {
00051
                                               name = rhs.name;
00052
                                               type = rhs.type;
return *this;
00053
00054
                              };
00055
00056
00057 typedef list<file_chunk> file_chunk_type;
00058 typedef file_chunk_type::iterator file_chunk_iter;
00059
00060
00061
00062
00063 class file_chunk_request {
00064
00065 public:
00066
00067
               file_chunk_request(string name_, string type_, void* who_) {
00068
                        name = name_;
00069
                        type = type_;
00070
                        who = who_i
00071
               }
00072
```

```
file chunk request() {
00074
                       name = "";
                       type = "";
00075
                       who = NULL;
00076
00077
00078
00079
               string name;
08000
               string type;
               void* who;
00081
00082
00083
               file_chunk_request& operator=(const file_chunk_request &rhs) {
00084
                                                       name = rhs.name;
00085
                                                       type = rhs.type;
00086
                                                       who = rhs.who;
00087
                                                       return *this;
                                              }
00088
                                     };
00089
00090
00091
00092 typedef list<file_chunk_request> file_request_type;
00093 typedef file_request_type::iterator file_request_iter;
00094
00095
00096
00097 using namespace std;
00098 using namespace ost;
00099
00100 enum connection_type { unconnected, incoming, outgoing };
00101
00102 struct known_peer {
00103
00104
               std::string trim_spaces( std::string const& str) {
00105
                       string res = str;
00106
00107
                       if (res.size() > 1)
                                std::string::size_type const first = str.find_first_not_of("
00108
");
00109
                                res = str.substr(first, str.find last not of(" ") - first +
1);
00110
                       }
00111
00112
                       return res;
00113
00114
00115
               known_peer() {
                       ct = unconnected;
00116
00117
                       tried = false;
00118
                       distorrent = NULL;
00119
00120
00121
              known_peer(string host_ip, string host_port, connection_type ct_ =
unconnected, string routeid_ = "0")
00122
00123
                       routeid = trim_spaces( routeid_ );
00124
00125
                       ip = host_ip;
00126
                       port = host_port;
                       ct = ct_;
00127
00128
                       tried = false;
00129
                       distorrent = NULL;
00130
00131
00132
               known_peer(string host_ip, int host_port, connection_type ct_) {
00133
00134
                        // convert the int to a string and send it to the other contructor
00135
                       tried = false;
00136
                       ostringstream host port ;
                       host_port_ << host_port;
ip = host_ip;</pre>
00137
00138
                       port = host_port_.str();
00139
00140
                       ct = ct_i
00141
00142
00143
               known_peer(string name_, string uuid_, string ip_, string port_, string
update_time_,
00144
                           unsigned long num_computed_chunks_, unsigned long
num_original_chunks_, unsigned long num_connections_) {
00145
00146
                       name = name
00147
                       uuid = uuid_;
00148
                       ip = ip ;
```

```
00149
                       port = port_;
                       update time = update time ;
00150
                       num_computed_chunks = num_computed_chunks_;
00151
                       num_original_chunks = num_original_chunks_;
00152
                       num_connections = num_connections_;
00153
00154
                       distorrent = NULL;
00155
00156
00157
               void push data chunk(bool have chunk) {
00158
                       data_chunks.push_back(have_chunk);
00159
00160
00161
               void push orig chunk(bool have chunk)
00162
                       orig_chunks.push_back(have_chunk);
00163
00164
00165
               string comp_chunk_string() {
00166
00167
                       ostringstream str;
                       for (vector<bool>::iterator i = data_chunks.begin(); i !=
00168
data chunks.end(); i++) {
00169
                                str << (int)*i;
00170
00171
                       return str.str();
00172
00173
00174
               string orig_chunk_string() {
00175
00176
                       ostringstream str;
00177
                       for (vector<bool>::iterator i = orig_chunks.begin(); i !=
orig chunks.end(); i++)
00178
                                str << (int)*i;
00179
00180
                       return str.str();
00181
00182
00183
               known_peer& operator=(known_peer &rhs)
00184
00185
                                              routeid = rhs.routeid;
00186
                                              name = rhs.name;
00187
                                              uuid = rhs.uuid;
00188
                                              ip = rhs.ip;
                                              port = rhs.port;
00189
00190
                                              update_time = rhs.update_time;
00191
                                              num_computed_chunks = rhs.num_computed_chunks;
                                              num original chunks = rhs.num original chunks;
00192
00193
                                              num_connections = rhs.num_connections;
00194
                                              tried = rhs.tried;
00195
                                              distorrent = rhs.distorrent;
00196
00197
                                              // data chunks
00198
                                              data_chunks.clear();
00199
                                              for (vector<bool>::iterator i =
rhs.data_chunks.begin(); i != rhs.data_chunks.end(); i++)
00200
                                                       data_chunks.push_back(*i);
00201
00202
00203
                                              // orig chunks
00204
                                              orig_chunks.clear();
                                              for (vector<bool>::iterator i =
00205
rhs.orig_chunks.begin(); i != rhs.orig_chunks.end(); i++)
00206
                                                       orig_chunks.push_back(*i);
00207
00208
                                              return *this;
00209
00210
00211
                                     bool operator==(known_peer rhs) {
    return ((ip == rhs.ip)); // && (port ==
00212
00213
rhs.port)/* && (uuid == rhs.uuid)*/);
00214
00215
00216
                                     bool operator!=(known_peer rhs)
                                              return ((ip != rhs.ip)); // || (port !=
00217
rhs.port)/* || (uuid != rhs.uuid)*/);
00218
00219
                                     bool operator>(known_peer rhs) {
00220
00221
                                              return (num connections > rhs.num connections);
00222
00223
```

```
bool operator<(known_peer rhs) {</pre>
00224
00225
                                                         return (num_connections < rhs.num_connections);</pre>
00226
00227
                                              void set_tried(bool tried_) { tried = tried_; }
bool have_tried() { return tried; }
00228
00229
00230
00231
                                              vector<bool> data_chunks;
00232
                  vector<bool> orig_chunks;
00233
                  connection_type ct;
00234
00235
                  bool tried;
00236
                  string name, uuid, ip, port, update_time; unsigned long num_computed_chunks; unsigned long num_original_chunks;
00237
00238
00239
                  unsigned long num_connections;
00240
00241
                  string routeid;
00242
00243
                  void* distorrent;
00244 };
00245
00246 #endif
```

Generated on Sat Mar 27 12:27:34 2010 for CompTorrent by







listener.h

```
00001
00002 #ifndef _Listener_h_
00003 #define _Listener_h_
00004
00005 #include <iostream>
00006 #include <list>
00007 #include <cc++/xml.h>
00008 #include "network_socket.h" 00009 #include "comptorrentparser.h"
00010 #include "processor.h"
00011
00012 using namespace std; 00013 using namespace ost;
00014
00015 class Listener : public Thread {
00016
00017 public:
00018
                Listener(Controller* parent_, string host_ip_, string host_port_) {
00019
00020
                          host_ip = host_ip_;
00021
                          host_port = host_port_;
00022
                          parent = parent_;
00023
00024
00025
                void run();
00026
00027 protected:
00028
00029
                string host_ip;
00030
                string host_port;
                Controller * parent;
00031
00032 };
00033
00034 #endif
```



Main Page Classes Files

CompTorrent Documentation



-rw-r--r-- 1 aburto 652 Dec 7 1995 Makefile -rw-r--r-- 1 aburto 80 Dec 1 1995 dhry.bat -rw-r--r-- 1 aburto 4090 Oct 9 07:59 dhry.doc -rw-r--r-- 1 aburto 16173 Aug 3 1995 dhry.h -rw-r--r-- 1 aburto 10446 Aug 3 1995 dhry11.c -rw-r--r-- 1 aburto 11527 Aug 3 1995 dhry21a.c -rw-r--r-- 1 aburto 5140 Aug 3 1995 dhry21b.c -rw-r--r-- 1 aburto 366 Oct 9 08:00 dhryrun.doc -rw-r--r-- 1 aburto 9 Aug 3 1995 input -rw-r--r-- 1 aburto 10312 Mar 3 09:58 timers_b.c



messagecollection.h

```
00001 #ifndef _MessageCollection_h_
00002 #define _MessageCollection_h_
00003
00004 #include <iostream>
00005 #include <list>
00006 #include <cc++/xml.h>
00007
00008 using namespace std;
00009 using namespace ost;
00010
00011 typedef list<string> collection_type;
00012 typedef collection_type::iterator collection_iter;
00013
00014 class MessageCollection {
00015
00016 public:
00017
                  MessageCollection() { num_items = 0; }
void push_back(const string);
00018
00019
                  void push_front(const string);
00020
00021
                  bool pop(string&);
                  collection_iter begin() { return collection.begin(); }
collection_iter end() { return collection.end(); }
00022
00023
00024
                  void clear();
00025
                  unsigned long size() { return collection.size(); }
00026
00027 private:
00028
00029
                  collection_type collection;
00030
                  Mutex collection_mutex;
00031
00032
                  int num_items;
00033 };
00034
00035 #endif
```

Generated on Sat Mar 27 12:27:34 2010 for CompTorrent by





network_socket.h

```
00001
00002 #ifndef _NetworkSocket_h_
00003 #define _NetworkSocket_h_
00004
00005 #include <cc++/socket.h>
00006
00007 using namespace std;
00008 using namespace ost;
00009
00010 class NetworkSocket : public TCPSocket
00011 {
00012 public:
00013
                 NetworkSocket(tpport_t port);
NetworkSocket(const IPV4Address &ia, tpport_t port);
00014
00015
00016
                 ~NetworkSocket();
00017
00018 };
00019
00020 #endif
```



No Matches

Main Page Classes **Files** File List

orig_data_chunk.h

```
00001 #ifndef _orig_data_chunk_h_
00002 #define _orig_data_chunk_h_
00003
00004 using namespace std;
00005 using namespace ost;
00006
00007 const std::string UNDEFINED = "UNDEFINED";
80000
00009 enum computation_state { unknown, waiting, computed };
00010
00011 struct orig_data_chunk {
00012
00013
               orig_data_chunk(const orig_data_chunk &rhs) {
00014
00015
                        md5 = rhs.md5;
00016
                        name = rhs.name;
00017
                        size = rhs.size;
00018
                        num_requests = rhs.num_requests;
00019
                        num_sent = rhs.num_sent;
                        comp_state = rhs.comp_state;
00020
                        orig_file_exists = rhs.orig_file_exists;
00021
00022
                        comp_file_exists = rhs.comp_file_exists;
00023
                        requested = rhs.requested;
00024
00025
00026
               orig data chunk()
00027
                        md5 = UNDEFINED;
00028
                        name = UNDEFINED;
00029
                        size = UNDEFINED;
00030
                        comp_state = unknown;
00031
                        orig_file_exists = false;
                        comp_file_exists = false;
requested = false;
00032
00033
00034
00035
00036
               orig_data_chunk& operator=(const orig_data_chunk &rhs)
00037
00038
                                                     md5 = rhs.md5;
00039
                                                     name = rhs.name;
                                                     size = rhs.size;
00040
00041
                                                     num_requests = rhs.num_requests;
00042
                                                     num_sent = rhs.num_sent;
00043
                                                     comp_state = rhs.comp_state;
00044
                                                     orig_file_exists = rhs.orig_file_exists;
00045
                                                     comp_file_exists = rhs.comp_file_exists;
00046
                                                     requested = rhs.requested;
00047
                                                     return *this;
00048
00049
                                            string md5, name, size;
00050
00051
               unsigned long num_requests;
00052
               unsigned long num_sent;
00053
               computation_state comp_state;
00054
00055
               bool orig_file_exists;
00056
               bool comp_file_exists;
               bool requested;
00057
00058
00059
00060
00061 #endif
```





processor.h

```
00001 #include "worker_thread.h" 00002 #include "controller.h"
00003
00004 #ifndef _Processor_h_
00005 #define _Processor_h_
00006
00007 class Controller; // forward declaration
80000
00009 using namespace std;
00010 using namespace ost;
00011
00012 enum processor state { idle, waiting on origchunk, processing };
00013
00014 class Processor : public Thread
00015 {
00016 public:
00017
               Processor(Controller* s);
00018
00019
                void run();
               bool processing_finished() { return finished; }
00020
00021
00022 protected:
00023
00024
               void process_orig_chunk(string chunk_name);
00025
               processor_state state;
                Controller* parent;
00026
00027
               string current_chunk_name;
00028
00029
                int wait_timeout;
00030
               bool finished;
00031 };
00032
00033 #endif
```

Generated on Sat Mar 27 12:27:34 2010 for CompTorrent by 1.6.1



THE Q PUBLIC LICENSE version 1.0

Copyright (C) 1999 Trolltech AS, Norway. Everyone is permitted to copy and distribute this license document.

The intent of this license is to establish freedom to share and change the software regulated by this license under the open source model.

This license applies to any software containing a notice placed by the copyright holder saying that it may be distributed under the terms of the Q Public License version 1.0. Such software is herein referred to as the Software. This license covers modification and distribution of the Software, use of third-party application programs based on the Software, and development of free software which uses the Software.

Granted Rights

- 1. You are granted the non-exclusive rights set forth in this license provided you agree to and comply with any and all conditions in this license. Whole or partial distribution of the Software, or software items that link with the Software, in any form signifies acceptance of this license.
- 2. You may copy and distribute the Software in unmodified form provided that the entire package, including but not restricted to copyright, trademark notices and disclaimers, as released by the initial developer of the Software, is distributed.
- 3. You may make modifications to the Software and distribute your modifications, in a form that is separate from the Software, such as patches. The following restrictions apply to modifications:
 - a. Modifications must not alter or remove any copyright notices in the Software.
 - b. When modifications to the Software are released under this license, a non-exclusive royalty-free right is granted to the initial developer of the Software to distribute your modification in future versions of the Software provided such versions remain available under these terms in addition to any other license(s) of the initial developer.
- 4. You may distribute machine-executable forms of the Software or machine-executable forms of modified versions of the Software, provided that you meet these restrictions:
 - a. You must include this license document in the distribution.
 - b. You must ensure that all recipients of the machine-executable forms

are also able to receive the complete machine-readable source code to the distributed Software, including all modifications, without any charge beyond the costs of data transfer, and place prominent notices in the distribution explaining this.

- c. You must ensure that all modifications included in the machine-executable forms are available under the terms of this license.
- 5. You may use the original or modified versions of the Software to compile, link and run application programs legally developed by you or by others.
- 6. You may develop application programs, reusable components and other software items that link with the original or modified versions of the Software. These items, when distributed, are subject to the following requirements:
 - a. You must ensure that all recipients of machine-executable forms of these items are also able to receive and use the complete machine-readable source code to the items without any charge beyond the costs of data transfer.
 - b. You must explicitly license all recipients of your items to use and re-distribute original and modified versions of the items in both machine-executable and source code forms. The recipients must be able to do so without any charges whatsoever, and they must be able to re-distribute to anyone they choose.
 - c. If the items are not available to the general public, and the initial developer of the Software requests a copy of the items, then you must supply one.

Limitations of Liability

In no event shall the initial developers or copyright holders be liable for any damages whatsoever, including - but not restricted to - lost revenue or profits or other direct, indirect, special, incidental or consequential damages, even if they have been advised of the possibility of such damages, except to the extent invariable law, if any, provides otherwise.

No Warranty

The Software and this license document are provided AS IS with NO WARRANTY OF ANY KIND, INCLUDING THE WARRANTY OF DESIGN,

MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Choice of Law This license is governed by the Laws of Norway. Disputes shall be

settled by Oslo City Court.

Compilation instructions for an Ubuntu system: Install libcommoncpp2-dev using synaptic

then make

See comptorrent.org for more up to date information.

/** @mainpage

<h1> TinyXml </h1>

TinyXml is a simple, small, C++ XML parser that can be easily integrating into other programs.

<h2> What it does. </h2>

In brief, TinyXml parses an XML document, and builds from that a Document Object Model (DOM) that can be read, modified, and saved.

XML stands for "eXtensible Markup Language." It allows you to create your own document markups. Where HTML does a very good job of marking documents for browsers, XML allows you to define any kind of document markup, for example a document that describes a "to do" list for an organizer application. XML is a very structured and convenient format. All those random file formats created to store application data can all be replaced with XML. One parser for everything.

The best place for the complete, correct, and quite frankly hard to read spec is at http://www.w3.org/TR/2004/REC-xml-20040204/. An intro to XML (that I really like) can be found at http://skew.org/xml/tutorial.

There are different ways to access and interact with XML data. TinyXml uses a Document Object Model (DOM), meaning the XML data is parsed into a C++ objects that can be browsed and manipulated, and then written to disk or another output stream. You can also construct an XML document from scratch with C++ objects and write this to disk or another output stream.

TinyXml is designed to be easy and fast to learn. It is two headers and four cpp files. Simply add these to your project and off you go. There is an example file - xmltest.cpp - to get you started.

TinyXml is released under the ZLib license, so you can use it in open source or commercial code. The details of the license are at the top of every source file.

TinyXml attempts to be a flexible parser, but with truly correct and compliant XML output. TinyXml should compile on any reasonably C++ compliant system. It does not rely on exceptions or RTTI. It can be compiled with or without STL support. TinyXml fully supports the UTF-8 encoding, and the first 64k character entities.

<h2> What it doesn't do. </h2>

It doesnt parse or use DTDs (Document Type Definitions) or XSLs (eXtensible Stylesheet Language.) There are other parsers out there (check out www.sourceforge.org, search for XML) that are much more fully

featured. But they are also much bigger, take longer to set up in your project, have a higher learning curve, and often have a more restrictive license. If you are working with browsers or have more complete XML needs, TinyXml is not the parser for you.

The following DTD syntax will not parse at this time in TinyXml:

```
@verbatim
     <!DOCTYPE Archiv [
          <!ELEMENT Comment (#PCDATA)>
          ]>
@endverbatim
```

because TinyXml sees this as a !DOCTYPE node with an illegally embedded !ELEMENT node. This may be addressed in the future.

```
<h2> Tutorials. </h2>
```

For the impatient, here is a tutorial to get you going. A great way to get started, but it is worth your time to read this (very short) manual completely.

- @subpage tutorial0

```
<h2> Code Status. </h2>
```

TinyXml is mature, tested code. It is very stable. If you find bugs, please file a bug report on the sourceforge web site (www.sourceforge.net/projects/tinyxml). We'll get them straightened out as soon as possible.

There are some areas of improvement; please check sourceforge if you are interested in working on TinyXml.

```
<h2> Features </h2> <h3> Using STL </h3>
```

TinyXml can be compiled to use or not use STL. When using STL, TinyXml uses the std::string class, and fully supports std::istream, std::ostream, operator<<, and operator>>. Many API methods have both 'const char*' and 'const std::string&' forms.

When STL support is compiled out, no STL files are included whatsover. All the string classes are implemented by TinyXml itself. API methods all use the 'const char*' form for input.

Use the compile time #define:

```
TIXML_USE_STL
```

to compile one version or the other. This can be passed by the compiler, or set as the first line of "tinyxml.h".

Note: If compiling the test code in Linux, setting the environment variable TINYXML_USE_STL=YES/NO will control STL compilation. In the Windows project file, STL and non STL targets are provided. In your project, its probably easiest to add the line "#define TIXML_USE_STL" as the first line of tinyxml.h.

<h3> UTF-8 </h3>

TinyXml supports UTF-8 allowing to manipulate XML files in any language. TinyXml also supports "legacy mode" - the encoding used before UTF-8 support and probably best described as "extended ascii".

Normally, TinyXml will try to detect the correct encoding and use it. However, by setting the value of TIXML_DEFAULT_ENCODING in the header file, TinyXml can be forced to always use one encoding.

TinyXml will assume Legacy Mode until one of the following occurs:

- If the non-standard but common "UTF-8 lead bytes" (0xef 0xbb 0xbf) begin the file or data stream, TinyXml will read it as UTF-8.
- If the declaration tag is read, and it has an encoding="UTF-8", then
 TinyXml will read it as UTF-8.
- If the declaration tag is read, and it has no encoding specified, then TinyXml will read it as UTF-8.
- If the declaration tag is read, and it has an encoding="something else", then TinyXml will read it as Legacy Mode. In legacy mode, TinyXml will work as it did before. It's not clear what that mode does exactly, but old content should keep working.
- Until one of the above criteria is met, TinyXml runs in Legacy Mode.

What happens if the encoding is incorrectly set or detected? TinyXml will try to read and pass through text seen as improperly encoded. You may get some strange results or mangled characters. You may want to force TinyXml to the correct mode.

<by You may force TinyXml to Legacy Mode by using LoadFile(TIXML_ENCODING_LEGACY) or LoadFile(filename, TIXML_ENCODING_LEGACY). You may force it to use legacy mode all the time by setting TIXML_DEFAULT_ENCODING = TIXML_ENCODING_LEGACY. Likewise, you may force it to TIXML_ENCODING_UTF8 with the same technique.

For English users, using English XML, UTF-8 is the same as low-ASCII. You don't need to be aware of UTF-8 or change your code in any way. You can think of UTF-8 as a "superset" of ASCII.

UTF-8 is not a double byte format - but it is a standard encoding of Unicode! TinyXml does not use or directly support wchar, TCHAR, or Microsofts _UNICODE at this time. It is common to see the term "Unicode" improperly refer to UTF-16, a wide byte encoding of unicode. This is a source of confusion.

For "high-ascii" languages - everything not English, pretty much - TinyXml can handle all languages, at the same time, as long as the XML is encoded in UTF-8. That can be a little tricky, older programs and operating systems tend to use the "default" or "traditional" code page. Many apps (and almost all modern ones) can output UTF-8, but older or stubborn (or just broken) ones

still output text in the default code page.

For example, Japanese systems traditionally use SHIFT-JIS encoding. Text encoded as SHIFT-JIS can not be read by tinyxml. A good text editor can import SHIFT-JIS and then save as UTF-8.

The Skew.org link does a great job covering the encoding issue.

The test file "utf8test.xml" is an XML containing English, Spanish, Russian, and Simplified Chinese. (Hopefully they are translated correctly). The file "utf8test.gif" is a screen capture of the XML file, rendered in IE. Note that if you don't have the correct fonts (Simplified Chinese or Russian) on your system, you won't see output that matches the GIF file even if you can parse it correctly. Also note that (at least on my Windows machine) console output is in a Western code page, so that Print() or printf() cannot correctly display the file. This is not a bug in TinyXml - just an OS issue. No data is lost or destroyed by TinyXml. The console just doesn't render UTF-8.

```
<h3> Entities </h3>
```

TinyXml recognizes the pre-defined "character entities", meaning special characters. Namely:

@verbatim

& & < < > > " " ' ' @endverbatim

These are recognized when the XML document is read, and translated to there UTF-8 equivalents. For instance, text with the XML of:

```
@verbatim
Far & Camp; Away
@endverbatim
```

will have the Value() of "Far & Away" when queried from the TiXmlText object, and will be written back to the XML stream/file as an ampersand. Older versions of TinyXml "preserved" character entities, but the newer versions will translate them into characters.

Additionally, any character can be specified by its Unicode code point: The syntax " " or " " are both to the non-breaking space characher.

```
<h3> Streams </h3> With TIXML_USE_STL on, TiXml has been modified to support both C (FILE) and C++ (operator <<,>>) streams. There are some differences that you may need to be aware of.
```

C style output:

- based on FILE*
- the Print() and SaveFile() methods

Generates formatted output, with plenty of white space, intended to be as human-readable as possible. They are very fast, and tolerant of ill formed XML documents. For example, an XML document that contains 2 root elements and 2 declarations, will still print.

C style input:

- based on FILE*
- the Parse() and LoadFile() methods

A fast, tolerant read. Use whenever you don't need the C++ streams.

C++ style ouput:

- based on std::ostream
- operator<<

Generates condensed output, intended for network transmission rather than readability. Depending on your system's implementation of the ostream class, these may be somewhat slower. (Or may not.) Not tolerant of ill formed XML: a document should contain the correct one root element. Additional root level elements will not be streamed out.

C++ style input:

- based on std::istream
- operator>>

Reads XML from a stream, making it useful for network transmission. The tricky part is knowing when the XML document is complete, since there will almost certainly be other data in the stream. TinyXml will assume the XML data is complete after it reads the root element. Put another way, documents that are ill-constructed with more than one root element will not read correctly. Also note that operator>> is somewhat slower than Parse, due to both implementation of the STL and limitations of TinyXml.

<h3> White space </h3>

The world simply does not agree on whether white space should be kept, or condensed. For example, pretend the '_' is a space, and look at "Hello___world". HTML, and at least some XML parsers, will interpret this as "Hello_world". They condense white space. Some XML parsers do not, and will leave it as "Hello___world". (Remember to keep pretending the _ is a space.) Others suggest that __Hello__world__ should become Hello__world.

It's an issue that hasn't been resolved to my satisfaction. TinyXml supports the first 2 approaches. Call TiXmlBase::SetCondenseWhiteSpace(bool) to set the desired behavior. The default is to condense white space.

If you change the default, you should call TiXmlBase::SetCondenseWhiteSpace(bool) before making any calls to Parse XML data, and I don't recommend changing it after it has been set.

<h3> Handles </h3>

Where browsing an XML document in a robust way, it is important to check for null returns from method calls. An error safe implementation can generate a lot of code like:

```
@verbatim
TiXmlElement* root = document.FirstChildElement( "Document");
if (root)
{
    TiXmlElement* element = root->FirstChildElement( "Element" );
    if ( element )
         TiXmlElement* child = element->FirstChildElement( "Child" );
         if (child)
              TiXmlElement* child2 = child->NextSiblingElement("Child");
              if (child2)
                  // Finally do something useful.
@endverbatim
Handles have been introduced to clean this up. Using the TiXmlHandle class,
the previous code reduces to:
@verbatim
TiXmlHandle docHandle( &document );
TiXmlElement* child2 = docHandle.FirstChild( "Document" ).FirstChild( "Element" ).Child( "Child", 1 ).Element();
if (child2)
    // do something useful
@endverbatim
```

Which is much easier to deal with. See TiXmlHandle for more information.

<h3> Row and Column tracking </h3>

Being able to track nodes and attributes back to their origin location in source files can be very important for some applications. Additionally, knowing where parsing errors occured in the original source can be very time saving.

TinyXml can tracks the row and column origin of all nodes and attributes in a text file. The TiXmlBase::Row() and TiXmlBase::Column() methods return the origin of the node in the source text. The correct tabs can be configured in TiXmlDocument::SetTabSize().

<h2> Using and Installing </h2>

To Compile and Run xmltest:

A Linux Makefile and a Windows Visual C++ .dsw file is provided. Simply compile and run. It will write the file demotest.xml to your disk and generate output on the screen. It also tests walking the

DOM by printing out the number of nodes found using different techniques.

The Linux makefile is very generic and will probably run on other systems, but is only tested on Linux. You no longer need to run 'make depend'. The dependecies have been hard coded.

```
<h3>Windows project file for VC6</h3>
tinyxml:
                  tinyxml library, non-STL 
tinyxmlSTL:
                     tinyxml library, STL 
tinyXmlTest:
                    test app, non-STL 
tinyXmlTestSTL: test app, STL 
<h3>Linux Make file</h3>
At the top of the makefile you can set:
PROFILE, DEBUG, and TINYXML_USE_STL. Details (such that they are) are in
the makefile.
In the tinyxml directory, type "make clean" then "make". The executable
file 'xmltest' will be created.
<h3>To Use in an Application:</h3>
Add tinyxml.cpp, tinyxml.h, tinyxmlerror.cpp, tinyxmlparser.cpp, tinystr.cpp, and tinystr.h to your
project or make file. That's it! It should compile on any reasonably
compliant C++ system. You do not need to enable exceptions or
RTTI for TinyXml.
<h2> How TinyXml works. </h2>
An example is probably the best way to go. Take:
@verbatim
    <?xml version="1.0" standalone=no>
    <!-- Our to do list data -->
    <ToDo>
         <Item priority="1"> Go to the <bold>Toy store!</bold></Item>
         <Item priority="2"> Do bills</Item>
    </ToDo>
@endverbatim
Its not much of a To Do list, but it will do. To read this file
(say "demo.xml") you would create a document, and parse it in:
@verbatim
    TiXmlDocument doc( "demo.xml" );
```

doc.LoadFile();

@endverbatim

And its ready to go. Now lets look at some lines and how they relate to the DOM.

```
@verbatim
<?xml version="1.0" standalone=no>
@endverbatim
```

The first line is a declaration, and gets turned into the TiXmlDeclaration class. It will be the first child of the document node.

This is the only directive/special tag parsed by by TinyXml. Generally directive targs are stored in TiXmlUnknown so the commands wont be lost when it is saved back to disk.

```
@verbatim
<!-- Our to do list data -->
@endverbatim
```

A comment. Will become a TiXmlComment object.

```
@verbatim
<ToDo>
@endverbatim
```

The "ToDo" tag defines a TiXmlElement object. This one does not have any attributes, but does contain 2 other elements.

```
@verbatim
<Item priority="1">
@endverbatim
```

Creates another TiXmlElement which is a child of the "ToDo" element. This element has 1 attribute, with the name "priority" and the value "1".

Go to the

A TiXmlText. This is a leaf node and cannot contain other nodes. It is a child of the "Item" TiXmlElement.

```
@verbatim
<bol>
bold>
@endverbatim
```

Another TiXmlElement, this one a child of the "Item" element.

Etc.

Looking at the entire object tree, you end up with: @verbatim

TiXmlDocument "demo.xml"

TiXmlDeclaration "version='1.0" "standalone=no"

```
" Our to do list data"
TiXmlComment
TiXmlElement
                         "ToDo"
    TiXmlElement
                         "Item"
                                    Attributes: priority = 1
         TiXmlText
                            "Go to the "
         TiXmlElement "bold"
                            "Toy store!"
             TiXmlText
                             "Item"
    TiXmlElement
                                         Attributes: priority=2
         TiXmlText
                                "Do bills"
```

@endverbatim

<h2> Documentation </h2>

The documentation is build with Doxygen, using the 'dox' configuration file.

<h2> License </h2>

TinyXml is released under the zlib license:

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

- 1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
- 2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
- 3. This notice may not be removed or altered from any source distribution.

<h2> References </h2>

The World Wide Web Consortium is the definitive standard body for XML, and there web pages contain huge amounts of information.

The definitive spec: http://www.w3.org/TR/2004/REC-xml-20040204/

I also recommend "XML Pocket Reference" by Robert Eckstein and published by OReilly...the book that got the whole thing started.

<h2> Contributors, Contacts, and a Brief History </h2>

Thanks very much to everyone who sends suggestions, bugs, ideas, and encouragement. It all helps, and makes this project fun. A special thanks to the contributors on the web pages that keep it lively.

So many people have sent in bugs and ideas, that rather than list here we try to give credit due in the "changes.txt" file.

TinyXml was originally written be Lee Thomason. (Often the "I" still in the documenation.) Lee reviews changes and releases new versions, with the help of Yves Berquin and the tinyXml community.

We appreciate your suggestions, and would love to know if you use TinyXml. Hopefully you will enjoy it and find it useful. Please post questions, comments, file bugs, or contact us at:

www.sourceforge.net/projects/tinyxml

Lee Thomason, Yves Berquin */ To compile on Ubuntu, install the following packages and then make.

libcurl4-gnutls-dev libxml2-dev libcrypto++-dev .



router.h

```
00001
00002 #ifndef _Router_h_
00003 #define _Router_h_
00004
00005 #include <iostream>
00006 #include <list>
00007 #include <cc++/xml.h>
00008 #include "network_socket.h" 00009 #include "comptorrentparser.h"
00010 #include "processor.h"
00011
00012 using namespace std; 00013 using namespace ost;
00014
00015 class Router : public Thread {
00016
00017 public:
00018
                Router(Controller* parent_) {
00019
00020
                          parent = parent_;
00021
00022
00023
                void run();
00024
00025 protected:
00026
00027
                bool get_next_orig_chunk();
00028
                bool get_next_comp_chunk();
00029
00030
                 int num_requested_orig_chunks;
00031
                int num_requested_comp_chunks;
00032
00033
                Controller* parent;
00034 };
00035
00036 #endif
```

Generated on Sat Mar 27 12:27:34 2010 for CompTorrent by







Main Page Classes Files
File List

SP2PCommon.h

```
00001 #ifndef _SP2PCommon_h_
00002 #define _SP2PCommon_h_
00003
00004 #include <iostream>
00005
00006 using namespace std;
00007
00008 // include cc++ and turn compression off
00009 #define NO COMPRESSION
00010 #include <cc++/common.h>
00011
00012 #define WORKING DIRECTORY
                                              "comptorrents/working"
00013 #define FINISHED_DIRECTORY
                                              "finished"
00014
00015 #define ORIG_DIRECTORY 00016 #define COMP_DIRECTORY
                                    "orig data"
                                    "comp_data"
00017
00018 #define DEFAULT_SLEEP 1
       #define DEFAULT_SLEEP_MSEC 1000
00019
00020
00021
00022 #define DEFAULT MAX CHUNK SIZE 3000000
00023
00024 #define MAX_INCOMING_CONNECTIONS 8
00025 #define MAX OUTGOING CONNECTIONS 4
00026
00027 #define WAIT TIMEOUT FOR CHUNKS 6000
00028
00029 #define NUM_PROCESSORS
                                                       1
00030
00031 #include <iostream>
00032 #include <cstdlib>
00033 #include <list>
00034 #include <map>
00035
00036 const string DEFAULT_PORT("60000");
00037
00038 const string COMP_DATA("comp_data");
00039 const string ORIG_DATA("orig_data");
00040 const string NO_ROUTE("0");
00041
00042
00043 using namespace std;
00044
00045 typedef map<string, string> filemap; 00046 typedef filemap::iterator filemap_iter;
00047
00048
00049
00050 // experiment settings
00051
00052
00053 const bool routing peeking = false;
00054
00055
00056 // end experiment settings
00057
00058 #endif
00059
00060
```

Generated on Sat Mar 27 12:27:34 2010 for CompTorrent by





known_peer Struct Reference

List of all members.

Public Member Functions

std::string	trim_spaces (std::string const &str)
	<pre>known_peer (string host_ip, string host_port, connection_type ct_=unconnected, string routeid_="0")</pre>
	known_peer (string host_ip, int host_port, connection_type ct_)
	known_peer (string name_, string uuid_, string ip_, string port_, string update_time_, unsigned long num_computed_chunks_, unsigned long num_original_chunks_, unsigned long num_connections_)
void	push_data_chunk (bool have_chunk)
void	push_orig_chunk (bool have_chunk)
string	comp_chunk_string ()
string	orig_chunk_string ()
known_peer &	<pre>operator= (known_peer &rhs)</pre>
bool	<pre>operator== (known_peer rhs)</pre>
bool	operator!= (known_peer rhs)
bool	operator> (known_peer rhs)
bool	operator< (known_peer rhs)
void	set_tried (bool tried_)
bool	have_tried ()

Public Attributes

vector< bool >	data_chunks
vector< bool >	orig_chunks
connection_type	ct
bool	tried
string	name
string	uuid
string	ip
string	port
string	update_time
unsigned long	num_computed_chunks
unsigned long	num_original_chunks
unsigned long	num_connections
string	routeid
void *	distorrent

The documentation for this struct was generated from the following file:

known_peer.h

Generated on Sat Mar 27 12:27:44 2010 for CompTorrent by 1.6.1

Files Main Page Classes **Class Members Class List**

known_peer Member List

This is the complete list of members for **known_peer**, including all inherited members.

This is the complete list of members for known peer, including an innerted members.		
comp_chunk_string() (defined in known_peer)	known_peer	[inline]
ct (defined in known_peer)	known_peer	
data_chunks (defined in known_peer)	known_peer	
distorrent (defined in known_peer)	known_peer	
have_tried() (defined in known_peer)	known_peer	[inline]
<pre>ip (defined in known_peer)</pre>	known_peer	
known_peer() (defined in known_peer)	known_peer	[inline]
<pre>known_peer(string host_ip, string host_port, connection_type ct_=unconnected, string routeid_="0") (defined in known_peer)</pre>	known_peer	[inline]
<pre>known_peer(string host_ip, int host_port, connection_type ct_) (defined in known_peer)</pre>	known_peer	[inline]
<pre>known_peer(string name_, string uuid_, string ip_, string port_, string update_time_, unsigned long num_computed_chunks_, unsigned long num_original_chunks_, unsigned long num_connections_) (defined in known_peer)</pre>	known_peer	[inline]
name (defined in known_peer)	known_peer	
num_computed_chunks (defined in known_peer)	known_peer	
num_connections (defined in known_peer)	known_peer	
num_original_chunks (defined in known_peer)	known_peer	
	known_peer	
<pre>operator<(known_peer rhs) (defined in known_peer)</pre>	known_peer	[inline]
<pre>operator=(known_peer &rhs) (defined in known_peer)</pre>	known_peer	[inline]
<pre>operator == (known_peer rhs) (defined in known_peer)</pre>	known_peer	[inline]
<pre>operator>(known_peer rhs) (defined in known_peer)</pre>	known_peer	[inline]
orig_chunk_string() (defined in known_peer)	known_peer	[inline]
-	known_peer	
• • •	known_peer	
	known_peer	
. – •– · · · · · · · · · · · · · · · · ·	known_peer	[inline]
· _ ,	known_peer	
_ · · · · · · · · · · · · · · · · · · ·	known_peer	[inline]
•	known_peer	
-· · · · · · · · · · · · · · · · · · ·	known_peer	[inline]
,	known_peer	
uuid (defined in known_peer)	known_peer	





orig_data_chunk Struct Reference

List of all members.

Public Member Functions

	<pre>orig_data_chunk (const orig_data_chunk &rhs)</pre>
orig_data_chunk &	<pre>operator= (const orig_data_chunk &rhs)</pre>

Public Attributes

string	md5
string	name
string	size
unsigned long	num_requests
unsigned long	num_sent
computation_state	comp_state
bool	orig_file_exists
bool	comp_file_exists
bool	requested

The documentation for this struct was generated from the following file:

• orig_data_chunk.h

Generated on Sat Mar 27 12:27:44 2010 for CompTorrent by 1.6.1



orig_data_chunk Member List

This is the complete list of members for **orig_data_chunk**, including all inherited members.

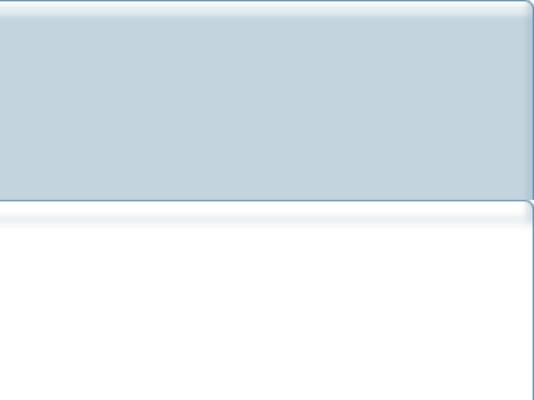
comp_file_exists (defined in orig_data_chunk)	orig_data_chunk
comp_state (defined in orig_data_chunk)	orig_data_chunk
md5 (defined in orig_data_chunk)	orig_data_chunk
name (defined in orig_data_chunk)	orig_data_chunk
num_requests (defined in orig_data_chunk)	orig_data_chunk
num_sent (defined in orig_data_chunk)	orig_data_chunk
<pre>operator=(const orig_data_chunk &rhs) (defined in orig_data_chunk)</pre>	<pre>orig_data_chunk [inline]</pre>
<pre>orig_data_chunk(const orig_data_chunk &rhs) (defined in orig_data_chunk)</pre>	<pre>orig_data_chunk [inline]</pre>
orig_data_chunk() (defined in orig_data_chunk)	<pre>orig_data_chunk [inline]</pre>
orig_file_exists (defined in orig_data_chunk)	orig_data_chunk
requested (defined in orig_data_chunk)	orig_data_chunk
size (defined in orig_data_chunk)	orig_data_chunk





.





tion marks" and 'ant \hat{A} ©.'/>	postrophe marks'. It al	so has <, >,
	tion marks" and 'alt ©.'/>	tion marks" and 'apostrophe marks'. It al t ©.'/>









trackerparser.h

```
00001 #ifndef _TrackerParser_h_
00002 #define _TrackerParser_h_
00003
00004
00005 #include <iostream>
00006 #include <cstdlib>
00007 #include <list>
00008 #include "tinyxml/tinystr.h"
00009 #include "tinyxml/tinyxml.h"
00010 #include "SP2PCommon.h"
00011 #include "known_peer.h"
00012 #include "orig data chunk.h"
00013 #include "controller.h"
00014
00015 #include <curl/curl.h>
00016
00017 class Controller; // forward declaration
00018
00019 class TrackerParser {
00020
00021 public:
00022
00023
              TrackerParser(Controller* s);
00024
              ~TrackerParser();
00025
00026
              void init();
00027
00028
              bool clear_xml();
00029
              bool load_xml();
00030
00031
              bool send work xml();
00032
              bool get_data_from_tracker(string url, string& data);
00033
00034
              bool report stats to tracker();
00035
00036
               void set_tracker_url(string tracker_url_) { tracker_url = tracker_url_; }
00037
              string get_tracker_url() { return tracker_url; }
00038
00039
               unsigned long get num known peers();
00040
              bool pop_next_known_peer(known_peer& kp);
00041
00042
               void copy_chunks_from_comptorrentparser();
00043
              bool prime_chunk_records();
00044
00045
              bool get_next_chunk_name(string& chunk_name);
00046
               // singular file notifiers
00047
              bool set_chunk_done(string chunk_name, string resulthash, string
00048
orig_chunk_hash, bool report);
00049
              bool report_file_exists(string resulthash, string filename);
00050
00051
               // batch file notifies
00052
               bool set_chunks_done(string comp_hash);
00053
              bool set_orig_data(string orig_data);
00054
00055
              bool suggest_orig_chunks(string lastchunk, string& xml_suggestions);
00056
00057
00058
              bool get_comp_hash(string chunk_name, string& resulthash);
               bool get_next_missing_orig_chunk_name(string& chunk_name);
00059
              bool get next missing comp chunk name(string& chunk name);
00060
00061
00062
              void register_node(string ip, string port, string uuid, string
comptorrentname, int routeid);
00063
              bool stats(string type, string statistic);
00064
00065
              bool report_connection(string server, string client);
00066
              bool suggest_peer_to_try(known_peer& kp);
00067
00068
              bool report_ipconnection(string client, string server);
00069
              bool ipconnection_exists(string client, string server);
00070
```

```
bool get route to file(const string chunk hash, string& routeid);
00072
                void debug() {
00073
00074
                          cout << "\n\n*** Tracker class stats ***\n";</pre>
                          cout << "stat_get_num_known_peers " << stat_get_num_known_peers <<</pre>
00075
endl;
00076
                          cout << "stat_pop_next_known_peer " << stat_pop_next_known_peer<</pre>
endl;
00077
                          cout << "stat_is_suitable_peer " << stat_is_suitable_peer<< endl;</pre>
                          cout << "stat_copy_chunks_from_comptorrentparser " <<</pre>
00078
stat_copy_chunks_from_comptorrentparser << endl;
00079
                          cout << "stat_prime_chunk_records " << stat_prime_chunk_records<</pre>
endl;
00080
                          cout << "stat_get_next_chunk_name " << stat_get_next_chunk_name<</pre>
endl;
                          cout << "stat_set_chunk_done " << stat_set_chunk_done<< endl;
cout << "stat_set_chunks_done " << stat_set_chunks_done<< endl;</pre>
00081
00082
                          cout << "stat_report_file_exists " << stat_report_file_exists<< endl;</pre>
00083
                          cout << "stat_get_comp_hash " << stat_get_comp_hash << endl;</pre>
00084
                          cout << "stat_get_next_missing_orig_chunk_name " <</pre>
00085
stat_get_next_missing_orig_chunk_name << endl;</pre>
00086
                          cout << "stat_get_next_missing_comp_chunk_name " <<</pre>
stat_get_next_missing_comp_chunk_name<< endl;</pre>
00087
                          cout << "stat_report_files_exists " << stat_report_files_exists <<</pre>
endl;
                          cout << "stat_register_node " << stat_register_node<< endl;
cout << "stat_report_connection " << stat_report_connection << endl;</pre>
00088
00089
                          cout << "stat_suggest_peer_to_try " << stat_suggest_peer_to_try <<</pre>
00090
endl:
00091
                          cout << "stat_get_route_to_file " << stat_get_route_to_file << endl;</pre>
                          cout << "stat_add_known_peer " << stat_add_known_peer << endl;
cout << "stat_known_peer_exists " << stat_known_peer_exists << endl;</pre>
00092
00093
                          cout << "stat_report_stats_to_tracker " <<</pre>
00094
stat_report_stats_to_tracker << endl;</pre>
00095
00096
                          cout << "stat_set_orig_data " << stat_set_orig_data << endl << endl;</pre>
                          cout << "total_in: " << stat_total_in << " total out: " <<</pre>
00097
stat_total_out << endl << endl;
00098
00099
00100
00101 protected:
00102
00103
                Mutex curl_mutex;
00104
00105
                bool get_xml_from_tracker();
00106
                bool send_data_to_tracker(string data);
00107
                bool post_data_to_tracker(string data);
00108
00109
                void add_known_peer(known_peer kp);
00110
                bool known_peer_exists(known_peer kp);
00111
00112
                void tokenize(string& str,vector<string>& tokens,string delimiters);
00113
00114
                TiXmlDocument doc;
00115
                Controller* parent;
00116
00117
                string tracker_url;
00118
                string tracker_xml;
00119
00120
                vector<known_peer> known_peers;
00121
                vector<orig data chunk> data chunks;
00122
00123
                unsigned long computed_chunks, original_chunks;
00124
00125
                unsigned original_hits;
00126
00127
00128
                int stat_get_num_known_peers;
                int stat_pop_next_known_peer;
00129
00130
                int stat_is_suitable_peer;
00131
                int stat_copy_chunks_from_comptorrentparser;
00132
                int stat_prime_chunk_records;
00133
                int stat_get_next_chunk_name;
00134
                int stat_set_chunk_done;
00135
                int stat_set_chunks_done;
00136
                int stat_report_file_exists;
00137
                int stat_get_comp_hash;
                int stat_get_next_missing_orig_chunk_name;
int stat_get_next_missing_comp_chunk_name;
00138
00139
00140
                int stat report files exists;
```

```
00141
                   int stat_register_node;
00142
                   int stat_report_connection;
                  int stat_suggest_peer_to_try;
int stat_get_route_to_file;
int stat_add_known_peer;
00143
00144
00145
                  int stat_known_peer_exists;
int stat_set_orig_data;
int stat_total_in;
00146
00147
00148
                   int stat_total_out;
00149
00150
                   int stat_report_stats_to_tracker;
00151 };
00152
00153 #endif
```



Q-

CompTorrent





File List

```
<?xml version="1.0" ?>
- <document>
   <English name="name" value="value">The world has many languages</English>
   <Russian name="название(имя)" value="ценность">Мир имеет много языков</Russian>
   <Spanish name="el nombre" value="el valor">el mundo tiene muchos idiomas
   <SimplifiedChinese name="名字" value="价值">世界有很多语言</SimplifiedChinese>
   < Русский название = "name" ценность = "value" > < имеет > < / Русский >
   <汉语 名字="name" 价值="value">世界有很多语言</汉语>
   <Heavy>"Mëtæl!"</Heavy>
 </document>
```

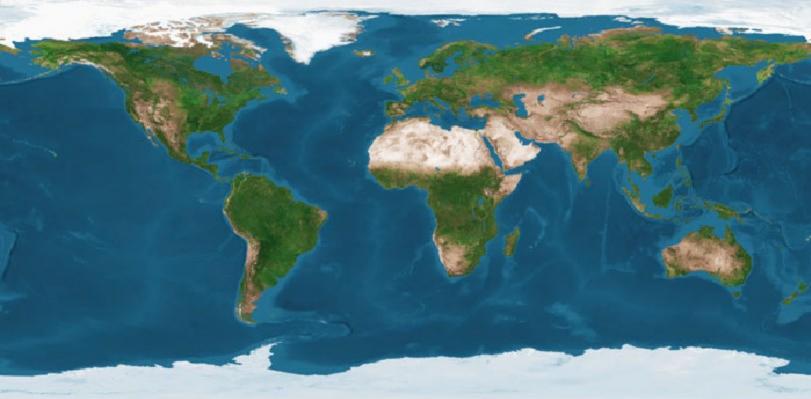


worker_thread.h

```
00001 #ifndef _WorkerThread_h_
00002 #define _WorkerThread_h_
00003
00004 #include <cc++/thread.h>
00005
00006 using namespace std;
00007 using namespace ost;
00008
00009 class WorkerThread: public Thread
00010 {
00011 public:
00012
00013
                WorkerThread();
00014
                ~WorkerThread();
00015
                virtual void run();
00016 };
00017
00018 #endif
```

Generated on Sat Mar 27 12:27:34 2010 for CompTorrent by





Appendix A

The source code for CompTorrent, including the database schema and doxygen documentation (including collaboration diagrams), accompanies the thesis electronically on a DVD.

Building Source Code

These instructions are for building CompTorrent using gcc 4.3.2 on Ubuntu 8.10.

It is assumed that sharutils has been installed on the machine (for uuencode/uudecode).

Obtain the following 3rd party libraries and compile and install them as per their individual instructions.

```
commoncpp2-1.7.3.tar.gz - configure, make, sudo make install
cryptopp560.zip - make, sudo make install
tinyxm1_2_5_3.tar.gz - make
curl-7.19.4.tar.gz - configure, make, sudo make install
```

Installing CompTracker

```
sudo ln -s /home/bcg/source/sp2p/trunk/comptracker/public_html/ comptracker
```

edit globalconfig.php to match your database connection.

```
$global_vars = array(

    "dbhost" => "localhost",
    "dbname" => "sp2p",
    "dbuser" => "root",
    "dbpass" => "77moke",
);
```

Tracker Database Schema

The tracker uses a MySQL database to maintain information about the computing projects its hosting. It provides much of the storage for the tracker interfaces as described in 7.1.2.2.

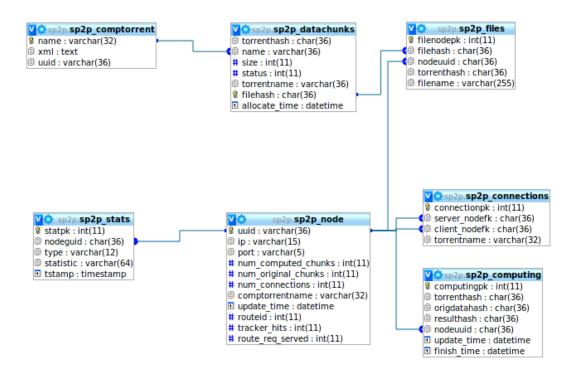
```
CREATE TABLE `sp2p_comptorrent` (
  `name` varchar(32) NOT NULL,
  `xml` text NOT NULL,
  `uuid` varchar(36) NOT NULL,
  PRIMARY KEY (`name`),
  KEY `uuid` (`uuid`)
);

CREATE TABLE `sp2p_computing` (
  `computingpk` int(11) NOT NULL auto_increment,
  `torrenthash` char(36) NOT NULL,
```

```
`origdatahash` char(36) NOT NULL,
   `resulthash` char(36) NOT NULL,
   `nodeuuid` char(36) NOT NULL,
  `update_time` datetime NOT NULL,
`finish_time` datetime NOT NULL,
   `filename` varchar(255) NOT NULL,
  PRIMARY KEY (`computingpk`),
  KEY `torrenthash` (`torrenthash`),
KEY `origdatahash` (`origdatahash`, `resulthash`, `nodeuuid`)
):
CREATE TABLE `sp2p_connections` (
   `connectionpk` int(11) NOT NULL auto_increment,
   `server_nodefk` char(36) NOT NULL,
`client_nodefk` char(36) NOT NULL,
  `torrentname` varchar(32) NOT NULL,
  PRIMARY KEY (`connectionpk`),
  KEY `server_nodefk` (`server_nodefk`, `client_nodefk`),
  KEY `torrentname` (`torrentname`),
  KEY `client_nodefk` (`client_nodefk`)
CREATE TABLE `sp2p_datachunks` (
   `torrenthash` char(36) NOT NULL,
   `name` varchar(36) NOT NULL,
`size` int(11) NOT NULL,
  `status` int(11) NOT NULL,
   `torrentname` varchar(36) NOT NULL,
  `filehash` char(36) NOT NULL,
  `allocate_time` datetime NOT NULL,
  `num_computed` int(11) NOT NULL,
PRIMARY KEY (`name`),
  KEY `torrenthash` (`torrenthash`)
CREATE TABLE `sp2p_files` (
  `filenodepk` int(11) NOT NULL auto_increment,
   `filehash` char(36) NOT NULL,
   `nodeuuid` char(36) NOT NULL,
   `torrenthash` char(36) NOT NULL,
  `filename` varchar(255) NOT NULL,
  PRIMARY KEY (`filenodepk`),
KEY `filehash` (`filehash`),
KEY `nodeuuid` (`nodeuuid`),
  KEY `torrenthash` (`torrenthash`)
);
CREATE TABLE `sp2p_ipconnections` (
   `ipconnectionspk` int(11) NOT NULL auto_increment,
   `client` varchar(15) NOT NULL,
`server` varchar(15) NOT NULL,
  PRIMARY KEY ('ipconnectionspk')
);
CREATE TABLE `sp2p_node` (
   uuid` varchar(36) NOT NULL,
   `ip` varchar(15) NOT NULL,
  `port` varchar(5) NOT NULL,
`num_computed_chunks` int(11) NOT NULL,
`num_original_chunks` int(11) NOT NULL,
   `num_connections` int(11) NOT NULL,
`comptorrentname` varchar(32) NOT NULL,
   `update_time` datetime NOT NULL,
   `routeid` int(11) NOT NULL,
   `tracker_hits` int(11) NOT NULL,
   `route_req_served` int(11) NOT NULL,
   `mutex` varchar(36) NOT NULL,
  PRIMARY KEY (`uuid`)
);
CREATE TABLE `sp2p_stats` (
   `statpk` int(11) NOT NULL auto_increment,
   `nodeguid` char(36) NOT NULL,
   `type` varchar(12) NOT NULL,
   `statistic` varchar(64) NOT NULL,
   `tstamp` timestamp NOT NULL default CURRENT_TIMESTAMP on update CURRENT_TIMESTAMP,
```

```
PRIMARY KEY (`statpk`),
  KEY `nodeguid` (`nodeguid`)
);
```

Tracker Database Schema Diagram



UML Class Diagrams

```
*known_files: filemap
*loaded_ok: bool
*my_peer_record: known_peer
*status_mutex: Mutex
#Processor: class
#CompTorrent: class
#CompTorrent: class
#CompTorrent: class
#CompTorrent: class
#mum_computed_chunks: long
#mum original_chunks: long
#parent: Controller*

*CompTorrentParser(parent_:Controller*)
*load_xml(g:strin): bool
*get_version(): string
*get_tracker_ur(): string
*get_tracker_ur(): string
*get_tracker_ur(): string
*get_algorithm_base64(): string
*get_algorithm_bas
```

```
Controller
             +connect_mutex: Mutex
+comp_torrent: CompTorrentParser*
+tracker: TrackerParser*
+parser_tracker_mutex: Mutex
             +get_next_comp_chunk_mutex: Mutex
+get_next_orig_chunk_mutex: Mutex
+comp_chunk_collection: Mutex
+orig_chunk_collection: Mutex
             +Router: class
#router: Router*
             #router: Router
#peers being considered: peer_type
#peers_considered_mutex: Mutex
   #peers_Deing_considered: peer_type
#peers_considered_mutex: Mutex
#backLog_mutex: Mutex
#comp_chunks_we_want: file_chunk_type
#comp_chunks_we_have: file_chunk_type
#orig_chunks_we_want: file_chunk_type
#orig_chunks_we_want: file_chunk_type
#request_backLog: file_request_type
#retwork_device: string
#routing_id: string
#waiting: bool
#num_incoming_connections: int
#num_outgoing_connections: int
#num_o
                #next route id: long
             #change: bool
                +Controller(index:bool,is processing:bool)
      +Controller()
+debug_stats(): void
+start_distributed(host_ip_:string,host_port_:string): void
+start_distributed(host_ip_:string,host_port_:string): void
+make_connection(host_ip_:string,host_port_:string): void
+attempt_outgoing_connection(host_ip_:string,host_port_:string): void
+attempt_incoming_connection(host_ip_:string,host_port_:string): void
+parse_torrent(file_path:string): bool
+create_working(): bool
+log(s:ostring): void
+log(s:string): void
+log(s:string): void
+get_num_connections(): int
+get_num_incoming_connections(): int
+get_num_outgoing_connections(): void
+increment_num_incoming_connections(): void
+increment_num_outgoing_connections(): void
                   +~Controller()
      +get_num_outgoing_connections(): int
+increment_num_incoming_connections(): void
+increment_num_outgoing_connections(): void
+decrement_num_outgoing_connections(): void
+decrement_num_incoming_connections(): void
+set_tracker_url(istring_connections(): void
+set_tracker_url(istring_connections(): void
+set_tracker_url(istring_connections(): void
+set_tracker_url(istring_connections(): void
+set_tracker_url(istring_connections(): void
+clear_tracker_data(): void
+vml_to_map(xml:string,xml_map:xmlmap&): static bool
+get_device_ip(device:string): static string
+set_routing_id(): string
+set_routing_id(): string
+get_next_routing_id(): string
+set_ip(ip_:string): void
+set_port(port_:string): void
+set_port(port_:string)
+trim_spaces(const:std::string): static string
+set_network_device(my_device:string): void
+get_network_device(my_device:string): void
+get_network_device(): string
+report_change(change_ibool): void
+change_occured(): bool
+have_comp_chunk(filename.string_regulthasb:string): void
report_change(change_:bool): void
+change_occured(): bool
+have_comp_chunk(fulname:string): bool
+add_comp_chunk(fulname:string,resulthash:string): void
+get_known_comp_chunks(): int
+remove_required_comp_chunk(fulname:string): bool
+add_required_comp_chunk(filename:string): bool
+add_required_comp_chunk(filename:string): void
+get_next_required_comp_chunk(fifile_chunk&): bool
+num_required_comp_chunk(fifile_chunk&): bool
+num_required_comp_chunk(s): int
+have_file(chunkname:string, filetype:string): void
+have_orig_chunk(chunkname:string): bool
+add_orig_chunk(filename:string, resulthash:string): void
+add_orig_chunk(filename:string): void
+get_known_orig_chunks_xml(xml:ostringstream&): void
+get_known_orig_chunks_(): int
+remove_required_orig_chunk(filename:string): void
+add_required_orig_chunk(filename:string): void
+add_required_orig_chunk(fifile_chunk&): bool
+num_required_orig_chunks(): int
+add_connected_peer_(ip:string,port:string,it)
+ound_trequired_orig_chunks(): int
+add_connected_peer_(ip:string,port:string): bool
+out_broadcast(msg:string): void
+remove_connected_peer(ip:string,port:string): bool
+out_broadcast(msg:string): void
+peer_under_consideration(ip:string,port:string): bool
+add_peer_under_consideration(ip:string,port:string): bool
+add_backlog_file(filename:string, filetype:string, who:CompTorrentPeer*): bool
+check_backlog_file(filename:string, filetype:string, who:CompTorrentPeer*): bool
```

Processor

#state: processor_state
#parent: Controller*
#current_chunk_name: string
#wait_timeout: int
#finished: bool

+Processor(s:Controller*)

+run(): void

+processing_finished(): bool #process_orig_chunk(chunk_name:string): void

Router

#num_requested_orig_chunks: int #num_requested_comp_chunks: int

#parent: Controller* +Router(parent_:Controller*)

+run(): void

#get_next_orig_chunk(): bool #get_next_comp_chunk(): bool

AbstractPeer

+out_box: MessageCollection
+in_box: MessageCollection
#current_wait: int
#die: bool #die: bool
#sync_mode: short
#dt: CompTorrentPeer*
#parent: Controller*
#virtcount: int
#max_chunk_size: int
#timeout: short
#num_messages: long #num_messages: long #threadid: int

+AbstractPeer(server:TCPSocket&,p:Controller*,node_type_:short)
+AbstractPeer(i:InetHostAddress,port:tpport_t,p:Controller*,node_type_:short)

+AbstractPeer(i:InetHostAddress,port:tpp
+AbstractPeer()
+kill_yourself(): void
+set_my_ip(ip:string,port:string): void
+throttle_wait(time_:int): void
+push_msg(msg:string): void
+pet_my_ip(): string
+get_my_port(): string
+get_parent(): Controller*
#run(): void
#log(s:string): void



CompTorrentPeer

#simplep2p: AbstractPeer #parent: Controller #num_messages: int #peer_type: short
#state: distributed_state #buddy_route: string #files_available: file_chunk_type

+CompTorrentPeer(s:Controller*,simplep2p :AbstractPeer*)

+~CompTorrentPeer()

+push_msg(msg:string): void +has_file_available(filename:string,type:string): bool

+debug_stats(): void

+get_buddy_route(): string +is_higher_route(): bool #add_available_file(g:strin,g:strin): bool

#string_find(needle:string,haystack:string): static bool #send_welcome_message(): void

#send noop(): void

#run(): void

#process_noop(intmp:string): bool

#process_file_update(intmp:string): bool
#process_welcome_message(intmp:string): bool
#process_connect_message(intmp:string): bool

#process_file_reply(intmp:string): bool

#process_file_request(intmp:string): bool
#process_overlay_message(intmp:string): bool
#process_overlay_update_message(intmp:string): bool

#ask_for_next_comp_file(): bool #ask_for_next_orig_file(): bool #save_file_reply(xml_map:xmlmap&): void

#parse_datachunks(xml:string): void

#trim_spaces(const:std::string): static string

Listener

```
#host_ip: string
#host_port: string
#parent: Controller*
```

+Listener(parent_:Controller*,host_ip_:string,host_port_:string)

+run(): void

#curl_mutex: Mutex +TrackerParser(s:Controller*) +-TrackerParser() +init(): void +clear_xml(): bool +send_work_xml(): bool +send_work_xml(): bool +set_data_from_tracker(url:string,data:string&): bool +report_stats_to_tracker(url:string): void +get_data_from_tracker(url:string): void +get_tracker_url(): string +get_num_known_peers(): unsigned long +pop_next_known_peers(): unsigned long +pop_next_known_peers(): unsigned long +pop_next_known_peers(): void +prime_chunk_records(): bool +get_next_chunk_name(chunk_name:string&): bool +get_next_chunk_name(chunk_name:string&): bool +get_next_chunk_name(chunk_name:string,filename:string, orig_chunk_hash:string, report:bool): bool +report_file_exists(resulthash:string, filename:string): bool +set_chunk_done(chunk_name:string, filename:string): bool +set_orig_data(orig_data:string): bool +set_orig_data(orig_data:string): bool +set_orig_data(orig_data:string): bool +set_ent_missing_orig_chunk_name(chunk_name:string&): bool +get_next_missing_orig_chunk_name(chunk_name:string&): bool +get_next_missing_orig_chunk_name(chunk_name:string&): bool +register_node(ip:string,port:string,uuid:string,comptorrentname:string,routeid:int): void +stats(type:string,statistic:string): bool +report_connection(cserver:string,client:string): bool +report_connection(eserver:string,client:string): bool +report_connection(eserver:string,client:string): bool +report_connection(eserver:string,client:string): bool +report_connection(eserver:string,client:string): bool +report_connection(client:string,server:string): bool +report_connection(client:string,server:string): bool +report_connection(client:string,server:string): bool +report_connection(client:string,server:string): bool +report_connection(client:string,server:string): bool +report_connection(client:string,server:string): bool +report_connection(client:string): bool +report_connection(client:string): bool +report_connection(client:string): bool +report_connection(client:string): bool +report_connection(client:string)

Appendix B

Amdahl's Law

Amdahl's Law (Amdahl 1967) explains that all programs have a limit to which they can be parallelised, and gives us an algorithm with which to calculate the maximum possible speedup of a parallel program. The limitation arises from the non parallelisable part of the application, the part that must be executed sequentially.

$$\mathbf{S} = \frac{1}{(1-P) + \frac{P}{N}}$$

Where:

- S is the speedup of the program,.
- P is the part of the application that can be parallelised,
- (1 P) is the part of the program which must be executed sequentially.
- N is the number of processing elements working on the task.

As N grows large the maximum speedup tends towards (1 - P). This gives an upper limit the amount of processors can be usefully utilised in a parallel computing exercise for a particular application or algorithm.

Gustafson-Barsis' Law

Gustafson-Barsis' Law allows for more flexibility over Amdahl's Law by allowing for the number of processors to change and the size of the problem to change as well during computation. This makes Gustafson-Barsis' Law much more suitable for modelling volunteer networks where nodes come and go and the overall problem size is often in flux.

$$S(P) = P - \alpha (P - 1)$$

Where:

- S is the speedup of the application.
- P is the number of processing elements.
- α is the part of the algorithm that is sequential.

Gustafson-Barsis' Law achieves this through considering the overall time of sequential execution rather than a fixed amount of sequential execution per processor.

Appendix C

Experimental Setup Instructions

This thesis contains experiments covering several different existing distributed computing systems. These systems are non-trivial to install, configure and operate. This appendix covers the installation instructions and settings used in detail for the experimental results presented in this thesis. These should serve as a starting point for building on or verification of this work for BOINC and Condor (See section 6.3 for instructions for CompTorrent).

In these examples the address 144.6.40.251 and 127.0.0.1 are used interchangeably to represent the master server machine. The user name "bcg" is used to represent the logged in account of the server administrator.

BOINC

The host system was Ubuntu Linux 6.06 with a full complement of GNU development tools, mysql and php installed via apt-get.

Server

```
Check out the code thus:
```

```
svn co http://boinc.berkeley.edu/svn/trunk/boinc
```

(Revision 14015 was the latest revision at time of experimentation).

Build the code:

```
./_autosetup
./configure --disable-client
make
```

Create a project:

```
./make_project --url_base http://144.6.40.251/boinc/ --db_host localhost
--db user root test1
```

Steps to complete installation:

1. Change Apache configuration (as root):

cat /home/bcg/projects/test1/test1.httpd.conf >> /etc/apache/httpd.conf && apachectl restart

(note: path to httpd.conf varies)

sudo cp /home/bcg/projects/test1/test1.httpd.conf /etc/apache2/sites-available/

2. Add to crontab (as bcg (my username) in this case)

```
0,5,10,15,20,25,30,35,40,45,50,55 * * * * * /home/bcq/projects/test1/bin/start --cron
```

(If cron cannot run "start", try using a helper script to set PATH and PYTHONPATH)

3. The project is configured with a test application.

To install this application (recommended) run:

cd /home/bcg/projects/test1
bin/xadd
bin/update_versions

To start, show status, and stop the project, run:

bin/start bin/status bin/stop

The project's URLs are (the url and project name will of course depend on installation):

Home page (and master URL): http://144.6.40.251/test1/

Administrative page:http://144.6.40.251/test1 ops/

Enable account creation – edit config.xml in the \sim /project/test1/ directory and set the the disable_account_creation element to 0

BOINC connects to Internet URLs in a few places in the code. This poses an annoying problem behind university proxy servers. To alleviate this, comment out the following lines of code:

/home/bcg/projects/test1/html/ops/index.php

Insert a /* at line 23 and a */ at the end of line 39.

/home/bcg/projects/test1/html/inc/user.inc

Modify the function get_other_projects at line 60 to return \$\suser and do nothing else eg:

```
function get_other_projects($user) {
    /* $cpid = md5($user->cross_project_id . $user->email_addr);
    $url = "http://boinc.netsoft-online.com/get_user.php?cpid=$cpid";
    $f = fopen($url, "r");
    if (!$f) {
        return $user;
    }
    $u = parse_user($f, $user);
    fclose($f);
    return $u; */
    return $user;
}
```

Client Side

Download boinc 5.10.21 i686-pc-linux-gnu.sh

```
./boinc --attach_project http://144.6.40.251/test1/de68da8503386b0232e664a27f2dad92
```

That should be all that is required however here are some problems encountered and solutions:

Problem: Getting the authenticator for each client.

Solution: Get it from the db.

Problem: No work from project. The feeder seemed to be dying as can be seen by the status.

bcg@rhdl-a2:~/projects/test1/bin\$./status

```
BOINC is ENABLED

DAEMON pid status lockfile disabled commandline
1 7907 NOT RUNNING UNLOCKED no feeder -d 3
2 7909 running locked no transitioner -d 3
3 7912 running locked no file_deleter -d 3
```

Tried running the feeder directly (as it would be by BOINC as defined in config.xml):

```
bcg@rhdl-a2:~/projects/test1/bin$ ./feeder -d 3
2007-11-20 16:42:29.7621 [normal ] Starting
shmctl RMID: Operation not permitted
2007-11-20 16:42:29.7631 [CRITICAL] can't destroy shmem
```

Solution:

change the group for the feeder thus:

```
bcg@rhdl-a2:~/projects/test1/bin$ sudo chgrp www-data feeder
bcg@rhdl-a2:~/projects/test1/bin$ sudo chmod g+xs feeder
```

But if that doesn't work, execute the start with sudo:

```
sudo bin/start
```

and that should be problem solved.

Here is an example for setting up a project using Mandelbrot:

Server:

Create the Project

```
cd /data/source/boinc/tools/
./make_project --url_base http://144.6.40.251/ --db_host localhost
--db user root mandelbrot16
```

Set up the Project Website

```
sudo cp mandelbrot16.httpd.conf /etc/apache2/sites-available/
sudo ln -s /etc/apache2/sites-available/mandelbrot16.httpd.conf
/etc/apache2/sites-enabled/mandelbrot16.httpd.conf
sudo apache2ctl restart
cd ~/projects/mandelbrot16
sudo chown www-data ~/projects/ -R
```

Enable Account Creation

Enable account creation – edit config.xml in the ~/project/mandelbrot16/ directory and set the disable_account_creation element to 0

Edit the Project

In this example, only Linux machines are being used, so par down the project xml to only include one target platform (or adjust for your circumstances).

Also, add in the name of the algorithm executable. In this case its Mandelbrot.

Edit config.xml

Make sure the db passwd element contains your db password

Compile the Wrapper

```
svn co http://boinc.berkeley.edu/svn/trunk/boinc_samples
cd boinc_samples/wrapper
ln -s `g++ -print-file-name=libstdc++.a`
make
cp boinc_samplaes/wrapper/wrapper
~/projects/mandelbrot16/apps/mandelbrot/wrapper_5.5_i686-pc-linux-
qnu
```

Copy the Algorithm

Also make sure to name your algorithm as per BOINC's naming convention so that it matches the target platforms set out in the project.xml.

```
mkdir ~/projects/mandelbrot16/apps/mandelbrot/
cp
/data/source/sp2p/experiments/mandelbrot/160/comptorrents/working/mandelbrot/mandelbr
ot ~/projects/mandelbrot16/apps/mandelbrot_5.5_i686-pc-linux-gnu
bin/xadd
bin/update_versions
```

Add Templates for Work Units and Results

Put the following into a file called result_template in ~/projects/mandelbrot16/templates/

```
<file_info>
<name><OUTFILE_0/></name>
```

138

```
<generated_locally/>
<upload_when_present/>
<max_nbytes>1000000</max_nbytes>
<url><UPLOAD_URL/></url>
</file_info>
<result>
<file_ref>
    <file_name><OUTFILE_0/></file_name>
    </file_ref>
    </file_ref>
    </file_ref>
</file_ref>
</file_ref>
</file_ref>
</file_ref>
</file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref></file_ref</file_ref></file_ref</fi>
```

Put the following into a file called work_unit_template in ~/projects/mandelbrot16/templates/

```
<file_info>
  <number>0</number>
  </file_info>
  <workunit>
  <file_ref>
    <file_number>0</file_number>
    <open_name>in</open_name>
  </file_ref>
  </workunit>
```

Add Work Units

The mandelbrot executable takes 2 command line arguments: an infile and an outfile.

The infile stipulates the parameters for the mandelbrot set being generated.

The outfile is a resulting jpeg.

For this example, there are 16 work units, each a file containing the settings for the overall Mandelbrot set as well as the region of the set to be calculated (so it can be split up and computed over multiple machines).

Copy each of these files into the ~/projects/mandelbrot16/download directory.

Now, tell BOINC about each one of these files by inserting each one of them as a work unit (using the templates from the previous step).

```
bin/create_work -appname mandelbrot -wu_name mandelbrot_00000001 -wu_template
templates/work_unit_template -result_template templates/result_template -min_quorum 1
-target nresults 1 mandelbrot 00000001
```

Do this changing the file and work unit name for each data set file eg. mandelbrot_00000002, mandelbrot_00000003 and so on.

Start BOINC

sudo bin/start

You can also stop it (sudo bin/stop) and get a status (sudo bin/status)

139

Create a Client User Account

Go to the project homepage (http://144.6.40.251/mandelbrot16/) and create an account.

If the server does not have public Internet access, as is often the case on a dedicated cluster, this page might take a while to load as it tries to get user info from the BOINC main site. You can hack this out editing the file user.inc (home/bcg/projects/test1/html/inc/user.inc). Modify the function get_other_projects at line 60 to return \$user and do nothing else eg:

```
function get_other_projects($user) {
    /* $cpid = md5($user->cross_project_id . $user->email_addr);
    $url = "http://boinc.netsoft-online.com/get_user.php?cpid=$cpid";
    $f = fopen($url, "r");
    if (!$f) {
        return $user;
    }
    $u = parse_user($f, $user);
    fclose($f);
    return $u; */
    return $user;
}
```

Since this is a dedicated cluster example, you like me will probably not bother setting up email on the servers. You need to edit the created user's database entry and also get the authentication key so command line clients can attach to the project.

Connect to the database using phpmyadmin or similar. Go to the mandelbrot16 database and browse the results in the user table.

Take note of the authenticator field value (eg. 84a35ba7615192bd3120019d8861ffac). You will need to to connect shortly.

Update the email validated field to contain a 1.

Client Machines:

Download boinc 5.10.21 i686-pc-linux-gnu.sh from the BOINC website.

Run it on the client machine from your home directory. This will install the runtime files into a BOINC directory.

In a terminal run:

```
./run_client
```

Alternatively, you can create the following script (from: http://blog.os-tools.net/? p=31):

```
#!/bin/sh
# /etc/init.d/boinc
# Start/stop/restart
test -x /home/bcg/BOINC/boinc || exit 0
case "$1" in
start)
echo "Starting BOINC."
cd "/home/bcg/BOINC" && exec ./boinc >>stdoutdae.txt
2>>stderrdae.txt &
stop)
echo "Stopping BOINC."
killall boinc
restart)
killproc boinc
sleep 2
cd "/home/bcg/BOINC" && exec ./boinc >>stdoutdae.txt
2>>stderrdae.txt &
;;
*)
echo "Usage: /etc/init.d/boinc {start|stop|restart}"
exit 2
esac
exit 0
```

Which will allow you to start and stop BOINC as a service – so you can maintain one terminal window only in which to run the client and then issue commands to it. This saves having 2 x n terminals open (where n = number of nodes). 16 cshh windows is bad enough without needing 32 open on a single desktop.

Attach to the Project

In another separate terminal window:

```
./boinc_cmd -project_attach http://144.6.40.251/mandelbrot16/mandelbrot16/cc5e5948e2ff9fe00dc2474d271753ad
```

Use your own authenticator at this point that you noted down earlier.

You can also detach from the project later on by running:

```
./boinc cmd -project http://144.6.40.251/mandelbrot16/ detach
```

To later reset projects:

Stop BOINC server.

Empty the workunit and result tables in the database. empty the ~projects/mandelbrot16/upload directory Add work units again.

Start BOINC server.

Condor

Master Machine

Download condor-6.8.7-linux-x86-rhel.tar.gz and uncompress it.

```
cd /data/condor-6.8.7
mkdir condor root
mkdir condor_local
sudo ./condor_configure --install-dir=/data/condor-6.8.7/condor_root/
 --type=manager,submit --local-dir=/data/condor-6.8.7/condor_local/ --owner=bcg
--install=/data/condor-6.8.7/release.tar
export CONDOR_CONFIG=/data/condor-6.8.7/condor_root/etc/condor_config
Edit condor_root/etc/condor_config:
Set RELEASE DIR to /data/condor-6.8.7/condor root/
Set HOSTALLOW_WRITE to *
Set HOSTALLOW ADMINISTRATOR = $(FULL HOSTNAME)
Start Condor:
sudo condor root/sbin/condor master
Stop Condor:
sudo condor root/sbin/condor off -master
Check that its running:
ps -ef | egrep condor_
bcg@rhdl-a2:/data/condor-6.8.7$ ps -ef | egrep condor_
bcg 24421 1 0 12:25 ? 00:00:00 condor_root/sbin/condor_master
bcg 24422 24421 0 12:25 ? 00:00:00 condor_collector -f
bcg 24423 24421 0 12:25 ? 00:00:00 condor_negotiator -f
```

```
24424 24421 0 12:25 ? 00:00:00 condor_schedd -f
24425 24421 7 12:25 ? 00:00:07 condor_startd -f
24475 5431 0 12:27 pts/0 00:00:00 grep -E condor_
bcg
bcg
```

Create a job

```
# file name: mandelbrot16.condor
# Condor submit description file for mandelbrot
Executable = path_to/mandelbrot
Universe = vanilla
Universe = vanilla
Error = logs/err.$(cluster)
Output = logs/out.$(cluster)
= logs/log.$(cluster)
should_transfer_files = YES
when_to_transfer_output = ON_EXIT
transfer_input_files = files/mandelbrot_00000001
Arguments
                = mandelbrot_00000001 mandelbrot_00000001_out
Queue
create a directory to put the job in.
mkdir mandelbrot16
cd mandelbrot16
mkdir logs
mkdir files
Submit a job
condor\_root/bin/condor\_submit\ mandelbrot16/mandelbrot16.condor
Check on jobs
All jobs:
```

bin/condor_q

A job:

bin/condor_q 3

Client Machine

Download condor-6.8.7-linux-x86-rhel.tar.gz and uncompress it.

```
cd /data/condor-6.8.7
mkdir condor_root
mkdir condor_local
sudo ./condor_confiqure --install-dir=/home/bcq/condor-6.8.7/condor_root/
--type=execute --local-dir=/home/bcg/condor-6.8.7/condor_local/ --owner=bcg
--install=/home/bcg/condor-6.8.7/release.tar
{\tt export~CONDOR\_CONFIG=/home/bcg/condor-6.8.7/condor\_root/etc/condor\_config}
Edit condor-6.8.7/condor_root/etc/condor_config
Set UID_DOMAIN = $ (FULL_HOSTNAME)
Set FILESYSTEM_DOMAIN=$(FULL_HOSTNAME)
Set HOSTALLOW_ADMINISTRATOR = $(FULL_HOSTNAME)
```

Set HOSTALLOW_WRITE to *

Edit condor-6.8.7/condor_local/condor_config.local

Set CONDOR_HOST = 144.6.40.251SET UID_DOMAIN and FILESYSTEM_DOMAIN to \$(FULL_HOSTNAME)

NETWORK_INTERFACE = 144.6.40.115
WANT_SUSPEND = FALSE
CONTINUE = TRUE
SUSPEND = FALSE
PREEMPT = FALSE
START=TRUE

Appendix D

WAN-DC Extras

Sequential read/write with underlying network changes (synthetic)

This is a variation on the earlier Sequential read/write test with the addition of modulating the performance of the underlying network.

Test Description	Perform the Sequential read/write test whilst changing the bandwidth and latency properties of the underlying network.
Test Aim	To observe how the system performs distributing test and result data under a variety of network conditions.
Input Data	A freely, Internet-available dataset of a suitable size for the test.
	Suggested sizes ranging from 100kb, 1000kb, 10Mb, 100Mb, 1Gb.
	The Internet Archive is a suggested repository for source data.
Output Data	Wall clock time to complete the file distribution from start to finish.
	Mbytes/sec if the tested system provides its own figure.
Method	1. Homogeneous
	Modulate the bandwidth of the underlying network to represent commonly available device bandwidths ranging from modem speeds to gigabit Ethernet.
	All devices should be set to the same speed.
	Apply the method of the sequential read/write for both uniform size and mixed data unit sizes.
	2. Heterogeneous
	Modulate the bandwidth of the underlying network to represent commonly available device bandwidths ranging from modem speeds to gigabit Ethernet.
	All devices should be set to a range of different speeds to represent a known mix of devices.
	Apply the method of the sequential read/write for both uniform size and mixed data unit sizes.
L	

Input/Output Intensive

The benchmark presented here is designed to compare the ability of cluster technologies to distributed data between nodes on the network.

Sequential read/write (synthetic / application)

The aim of this benchmark is to test how quickly a distributed computing system can copy data between nodes. For systems that use NFS (or similar) as their file system, this will ostensibly be a test of that technology, the network connection and the cluster hardware. For systems that divide data into discrete units and share it as packets, this will enable the measurement of the packet protocol, the network connection and the cluster hardware.

To gain an understanding of the efficiency of the system to distribute test and result data across the network using a variety of data set sizes. A freely, Internet-available dataset of a suitable size for the test. Suggested sizes ranging from 100kb, 1000kb, 10Mb, 100Mb, 1Gb. The Internet Archive is a suggested repository for source data.
Suggested sizes ranging from 100kb, 1000kb, 10Mb, 100Mb, 1Gb.
The Internet Archive is a suggested repository for source data.
Wall clock time to complete the file distribution from start to finish.
Mbytes/sec if the tested system provides its own figure.
1. Uniform size
For each data set size, each set should be split into smaller uniform elements to simulate it being a unit of work for processing.
Suggested sizes ranging from 512bytes, 1kb, 10kb, 100kb, 256kb, 1000kb, 10Mb, 20Mb, 50Mb, 100Mb.
2. Mix
For each data set size, each set should be split into smaller random sized elements to simulate it being a unit of work for processing.
Suggested sizes including 512bytes, 1kb, 10kb, 100kb, 256kb,1000kb, 10Mb, 20Mb, 50Mb, 100Mb.

BitTorrent (theoretical maximum)

Test Description	Obtain bandwidth figures (Mbytes/sec) and wall clock time to distribute data over the cluster using the BitTorrent protocol.
Test Aim	This test is an attempt to gain a best estimate of maximum disk and network performance of the cluster hardware. This is to serve as a working theoretical maximum for what a distributed computing system could hope to achieve.
Input Data	A freely, Internet-available dataset of a suitable size for the test. Suggested sizes ranging from 100kb, 1000kb, 10Mb, 100Mb, 1Gb. The Internet Archive is a suggested repository for source data.
Output Data	BitTorrent timing output for each worker node in the cluster (produced by application with results to be tabulated by tester). Wall clock time to complete the file distribution from start to finish.

Method	Application should be run from 1 worker to all worker machines with a reasonable number of steps to meet a stated confidence interval (95% suggested). Each step should be repeated multiple times in order to gauge variability. Statistics provided should carefully describe the final values given in terms of their origin.
	Care should be taken to ensure that one of the tested message chunk sizes corresponds to one of the sizes tried in sequential read/write. BitTorrent's usual size is 256kb.

Test Description	Perform the Mandelbrot benchmark under varying degrees of network load.
Test Aim	Load the network with traffic to investigate the effect of heavy traffic up to a denial of service attack on various components of the distributed computing system.
Input Data	Nil
Output Data	Wall clock time to complete the Mandelbrot set.
Method	1. General load Load the network with varying degrees of traffic using a tool such as WebLOAD. 2. Denial of Service Load individual parts of the distributed computing system with varying degrees of traffic. Individual clients, server components, name servers and other components, as per the system being tested, should be loaded in turn to examine the effect it has on performance.

Failure and Malfeasance

Distributed computing systems that operate over wide area networks must consider the uncertain nature of nodes providing services to the system.

Erroneous Results (synthetic)

Incorrect results

Test Description	Perform the Mandelbrot benchmark with deliberately introduced errors at a known probabilistic rate.
Test Aim	To understand how the system copes with varying degrees of incorrect results and how node exclusion policies can effect the overall process.
Input Data	A set of files each representing the region of the Mandelbrot set to calculate.
	There are three separate parameter sets to this bench mark. 1. The overall set size being generated. 2. The number of segments the set is to be split into to be computed in parallel. 3. The region of the Mandelbrot set to be calculated. 4. The probability of this work unit being calculated incorrectly. Code has been provided to randomly change some points within the calculated work unit at a particular probabilistic rate. Start with a low probability, i.e. 0.0001, and then step up towards 1 in reasonable increments.

	Depending on items 2 and 3, each work unit can be either relatively uniform or provide a mix ranging from intensive computation to near trivial. Two datasets are provided, Mandelbrot_A and Mandelbrot_B, which illustrate each option.
Output Data	Image files of the Mandelbrot set. The wall clock time to produce the set.
Method	Perform this test using one of the same sets of parameters as the standard Mandelbrot to give a starting point.
	1. Random node failure
	Arrange the work units so that all nodes have the same probability of failure.
	2. Bad node
	Select a single node as random to be a source of error at a probabilistic rate.

First work unit (synthetic derived from application)

Clock time taken for a work unit to be completed when measured from the start of computation.

Test Description	Obtain wall clock times for a first work unit to be received.
Test Description	
Test Aim	To investigate the lag, setup or joining time that occurs in a distributed computing system. This is especially important for comparing emerging systems that rely on a decentralised approach where a bootstrapping phase may occur as a node joins the network.
Input Data	Data can be gathered from the results of other benchmark experiments from both a server or node perspective. That is, the time taken for the first completed work unit to be received. Or, the time between starting a new worker node and actually receiving data to process (where data is available).
Output Data	The elapsed time in microseconds.
Method	Application should be run from 1 worker to all worker machines with a reasonable number of steps to meet a stated confidence interval (95% suggested) for each work unit length. Each step should be repeated multiple times in order to gauge variability. Statistics provided should carefully describe the final values given in terms of their origin.
	Care should be taken to ensure that the test is fair and that all systems being compared are set to begin computation as soon as possible and that network load is comparable between tests.

Appendix E

Data Sets

Transcode

The file CC_1914_03_26_CruelCruelLove.mpg is available from archive.org and has been split using the mpgtx tool thus:

mpgtx -200 .../CC_1914_03_26_CruelCruelLove.mpg -b CC_1914_03_26_CruelCruelLove

The individual files as used here are also available with this thesis and listed below with their sizes in bytes.

1035738 chunk-001.mpg	899402 chunk-035.mpg	992574 chunk-068.mpg
961558 chunk-002.mpg	996380 chunk-036.mpg	998936 chunk-069.mpg
997946 chunk-003.mpg	992284 chunk-037.mpg	741322 chunk-070.mpg
836572 chunk-004.mpg	749924 chunk-038.mpg	995420 chunk-071.mpg
995420 chunk-005.mpg	933264 chunk-039.mpg	993564 chunk-072.mpg
995420 chunk-006.mpg	1024964 chunk-040.mpg	980840 chunk-073.mpg
747462 chunk-007.mpg	1006002 chunk-041.mpg	813808 chunk-074.mpg
993884 chunk-008.mpg	746822 chunk-042.mpg	998552 chunk-075.mpg
995834 chunk-009.mpg	989344 chunk-043.mpg	989088 chunk-076.mpg
997626 chunk-010.mpg	1000408 chunk-044.mpg	748550 chunk-077.mpg
858346 chunk-011.mpg	799612 chunk-045.mpg	998680 chunk-078.mpg
999768 chunk-012.mpg	975434 chunk-046.mpg	993564 chunk-079.mpg
747048 chunk-013.mpg	1020616 chunk-047.mpg	997020 chunk-080.mpg
991900 chunk-014.mpg	991934 chunk-048.mpg	748388 chunk-081.mpg
999192 chunk-015.mpg	744616 chunk-049.mpg	991486 chunk-082.mpg
747304 chunk-016.mpg	994524 chunk-050.mpg	999704 chunk-083.mpg
998424 chunk-017.mpg	998488 chunk-051.mpg	1025762 chunk-084.mpg
993756 chunk-018.mpg	995804 chunk-052.mpg	745512 chunk-085.mpg
998202 chunk-019.mpg	743720 chunk-053.mpg	993692 chunk-086.mpg
749156 chunk-020.mpg	998138 chunk-054.mpg	995676 chunk-087.mpg

993884 chunk-021.mpg	976458 chunk-055.mpg	746344 chunk-088.mpg
1004086 chunk-022.mpg	798080 chunk-056.mpg	1146148 chunk-089.mpg
812272 chunk-023.mpg	1000792 chunk-057.mpg	795264 chunk-090.mpg
988256 chunk-024.mpg	997626 chunk-058.mpg	998872 chunk-091.mpg
997370 chunk-025.mpg	922678 chunk-059.mpg	995420 chunk-092.mpg
996666 chunk-026.mpg	799100 chunk-060.mpg	854894 chunk-093.mpg
812310 chunk-027.mpg	930640 chunk-061.mpg	904932 chunk-094.mpg
980042 chunk-028.mpg	997562 chunk-062.mpg	996986 chunk-095.mpg
995706 chunk-029.mpg	994938 chunk-063.mpg	995676 chunk-096.mpg
841082 chunk-030.mpg	916350 chunk-064.mpg	937322 chunk-097.mpg
1000920 chunk-031.mpg	992796 chunk-065.mpg	830658 chunk-098.mpg
747876 chunk-032.mpg	747014 chunk-066.mpg	1016042 chunk-099.mpg
997178 chunk-033.mpg	997690 chunk-067.mpg	703680 chunk-100.mpg
996218 chunk-034.mpg		

POV-Ray

The files benchmark.pov and benchmark.ini are widely available and accompany this thesis.

Mandelbrot

A typical configuration of the Mandelbrot set is rendered in vertical slices of 40 pixels wide with an overall images size of 6400 by 4800.

```
Mandelbrot_00000001:
```

```
-2.3333166666666667 1.0000166666666666666666666666666666 -1.25 1.25 0 0 40 4800 1024 6400 4800 mandelbrot_00000002 :
```

...

mandelbrot_00000159:

-2.333316666666667 1.0000166666666666 -1.25 1.25 6320 0 6360 4800 1024 6400 4800

mandelbrot 00000160:

-2.3333166666666667 1.0000166666666666 -1.25 1.25 6360 0 6400 4800 1024 6400 4800

One Second

Each file contains an increasing integer, starting from one, representing the number of seconds the processor should wait for.

Appendix F

Experimental Results

Please see accompanying file bcg_phd_thesis_appendix_F.pdf

Appendix F

Experimental Results

Povray

Condor

Condor Pov	Condor Povray 1 Machine									
Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A
1	1202282215	0	1202299058	0	1202457878	0	1202475903	0	1202493925	0
2	1202282418	203	1202299265	207	1202458080	202	1202476105	202	1202494127	202
3	1202282636	421	1202299483	425	1202458298	420	1202476323	420	1202494345	420
4	1202282857	642	1202299704	646	1202458519	641	1202476544	641	1202494566	641
5	1202283085	870	1202299932	874	1202458746	868	1202476772	869	1202494794	869
6	1202283338	1123	1202300186	1128	1202459000	1122	1202477025	1122	1202495047	1122
7	1202283608	1393	1202300456	1398	1202459270	1392	1202477296	1393	1202495318	1393
8	1202283864	1649	1202300712	1654	1202459526	1648	1202477551	1648	1202495573	1648
9	1202284141	1926	1202300988	1930	1202459802	1924	1202477828	1925	1202495851	1926
10	1202284407	2192	1202301255	2197	1202460069	2191	1202478095	2192	1202496117	2192
11	1202284671	2456	1202301519	2461	1202460332	2454	1202478359	2456	1202496380	2455
12	1202284971	2756	1202301818	2760	1202460632	2754	1202478659	2756	1202496680	2755
13	1202285273	3058	1202302120	3062	1202460934	3056	1202478960	3057	1202496982	3057
14	1202285620	3405	1202302467	3409	1202461281	3403	1202479307	3404	1202497329	3404
15	1202285977	3762	1202302824	3766	1202461638	3760	1202479664	3761	1202497686	3761
16	1202286298	4083	1202303145	4087	1202461959	4081	1202479984	4081	1202498007	4082
17	1202286611	4396	1202303457	4399	1202462272	4394	1202480297	4394	1202498320	4395
18	1202286914	4699	1202303761	4703	1202462575	4697	1202480600	4697	1202498623	4698
19	1202287198	4983	1202304044	4986	1202462858	4980	1202480883	4980	1202498906	4981
20	1202287488	5273	1202304334	5276	1202463148	5270	1202481173	5270	1202499197	5272
21	1202287731	5516	1202304577	5519	1202463390	5512	1202481416	5513	1202499439	5514
22	1202287984	5769	1202304831	5773	1202463644	5766	1202481670	5767	1202499692	5767
23	1202288264	6049	1202305108	6050	1202463922	6044	1202481948	6045	1202499970	6045
24	1202288553	6338	1202305397	6339	1202464211	6333	1202482237	6334	1202500259	6334
25	1202288783	6568	1202305627	6569	1202464441	6563	1202482468	6565	1202500489	6564
26	1202289004	6789	1202305847	6789	1202464661	6783	1202482687	6784	1202500709	6784
27	1202289258	7043	1202306101	7043	1202464916	7038	1202482941	7038	1202500963	7038
28	1202289578	7363	1202306421	7363	1202465236	7358	1202483262	7359	1202501285	7360
29	1202289885	7670	1202306728	7670	1202465542	7664	1202483568	7665	1202501591	7666
30	1202290099	7884	1202306944	7886	1202465757	7879	1202483782	7879	1202501806	7881
31	1202290305	8090	1202307150	8092	1202465963	8085	1202483988	8085	1202502012	8087
32	1202290506	8291	1202307350	8292	1202466163	8285	1202484189	8286	1202502212	8287
33	1202290699	8484	1202307544	8486	1202466357	8479	1202484382	8479	1202502406	8481
34	1202290872	8657	1202307717	8659	1202466529	8651	1202484554	8651	1202502579	8654
35	1202291038	8823	1202307884	8826	1202466696	8818	1202484721	8818	1202502746	8821
36	1202291218	9003	1202308064	9006	1202466876	8998	1202484901	8998	1202502928	9003
37	1202291394	9179	1202308241	9183	1202467052	9174	1202485077	9174	1202503117	9192
38	1202291572	9357	1202308419	9361	1202467231	9353	1202485255	9352	1202503311	9386
39	1202291773	9558	1202308620	9562	1202467431	9553	1202485456	9553	1202503526	9601
40	1202291998	9783	1202308850	9792	1202467656	9778	1202485680	9777	1202503751	9826
41	1202292236	10021	1202309089	10031	1202467895	10017	1202485919	10016	1202503990	10065
42	1202292498	10283	1202309352	10294	1202468158	10280	1202486181	10278	1202504252	10327
43	1202292785	10570	1202309638	10580	1202468444	10566	1202486468	10565	1202504538	10613
44	1202293069	10854	1202309923	10865	1202468729	10851	1202486752	10849	1202504822	10897
45	1202293360	11145	1202310213	11155	1202469019	11141	1202487043	11140	1202505113	11188

46	1202293647	11432	1202310500	11442	1202469306	11428	1202487330	11427	1202505400	11475
47	1202293939	11724	1202310794	11736	1202469599	11721	1202487623	11720	1202505692	11767
48	1202294219	12004	1202311073	12015	1202469878	12000	1202487903	12000	1202505971	12046
49	1202294510	12295	1202311364	12306	1202470171	12293	1202488195	12292	1202506263	12338
50	1202294806	12591	1202311661	12603	1202470467	12589	1202488491	12588	1202506559	12634
51	1202295090	12875	1202311944	12886	1202470751	12873	1202488775	12872	1202506842	12917
52	1202295397	13182	1202312251	13193	1202471057	13179	1202489081	13178	1202507149	13224
53	1202295732	13517	1202312586	13528	1202471393	13515	1202489417	13514	1202507485	13560
54	1202296028	13813	1202312882	13824	1202471688	13810	1202489713	13810	1202507781	13856
55	1202296272	14057	1202313126	14068	1202471933	14055	1202489957	14054	1202508026	14101
56	1202296509	14294	1202313363	14305	1202472169	14291	1202490194	14291	1202508263	14338
57	1202296745	14530	1202313599	14541	1202472405	14527	1202490430	14527	1202508499	14574
58	1202296976	14761	1202313830	14772	1202472636	14758	1202490661	14758	1202508729	14804
59	1202297213	14998	1202314067	15009	1202472873	14995	1202490898	14995	1202508967	15042
60	1202297490	15275	1202314344	15286	1202473150	15272	1202491174	15271	1202509243	15318
61	1202297748	15533	1202314601	15543	1202473407	15529	1202491432	15529	1202509501	15576
62	1202297985	15770	1202314838	15780	1202473644	15766	1202491669	15766	1202509738	15813
63	1202298209	15994	1202315063	16005	1202473869	15991	1202491893	15990	1202509962	16037
64	1202298424	16209	1202315277	16219	1202474083	16205	1202492108	16205	1202510177	16252

Condor Povray	2 Machines									
Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A
1	1202500489	0	1202514348	0	1202523534	0	1202532851	0	1202541579	0
2	1202500709	220	1202514512	164	1202523699	165	1202533007	156	1202541744	165
3	1202500963	474	1202514623	275	1202523810	276	1202533126	275	1202541855	276
4	1202501285	796	1202514832	484	1202524019	485	1202533327	476	1202542064	485
5	1202501591	1102	1202514956	608	1202524143	609	1202533459	608	1202542187	608
6	1202501806	1317	1202515047	699	1202524233	699	1202533542	691	1202542278	699
7	1202502012	1523	1202515179	831	1202524366	832	1202533683	832	1202542410	831
8	1202502212	1723	1202515248	900	1202524434	900	1202533742	891	1202542479	900
9	1202502406	1917	1202515389	1041	1202524576	1042	1202533893	1042	1202542620	1041
10	1202502579	2090	1202515420	1072	1202524606	1072	1202533914	1063	1202542652	1073
11	1202502746	2257	1202515570	1222	1202524757	1223	1202534073	1222	1202542800	1221
12	1202502928	2439	1202515600	1252	1202524786	1252	1202534094	1243	1202542832	1253
13	1202503117	2628	1202515761	1413	1202524947	1413	1202534264	1413	1202542991	1412
14	1202503311	2822	1202515778	1430	1202524964	1430	1202534273	1422	1202543011	1432
15	1202503526	3037	1202515979	1631	1202525165	1631	1202534482	1631	1202543209	1630
16	1202503751	3262	1202516003	1655	1202525189	1655	1202534498	1647	1202543235	1656
17	1202503990	3501	1202516238	1890	1202525424	1890	1202534740	1889	1202543469	1890
18	1202504252	3763	1202516269	1921	1202525452	1918	1202534760	1909	1202543498	1919
19	1202504538	4049	1202516548	2200	1202525736	2202	1202535045	2194	1202543779	2200
20	1202504822	4333	1202516552	2204	1202525737	2203	1202535051	2200	1202543783	2204
21	1202505113	4624	1202516840	2492	1202526025	2491	1202535335	2484	1202544070	2491
22	1202505400	4911	1202516865	2517	1202526051	2517	1202535364	2513	1202544096	2517
23	1202505692	5203	1202517132	2784	1202526318	2784	1202535628	2777	1202544362	2783
24	1202505971	5482	1202517168	2820	1202526355	2821	1202535668	2817	1202544400	2821
25	1202506263	5774	1202517424	3076	1202526609	3075	1202535919	3068	1202544652	3073
26	1202506559	6070	1202517490	3142	1202526677	3143	1202535989	3138	1202544722	3143
27	1202506842	6353	1202517707	3359	1202526892	3358	1202536202	3351	1202544936	3357
28	1202507149	6660	1202517822	3474	1202527010	3476	1202536322	3471	1202545055	3476
29	1202507485	6996	1202518044	3696	1202527228	3694	1202536538	3687	1202545272	3693
30	1202507781	7292	1202518143	3795	1202527331	3797	1202536643	3792	1202545376	3797
31	1202508026	7537	1202518291	3943	1202527473	3939	1202536782	3931	1202545516	3937
32	1202508263	7774	1202518402	4054	1202527588	4054	1202536901	4050	1202545633	4054
33	1202508499	8010	1202518528	4180	1202527709	4175	1202537018	4167	1202545753	4174
34	1202508729	8240	1202518652	4304	1202527837	4303	1202537151	4300	1202545883	4304
35	1202508967	8478	1202518766	4418	1202527947	4413	1202537256	4405	1202545990	4411
36	1202509243	8754	1202518953	4605	1202528138	4604	1202537451	4600	1202546184	4605

37	1202509501	9012	1202519024	4676	1202528205	4671	1202537514	4663	1202546248	4669
38	1202509738	9249	1202519211	4863	1202528395	4861	1202537708	4857	1202546441	4862
39	1202509962	9473	1202519248	4900	1202528429	4895	1202537738	4887	1202546472	4893
40	1202510177	9688	1202519444	5096	1202528627	5093	1202537941	5090	1202546673	5094
41	1202510980	10491	1202520168	5820	1202529490	5956	1202538213	5362	1202547536	5957
42	1202511033	10544	1202520217	5869	1202529522	5988	1202538262	5411	1202547568	5989
43	1202511198	10709	1202520386	6038	1202529708	6174	1202538431	5580	1202547754	6175
44	1202511273	10784	1202520458	6110	1202529762	6228	1202538502	5651	1202547807	6228
45	1202511426	10937	1202520614	6266	1202529936	6402	1202538659	5808	1202547981	6402
46	1202511549	11060	1202520733	6385	1202530038	6504	1202538778	5927	1202548083	6504
47	1202511696	11207	1202520884	6536	1202530206	6672	1202538929	6078	1202548252	6673
48	1202511826	11337	1202521011	6663	1202530315	6781	1202539056	6205	1202548360	6781
49	1202511973	11484	1202521161	6813	1202530483	6949	1202539206	6355	1202548528	6949
50	1202512115	11626	1202521300	6952	1202530605	7071	1202539345	6494	1202548649	7070
51	1202512237	11748	1202521425	7077	1202530747	7213	1202539470	6619	1202548793	7214
52	1202512440	11951	1202521625	7277	1202530930	7396	1202539671	6820	1202548974	7395
53	1202512539	12050	1202521727	7379	1202531049	7515	1202539771	6920	1202549094	7515
54	1202512817	12328	1202522002	7654	1202531307	7773	1202540049	7198	1202549351	7772
55	1202512897	12408	1202522084	7736	1202531406	7872	1202540129	7278	1202549451	7872
56	1202513165	12676	1202522351	8003	1202531655	8121	1202540397	7546	1202549700	8121
57	1202513211	12722	1202522397	8049	1202531719	8185	1202540441	7590	1202549765	8186
58	1202513494	13005	1202522680	8332	1202531984	8450	1202540725	7874	1202550028	8449
59	1202513494	13005	1202522680	8332	1202532002	8468	1202540725	7874	1202550048	8469
60	1202513759	13270	1202522946	8598	1202532244	8710	1202540990	8139	1202550291	8712
61	1202513784	13295	1202522971	8623	1202532299	8765	1202541015	8164	1202550343	8764
62	1202514035	13546	1202523221	8873	1202532498	8964	1202541266	8415	1202550545	8966
63	1202514061	13572	1202523248	8900	1202532600	9066	1202541293	8442	1202550644	9065
64	1202514292	13803	1202523479	9131	1202532787	9253	1202541524	8673	1202550833	9254

1202383349 0	Condor Pov	ray 4 Machines									
1202383402 53	Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A
1202383420 71	1	1202383349	0	1202388375	0	1202393398	0	1202398423	0	1202403445	0
1202383425	2	1202383402	53	1202388426	51	1202393449	51	1202398472	49	1202403495	50
1202383577 228 1202388603 228 1202393626 228 120239651 228 1202403673 228 1202383678 329 1202388701 326 1202393726 328 1202398748 325 1202403771 326 3202383701 352 1202388726 351 1202393750 352 1202398771 348 1202403795 350 1202383714 365 1202388739 364 1202393762 364 1202398783 360 1202403807 362	3	1202383420	71	1202388445	70	1202393467	69	1202398489	66	1202403513	68
1202383678 329	4	1202383425	76	1202388449	74	1202393472	74	1202398494	71	1202403518	73
1202383701 352	5	1202383577	228	1202388603	228	1202393626	228	1202398651	228	1202403673	228
1202383714 365 1202388739 364 1202393762 364 1202398783 360 1202403807 362 1202383854 505 1202388880 505 1202393903 505 1202398928 505 1202403949 504 1	6	1202383678	329	1202388701	326	1202393726	328	1202398748	325	1202403771	326
1202383854 505 1202388880 505 1202393903 505 1202398928 505 1202403949 504	7	1202383701	352	1202388726	351	1202393750	352	1202398771	348	1202403795	350
1202383966 617	8	1202383714	365	1202388739	364	1202393762	364	1202398783	360	1202403807	362
1 1202384039 690 1202389065 690 1202394087 689 1202399109 686 1202404081 636 2 1202384156 807 1202389181 806 1202394204 806 1202399229 806 1202404251 806 4 1202384344 995 1202389369 994 1202394393 995 1202399414 991 1202404437 992 5 1202384374 1025 1202389399 1024 1202394423 1025 1202399443 1020 1202404468 1023 6 1202384387 1038 1202389413 1038 1202394425 1037 1202399457 1034 1202404468 1035 7 1202384468 1119 1202389494 1119 1202394518 1120 1202399542 1119 1202404564 1119 8 1202384673 1324 1202389697 1322 1202394722 1324 1202399743 1320 1202404766 1321 9 1202384681 1332 1202389705 1330 1202394730 1332 1202399750 1327 1202404766 1321 1202384702 1353 1202389728 1353 1202394760 1362 1202399785 1362 1202404706 1351 1 1202384940 1591 1202389966 1591 120239490 1592 1202400016 1593 1202404504 1592 1 1202384982 1633 120238973 1598 120239498 1600 1202400016 1593 1202405077 1632 1 120238498 1599 1202389973 1598 120239498 1600 1202400016 1593 1202405077 1632 1 1202384982 1633 1202390006 1631 1202395064 1666 1202400017 1664 1202405109 1664 1 1202385160 1811 1202390186 1811 1202395274 1876 1202400294 1871 1202405317 1872 8 120238529 1980 120239055 1980 120239578 1980 120240099 1975 1202405424 1979	9	1202383854	505	1202388880	505	1202393903	505	1202398928	505	1202403949	504
1202384039 690	10	1202383966	617	1202388991	616	1202394016	618	1202399037	614	1202404060	615
1202384156 807 1202389181 806 1202394204 806 1202399229 806 1202404251 806 1202384344 995 1202389369 994 1202394393 995 1202399414 991 1202404437 992 1202384374 1025 1202389399 1024 1202394423 1025 1202399443 1020 1202404468 1023 1202384387 1038 1202389413 1038 1202394435 1037 1202399457 1034 1202404480 1035 1202384468 1119 1202389494 1119 1202394518 1120 120239942 1119 1202404664 1119 8	11	1202383987	638	1202389012	637	1202394036	638	1202399057	634	1202404081	636
4 1202384344 995 1202389369 994 1202394393 995 1202399414 991 1202404437 992 5 1202384374 1025 1202389399 1024 1202394423 1025 1202399443 1020 1202404468 1023 6 1202384387 1038 1202389413 1038 1202394435 1037 1202399457 1034 1202404480 1035 7 1202384468 1119 1202389494 1119 1202394518 1120 1202399542 1119 1202404564 1119 8 1202384673 1324 1202389697 1322 1202394722 1324 1202399743 1320 1202404766 1321 9 1202384681 1332 1202389705 1330 1202394750 1332 1202399750 1327 1202404776 1331 1 1202384702 1353 120238978 1353 1202394750 1362 1202399785 1362 1202404076 1351 1	12	1202384039	690	1202389065	690	1202394087	689	1202399109	686	1202404132	687
1202384374 1025 1202389399 1024 1202394423 1025 1202399443 1020 1202404468 1023 1202394437 1038 1202389413 1038 1202394435 1037 1202399457 1034 1202404480 1035 1202384468 1119 1202389494 1119 1202394518 1120 1202399542 1119 1202404564 1119 1202394518 1120 1202399542 1119 1202404564 1119 1202394518 1120 1202394518 1120 1202399743 1320 1202404766 1321 1202384673 1324 1202389697 1322 1202394722 1324 1202399743 1320 1202404766 1321 1202384681 1332 1202389705 1330 1202394730 1332 1202399750 1327 1202404776 1331 1202384702 1353 1202389728 1353 1202394751 1353 1202399772 1349 1202404796 1351 1202384710 1361 1202389736 1361 1202394760 1362 1202399785 1362 1202404007 1362 1202384940 1591 1202389966 1591 1202394990 1592 1202400016 1593 1202405037 1592 1202405044 1597 1202389978 1598 1202399998 1600 1202400018 1595 1202405042 1597 1202384982 1633 120239006 1631 1202395031 1633 1202400051 1628 1202405077 1632 1202385015 1666 1202390041 1666 120239504 1666 120239504 1664 1202400087 1664 1202405109 1664 1202385160 1811 1202390186 1811 1202395210 1812 1202400294 1871 1202405317 1872 1202385329 1980 1202390355 1980 1202395378 1980 1202400398 1975 1202405424 1979	13	1202384156	807	1202389181	806	1202394204	806	1202399229	806	1202404251	806
1202384387 1038 1202389413 1038 1202394435 1037 1202399457 1034 1202404480 1035 1202384468 1119 1202389494 1119 1202394518 1120 120239542 1119 1202404564 1119 18 1202384673 1324 1202389697 1322 1202394722 1324 1202399743 1320 1202404766 1321 1202384681 1332 1202389705 1330 1202394730 1332 1202399750 1327 1202404776 1331 1202384702 1353 1202389728 1353 1202399751 1353 1202399772 1349 1202404796 1351 1 1202384710 1361 1202389736 1361 1202394760 1362 1202399785 1362 1202404807 1362 1 1202384940 1591 1202389966 1591 1202394990 1592 1202400016 1593 1202405037 1592 1 1202384948 1599 1202389973 1598 1202394998 1600 1202400018 1595 1202405042 1597 1 1202384982 1633 1202390006 1631 1202394998 1600 1202400018 1595 1202405042 1597 1 1202385015 1666 1202390041 1666 1202395031 1633 1202400051 1628 1202405077 1632 1 1202385160 1811 1202390186 1811 1202395274 1876 1202400294 1871 1202405317 1872 1 1202385329 1980 1202390355 1980 1202395378 1980 120240038 1975 1202405424 1979	14	1202384344	995	1202389369	994	1202394393	995	1202399414	991	1202404437	992
1202384468 1119 1202389494 1119 1202394518 1120 1202399542 1119 1202404564 1119 8 1202384673 1324 1202389697 1322 1202394722 1324 1202399743 1320 1202404766 1321 9 1202384681 1332 1202389705 1330 1202394730 1332 1202399750 1327 1202404776 1331 1 1202384702 1353 1202389728 1353 1202394751 1353 1202399772 1349 1202404796 1351 1 1202384710 1361 1202389736 1361 1202394760 1362 1202399785 1362 120240407 1362 2 1202384940 1591 1202389966 1591 1202394990 1592 1202400016 1593 1202405037 1592 3 1202384948 1599 1202389973 1598 1202394998 1600 1202400018 1595 1202405042 1597 4 1202384982 1633 1202390006 1631 1202395031 1633 1202400051 1628 1202405077 1632 5 1202385015 1666 1202390041 1666 1202395064 1666 1202400087 1664 1202405109 1664 6 1202385160 1811 1202390186 1811 1202395274 1876 1202400294 1871 1202405317 1872 8 1202385329 1980 1202390355 1980 1202395378 1980 120240038 1975 1202405424 1979	15	1202384374	1025	1202389399	1024	1202394423	1025	1202399443	1020	1202404468	1023
1202384673 1324 1202389697 1322 1202394722 1324 1202399743 1320 1202404766 1321	16	1202384387	1038	1202389413	1038	1202394435	1037	1202399457	1034	1202404480	1035
1202384681 1332 1202389705 1330 1202394730 1332 1202399750 1327 1202404776 1331 1202384702 1353 1202389728 1353 1202394751 1353 1202399772 1349 1202404796 1351 1202384710 1361 1202389736 1361 1202394760 1362 1202399785 1362 1202404807 1362 1202384940 1591 1202389966 1591 1202394990 1592 1202400016 1593 1202405037 1592 1202384948 1599 1202389973 1598 1202394998 1600 1202400018 1595 1202405042 1597 1202384982 1633 1202390006 1631 1202395031 1633 1202400051 1628 1202405077 1632 1202385015 1666 1202390041 1666 1202395064 1666 1202400087 1664 1202405109 1664 1202385160 1811 120239186 1811 1202395210 1812 1202400236 1813 1202405317 1872 1202385329 1980 1202390355 1980 1202395378 1980 120240038 1975 1202405424 1979	17	1202384468	1119	1202389494	1119	1202394518	1120	1202399542	1119	1202404564	1119
1202384702 1353 1202389728 1353 1202394751 1353 1202399772 1349 1202404796 1351 1202384710 1361 1202389736 1361 1202394760 1362 1202399785 1362 1202404807 1362 1202384940 1591 1202389966 1591 1202394990 1592 1202400016 1593 1202405037 1592 1202384948 1599 1202389973 1598 1202394998 1600 1202400018 1595 1202405042 1597 1202384982 1633 1202390006 1631 1202395031 1633 1202400051 1628 1202405077 1632 1202385015 1666 1202390041 1666 1202395064 1666 1202400087 1664 1202405109 1664 1202385160 1811 120239186 1811 1202395210 1812 1202400236 1813 1202405257 1812 1202385224 1875 1202390248 1873 1202395274 1876 1202400294 1871 1202405317 1872 1202385329 1980 1202390355 1980 1202395378 1980 120240038 1975 1202405424 1979	18	1202384673	1324	1202389697	1322	1202394722	1324	1202399743	1320	1202404766	1321
1 1202384710 1361 1202389736 1361 1202394760 1362 1202399785 1362 1202404807 1362 2 1202384940 1591 1202389966 1591 1202394990 1592 1202400016 1593 1202405037 1592 3 1202384948 1599 1202389973 1598 1202394998 1600 1202400018 1595 1202405042 1597 4 1202384982 1633 1202390006 1631 1202395031 1633 1202400051 1628 1202405077 1632 5 1202385015 1666 1202390041 1666 1202395064 1666 1202400087 1664 1202405109 1664 6 1202385160 1811 1202390186 1811 1202395210 1812 1202400236 1813 1202405257 1812 7 1202385224 1875 1202390248 1873 1202395274 1876 1202400294 1871 1202405317 1872 8	19	1202384681	1332	1202389705	1330	1202394730	1332	1202399750	1327	1202404776	1331
2 1202384940 1591 1202389966 1591 1202394990 1592 1202400016 1593 1202405037 1592 3 1202384948 1599 1202389973 1598 1202394998 1600 1202400018 1595 1202405042 1597 4 1202384982 1633 1202390006 1631 1202395031 1633 1202400051 1628 1202405077 1632 5 1202385015 1666 1202390041 1666 1202395064 1666 1202400087 1664 1202405109 1664 6 1202385160 1811 1202390186 1811 1202395210 1812 1202400236 1813 1202405257 1812 7 1202385224 1875 1202390248 1873 1202395274 1876 1202400294 1871 1202405317 1872 8 1202385329 1980 1202390355 1980 1202395378 1980 1202400398 1975 1202405424 1979	20	1202384702	1353	1202389728	1353	1202394751	1353	1202399772	1349	1202404796	1351
1202384948 1599 1202389973 1598 1202394998 1600 1202400018 1595 1202405042 1597 1202384982 1633 1202390006 1631 1202395031 1633 1202400051 1628 1202405077 1632 1202385015 1666 1202390041 1666 1202395064 1666 1202400087 1664 1202405109 1664 1202385160 1811 1202390186 1811 1202395210 1812 1202400236 1813 1202405257 1812 1202385224 1875 1202390248 1873 1202395274 1876 1202400294 1871 1202405317 1872 1202385329 1980 1202390355 1980 1202395378 1980 1202400398 1975 1202405424 1979	21	1202384710	1361	1202389736	1361	1202394760	1362	1202399785	1362	1202404807	1362
4 1202384982 1633 1202390006 1631 1202395031 1633 1202400051 1628 1202405077 1632 5 1202385015 1666 1202390041 1666 1202395064 1666 1202400087 1664 1202405109 1664 6 1202385160 1811 1202390186 1811 1202395210 1812 1202400236 1813 1202405257 1812 7 1202385224 1875 1202390248 1873 1202395274 1876 1202400294 1871 1202405317 1872 8 1202385329 1980 1202390355 1980 1202395378 1980 1202400398 1975 1202405424 1979	22	1202384940	1591	1202389966	1591	1202394990	1592	1202400016	1593	1202405037	1592
5 1202385015 1666 1202390041 1666 1202395064 1666 1202400087 1664 1202405109 1664 6 1202385160 1811 1202390186 1811 1202395210 1812 1202400236 1813 1202405257 1812 7 1202385224 1875 1202390248 1873 1202395274 1876 1202400294 1871 1202405317 1872 8 1202385329 1980 1202390355 1980 1202395378 1980 1202400398 1975 1202405424 1979	23	1202384948	1599	1202389973	1598	1202394998	1600	1202400018	1595	1202405042	1597
1202385160 1811 1202390186 1811 1202395210 1812 1202400236 1813 1202405257 1812 7 1202385224 1875 1202390248 1873 1202395274 1876 1202400294 1871 1202405317 1872 8 1202385329 1980 1202390355 1980 1202395378 1980 1202400398 1975 1202405424 1979	24	1202384982	1633	1202390006	1631	1202395031	1633	1202400051	1628	1202405077	1632
7 1202385224 1875 1202390248 1873 1202395274 1876 1202400294 1871 1202405317 1872 8 1202385329 1980 1202390355 1980 1202395378 1980 1202400398 1975 1202405424 1979	25	1202385015	1666	1202390041	1666	1202395064	1666	1202400087	1664	1202405109	1664
8 1202385329 1980 1202390355 1980 1202395378 1980 1202400398 1975 1202405424 1979	26	1202385160	1811	1202390186	1811	1202395210	1812	1202400236	1813	1202405257	1812
	27	1202385224	1875	1202390248	1873	1202395274	1876	1202400294	1871	1202405317	1872
9 1202385348 1999 1202390373 1998 1202395397 1999 1202400424 2001 1202405442 1997	28	1202385329	1980	1202390355	1980	1202395378	1980	1202400398	1975	1202405424	1979
	29	1202385348	1999	1202390373	1998	1202395397	1999	1202400424	2001	1202405442	1997

30	1202385375	2026	1202390401	2026	1202395425	2027	1202400450	2027	1202405472	2027
31	1202385447	2098	1202390471	2096	1202395497	2099	1202400517	2094	1202405472	2096
32	1202385546	2197	1202390572	2197	1202395595	2197	1202400615	2192	1202405641	2196
33	1202385547	2198	1202390572	2198	1202395598	2200	1202400623	2200	1202405644	2199
34	1202385558	2209	1202390583	2208	1202395607	2209	1202400623	2211	1202405653	2208
35	1202385627	2278	1202390651	2276	1202395678	2280	1202400697	2274	1202405721	2276
36	1202385723	2374	1202390749	2374	1202395774	2376	1202400077	2376	1202405721	2375
37	1202385741	2392	1202390747	2392	1202395789	2391	1202400755	2387	1202405835	2390
38	1202385752	2403	1202390777	2402	1202395800	2402	1202400828	2405	1202405846	2401
39	1202385845	2496	1202390869	2494	1202395895	2497	1202400915	2492	1202405939	2494
40	1202385948	2599	1202390974	2599	1202396000	2602	1202400913	2601	1202406045	2600
41	1202386001	2652	1202391026	2651	1202396049	2651	1202401024	2646	1202406094	2649
42	1202386039	2690	1202391020	2687	1202396086	2688	1202401003	2690	1202406034	2687
43	1202386156	2807	1202391181	2806	1202396206	2808	1202401227	2804	1202406251	2806
44	1202386232	2883	1202391181	2883	1202396284	2886	1202401227	2885	1202406331	2884
45	1202386315	2966	1202391341	2966	1202396364	2966	1202401308	2961	1202406329	2964
46	1202386352	3003	1202391374	2999	1202396398	3000	1202401304	3002	1202406444	2999
47	1202386474	3125	1202391498	3123	1202396524	3126	1202401544	3121	1202406568	3123
48	1202386512	3163	1202391538	3163	1202396564	3166	1202401544	3165	1202406608	3163
49	1202386631	3282	1202391657	3282	1202396681	3283	1202401699	3276	1202406724	3279
50	1202386673	3324	1202391696	3321	1202396719	3321	1202401746	3323	1202406767	3322
51	1202386782	3433	1202391807	3432	1202396832	3434	1202401853	3430	1202406877	3432
52	1202386818	3469	1202391845	3470	1202396871	3473	1202401895	3472	1202406915	3470
53	1202386994	3645	1202392017	3642	1202397040	3642	1202402063	3640	1202407088	3643
54	1202386994	3645	1202392021	3646	1202397044	3646	1202402068	3645	1202407088	3643
55	1202387048	3699	1202392073	3698	1202397098	3700	1202402119	3696	1202407142	3697
56	1202387055	3706	1202392082	3707	1202397107	3709	1202402132	3709	1202407151	3706
57	1202387245	3896	1202392270	3895	1202397294	3896	1202402318	3895	1202407341	3896
58	1202387251	3902	1202392273	3898	1202397297	3899	1202402319	3896	1202407345	3900
59	1202387306	3957	1202392332	3957	1202397356	3958	1202402377	3954	1202407400	3955
60	1202387331	3982	1202392358	3983	1202397384	3986	1202402409	3986	1202407428	3983
61	1202387508	4159	1202392530	4155	1202397554	4156	1202402577	4154	1202407602	4157
62	1202387524	4175	1202392549	4174	1202397574	4176	1202402598	4175	1202407620	4175
63	1202387546	4197	1202392572	4197	1202397599	4201	1202402621	4198	1202407642	4197
64	1202387549	4200	1202392575	4200	1202397600	4202	1202402623	4200	1202407644	4199
	1.202307077	1	1-02372075	00	1 0237,030	02	1 02 .02023	00	1202.07074	1//

Condor Pov	ray 8 Machines									
Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A
1	1202334669	0	1202337529	0	1202374040	0	1202376378	0	1202379328	0
2	1202334682	13	1202337562	33	1202374077	37	1202376393	15	1202379361	33
3	1202334710	41	1202337576	47	1202374092	52	1202376425	47	1202379375	47
4	1202334720	51	1202337580	51	1202374096	56	1202376430	52	1202379379	51
5	1202334724	55	1202337587	58	1202374102	62	1202376434	56	1202379385	57
6	1202334762	93	1202337625	96	1202374142	102	1202376468	90	1202379426	98
7	1202334768	99	1202337631	102	1202374146	106	1202376475	97	1202379427	99
8	1202334782	113	1202337646	117	1202374150	110	1202376495	117	1202379447	119
9	1202334969	300	1202337830	301	1202374340	300	1202376678	300	1202379628	300
10	1202334971	302	1202337840	311	1202374356	316	1202376683	305	1202379639	311
11	1202334974	305	1202337851	322	1202374367	327	1202376689	311	1202379650	322
12	1202335045	376	1202337905	376	1202374421	381	1202376756	378	1202379705	377
13	1202335052	383	1202337915	386	1202374429	389	1202376762	384	1202379712	384
14	1202335130	461	1202337994	465	1202374499	459	1202376843	465	1202379795	467
15	1202335139	470	1202338002	473	1202374518	478	1202376844	466	1202379802	474
16	1202335154	485	1202338018	489	1202374533	493	1202376862	484	1202379815	487
17	1202335258	589	1202338143	614	1202374659	619	1202376973	595	1202379943	615
18	1202335302	633	1202338159	630	1202374675	635	1202377014	636	1202379958	630
19	1202335311	642	1202338170	641	1202374680	640	1202377018	640	1202379968	640
20	1202335315	646	1202338178	649	1202374692	652	1202377025	647	1202379975	647
21	1202335360	691	1202338221	692	1202374736	696	1202377071	693	1202380020	692
22	1202335406	737	1202338270	741	1202374775	735	1202377118	740	1202380071	743

23	1202335441	772	1202338305	776	1202374820	780	1202377146	768	1202380104	776
24	1202335467	798	1202338331	802	1202374846	806	1202377176	798	1202380129	801
25	1202335488	819	1202338373	844	1202374890	850	1202377203	825	1202380173	845
26	1202335160	872	1202338398	869	1202374913	873	1202377253	875	1202380197	869
27	1202335511	918	1202338446	917	1202374956	916	1202377293	915	1202380243	915
28	1202335639	970	1202338503	974	1202375008	968	1202377253	973	1202380304	976
29	1202335663	994	1202338526	997	1202375040	1000	1202377370	992	1202380323	995
30	1202335664	995	1202338528	999	1202375043	1003	1202377373	995	1202380327	999
31	1202335682	1013	1202338548	1019	1202375063	1023	1202377393	1015	1202380347	1019
32	1202335685	1016	1202338553	1024	1202375069	1029	1202377397	1019	1202380357	1024
33	1202335693	1024	1202338568	1039	1202375083	1043	1202377403	1025	1202380352	1039
34	1202335728	1059	1202338585	1056	1202375101	1061	1202377403	1062	1202380384	1056
35	1202335728	1099	1202338626	1097	1202375101	1098	1202377474	1096	1202380424	1096
36	1202335766	1165	1202338698	1169	1202375130	1163	1202377546	1168	1202380424	1171
37	1202335854	1185	1202338716	1187	1202375230	1190	1202377560	1182	1202380513	1185
38	1202335854	1190	1202338710	1194	1202375230	1197	1202377567	1189	1202380513	1192
39	1202335837	1214	1202338766	1237	1202375237	1241	1202377507	1233	1202380520	1237
40	1202335928	1259	1202338797	1268	1202375281	1273	1202377611	1244	1202380597	1269
41	1202335952	1283	1202338807	1278	1202375313	1282	1202377622	1285	1202380606	1278
42	1202336014	1345	1202338870	1341	1202375322	1347	1202377003	1347	1202380670	1342
43	1202336079	1410	1202338870	1409	1202375387	1409	1202377725	1407	1202380070	1408
				1478				1477		1479
44	1202336143	1474	1202339007	1503	1202375511	1471	1202377855	1477	1202380807	1501
45	1202336170		1202339032		1202375546	1506	1202377876		1202380829	
46	1202336171	1502	1202339035	1506	1202375549	1509	1202377879	1501	1202380832	1504
47	1202336176	1507	1202339083	1554	1202375597	1557	1202377902	1524	1202380883	1555
48	1202336232	1563	1202339097	1568	1202375613	1573	1202377929	1551	1202380897	1569
49	1202336268	1599	1202339101	1572	1202375617	1577	1202377979	1601	1202380901	1573
50	1202336336	1667	1202339192	1663	1202375708	1668	1202378047	1669	1202380991	1663
51	1202336387	1718	1202339247	1718	1202375757	1717	1202378094	1716	1202381044	1716
52	1202336421	1752	1202339334	1805	1202375845	1805	1202378146	1768	1202381134	1806
53	1202336476	1807	1202339341	1812	1202375850	1810	1202378186	1808	1202381141	1813
54	1202336489	1820	1202339347	1818	1202375863	1823	1202378188	1810	1202381148	1820
55	1202336493	1824	1202339356	1827	1202375871	1831	1202378204	1826	1202381152	1824
56	1202336525	1856	1202339357	1828	1202375873	1833	1202378236	1858	1202381159	1831
57	1202336535	1866	1202339396	1867	1202375911	1871	1202378240	1862	1202381194	1866
58	1202336588	1919	1202339442	1913	1202375959	1919	1202378297	1919	1202381241	1913
59	1202336645	1976	1202339506	1977	1202376015	1975	1202378352	1974	1202381303	1975
60	1202336698	2029	1202339591	2062	1202376106	2066	1202378424	2046	1202381391	2063
61	1202336737	2068	1202339600	2071	1202376107	2067	1202378446	2068	1202381395	2067
62	1202336746	2077	1202339604	2075	1202376114	2074	1202378448	2070	1202381406	2078
63	1202336756	2087	1202339611	2082	1202376120	2080	1202378466	2088	1202381410	2082
64	1202336759	2090	1202339620	2091	1202376145	2105	1202378469	2091	1202381421	2093
Condor Povrav	16 Machines									

Condor Povi	ay 16 Machines									
Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A
1	1202361657	0	1202363004	0	1202364830	0	1202366656	0	1202368483	0
2	1202361692	35	1202363019	15	1202364864	34	1202366690	34	1202368516	33
3	1202361711	54	1202363057	53	1202364884	54	1202366710	54	1202368535	52
4	1202361717	60	1202363060	56	1202364889	59	1202366714	58	1202368540	57
5	1202361728	71	1202363070	66	1202364898	68	1202366724	68	1202368551	68
6	1202361755	98	1202363099	95	1202364927	97	1202366753	97	1202368582	99
7	1202361758	101	1202363100	96	1202364928	98	1202366757	101	1202368583	100
8	1202361763	106	1202363106	102	1202364934	104	1202366761	105	1202368588	105
9	1202361778	121	1202363121	117	1202364949	119	1202366776	120	1202368603	120
10	1202361779	122	1202363121	117	1202364949	119	1202366776	120	1202368604	121
11	1202361789	132	1202363130	126	1202364959	129	1202366789	133	1202368615	132
12	1202361797	140	1202363141	137	1202364970	140	1202366798	142	1202368623	140
13	1202361801	144	1202363162	158	1202364990	160	1202366818	162	1202368629	146
14	1202361848	191	1202363192	188	1202365022	192	1202366825	169	1202368673	190
15	1202361872	215	1202363217	213	1202365046	216	1202366873	217	1202368698	215

16	1202361887	230	1202363232	228	1202365060	230	1202366888	232	1202368713	230
				331						333
17	1202361993	336	1202363335		1202365162	332	1202366988	332	1202368816	340
18	1202361996	339	1202363343	339	1202365171		1202366996	340	1202368823	
19	1202362010	353	1202363349	345	1202365181	351	1202367007	351	1202368845	362
20	1202362019	362	1202363353	349	1202365190	360	1202367017	361	1202368845	362
21	1202362021	364	1202363362	358	1202365193	363	1202367019	363	1202368846	363
22	1202362021	364	1202363366	362	1202365194	364	1202367020	364	1202368853	370
23	1202362028	371	1202363372	368	1202365200	370	1202367026	370	1202368856	373
24	1202362033	376	1202363376	372	1202365204	374	1202367029	373	1202368858	375
25	1202362060	403	1202363403	399	1202365231	401	1202367058	402	1202368862	379
26	1202362066	409	1202363407	403	1202365236	406	1202367060	404	1202368892	409
27	1202362077	420	1202363420	416	1202365249	419	1202367065	409	1202368902	419
28	1202362081	424	1202363425	421	1202365254	424	1202367076	420	1202368907	424
29	1202362095	438	1202363440	436	1202365270	440	1202367097	441	1202368922	439
30	1202362105	448	1202363450	446	1202365278	448	1202367105	449	1202368930	447
31	1202362119	462	1202363464	460	1202365292	462	1202367120	464	1202368944	461
32	1202362136	479	1202363495	491	1202365323	493	1202367151	495	1202368962	479
33	1202362176	519	1202363530	526	1202365348	518	1202367174	518	1202369010	527
34	1202362184	527	1202363530	526	1202365358	528	1202367183	527	1202369025	542
35	1202362204	547	1202363533	529	1202365372	542	1202367198	542	1202369026	543
36	1202362212	555	1202363545	541	1202365384	554	1202367210	554	1202369040	557
37	1202362214	557	1202363553	549	1202365385	555	1202367212	556	1202369042	559
38	1202362216	559	1202363559	555	1202365390	560	1202367215	559	1202369047	564
39	1202362248	591	1202363589	585	1202365418	588	1202367244	588	1202369074	591
40	1202362277	620	1202363620	616	1202365448	618	1202367274	618	1202369101	618
41	1202362320	663	1202363663	659	1202365491	661	1202367316	660	1202369102	619
42	1202362352	695	1202363692	688	1202365522	692	1202367345	689	1202369177	694
43	1202362388	731	1202363731	727	1202365561	731	1202367377	721	1202369213	730
44	1202362389	732	1202363733	729	1202365563	733	1202367385	729	1202369216	733
45	1202362412	755	1202363756	752	1202365585	755	1202367413	757	1202369238	755
46	1202362413	756	1202363757	753	1202365586	756	1202367413	757	1202369238	755
47	1202362417	760	1202363763	759	1202365591	761	1202367418	762	1202369243	760
48	1202362442	785	1202363769	795	1202365627	797	1202367455	799	1202369267	784
49	1202362442	810	1202363820	816	1202365640	810	1202367466	810	1202369326	843
		849				851		849		
50	1202362506		1202363847	843	1202365681		1202367505		1202369336	853
51	1202362513	856	1202363855	851	1202365681	851	1202367507	851	1202369338	855
52	1202362514	857	1202363855	851	1202365684	854	1202367511	855	1202369340	857
53	1202362534	877	1202363877	873	1202365705	875	1202367531	875	1202369347	864
54	1202362537	880	1202363879	875	1202365711	881	1202367537	881	1202369359	876
55	1202362546	889	1202363881	877	1202365717	887	1202367544	888	1202369370	887
56	1202362577	920	1202363919	915	1202365748	918	1202367572	916	1202369374	891
57	1202362580	923	1202363920	916	1202365751	921	1202367577	921	1202369409	926
58	1202362603	946	1202363943	939	1202365773	943	1202367596	940	1202369430	947
59	1202362646	989	1202363989	985	1202365820	990	1202367636	980	1202369472	989
60	1202362652	995	1202363997	993	1202365834	1004	1202367654	998	1202369478	995
61	1202362661	1004	1202364007	1003	1202365843	1013	1202367662	1006	1202369487	1004
62	1202362676	1019	1202364032	1028	1202365845	1015	1202367686	1030	1202369500	1017
63	1202362690	1033	1202364034	1030	1202365860	1030	1202367689	1033	1202369516	1033
64	1202362692	1035	1202364037	1033	1202365863	1033	1202367694	1038	1202369519	1036

BOINC

BOINC Povray	BOINC Povray 1 Machine												
Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A			
1	1202558161	0	1202637425	0	1202712680	0	1202731457	0	1202768076	0			
2	1202558418	257	1202637672	247	1202713053	373	1202731708	251	1202768382	306			
3	1202558685	524	1202637950	525	1202713321	641	1202731927	470	1202768585	509			
4	1202558915	754	1202638269	844	1202713640	960	1202732136	679	1202768799	723			

5	1202550116	055	1202/28570	1145	1202712070	1100	1202722254	897	1202760020	953
5	1202559116	955 1252	1202638570	1145	1202713870	1190	1202732354		1202769029	1144
7	1202559413	1466	1202638860	1649	1202714105	1425 1670	1202732621	1164	1202769220	1324
8	1202559813	1652	1202639074	1973	1202714530	1939	1202732831	1630	1202769400	1643
9	1202559815	2024	1202639398	2284	1202714819	2208	1202733087	1911	1202769913	1877
10	1202560415	2254	1202640044	2619	1202714888	2427	1202733368	2185	1202709933	2112
11	1202560650	2489	1202640261	2836	1202715391	2711	1202733042	2486	1202770188	2408
12	1202560891	2730	1202640470	3045	1202715659	2979	1202734234	2777	1202770730	2654
13	1202561160	2999	1202640761	3336	1202715037	3292	1202734234	3022	1202770730	2938
14	1202561432	3271	1202640761	3570	1202716262	3582	1202734479	3301	1202771014	3240
15	1202561740	3579	1202641225	3800	1202716558	3878	1202735054	3597	1202771516	3480
16	1202562030	3869	1202641466	4041	1202716837	4157	1202735034	3782	1202771824	3748
17	1202562380	4219	1202641745	4320	1202717032	4352	1202735557	4100	1202772097	4021
18	1202562594	4433	1202641936	4511	1202717313	4633	1202735870	4413	1202772338	4262
19	1202562928	4767	1202642258	4833	1202717615	4935	1202736187	4730	1202772655	4579
20	1202563262	5101	1202642505	5080	1202717828	5148	1202736559	5102	1202772947	4871
21	1202563590	5429	1202642691	5266	1202718145	5465	1202736932	5475	1202773159	5083
22	1202563870	5709	1202642959	5534	1202718413	5733	1202737237	5780	1202773439	5363
23	1202564104	5943	1202643243	5818	1202718786	6106	1202737528	6071	1202773762	5686
24	1202564356	6195	1202643512	6087	1202719027	6347	1202737829	6372	1202774040	5964
25	1202564563	6402	1202643809	6384	1202719377	6697	1202738164	6707	1202774269	6193
26	1202564885	6724	1202644088	6663	1202719673	6993	1202738399	6942	1202774544	6468
27	1202565161	7000	1202644268	6843	1202719924	7244	1202738728	7271	1202774778	6702
28	1202565478	7317	1202644586	7161	1202720236	7556	1202739034	7577	1202775092	7016
29	1202565790	7629	1202644838	7413	1202720483	7803	1202739330	7873	1202775405	7329
30	1202566104	7943	1202645150	7725	1202720816	8136	1202739537	8080	1202775776	7700
31	1202566420	8259	1202645342	7917	1202721144	8464	1202739838	8381	1202775991	7915
32	1202566793	8632	1202645614	8189	1202721441	8761	1202740139	8682	1202776231	8155
33	1202567116	8955	1202645805	8380	1202721692	9012	1202740390	8933	1202776422	8346
34	1202567411	9250	1202646074	8649	1202721993	9313	1202740571	9114	1202776795	8719
35	1202567713	9552	1202646446	9021	1202722246	9566	1202740882	9425	1202777063	8987
36	1202567965	9804	1202646715	9290	1202722481	9801	1202741151	9694	1202777331	9255
37	1202568200	10039	1202647088	9663	1202722760	10080	1202741447	9990	1202777666	9590
38	1202568430	10269	1202647421	9996	1202722995	10315	1202741699	10242	1202777872	9796
39	1202568732	10571	1202647650	10225	1202723329	10649	1202741950	10493	1202778124	10048
40	1202568912	10751	1202647971	10546	1202723587	10907	1202742252	10795	1202778452	10376
41	1202569209	11048	1202648274	10849	1202723816	11136	1202742498	11041	1202778753	10677
42	1202569455	11294	1202648531	11106	1202724139	11459	1202742818	11361	1202779049	10973
43	1202569762	11601	1202648826	11401	1202724368	11688	1202743086	11629	1202779284	11208
44	1202570007	11846	1202649061	11636	1202724588	11908	1202743420	11963	1202779586	11510
45	1202570226	12065	1202649280	11855	1202724795	12115	1202743709	12252	1202779843	11767
46	1202570493	12332	1202649587	12162	1202724981	12301	1202744031	12574	1202780144	12068
47	1202570707	12546	1202649801	12376	1202725160	12480	1202744246	12789	1202780478	12402
48	1202570948	12787	1202650041	12616	1202725352	12672	1202744477	13020	1202780697	12621
49	1202571139	12978	1202650294	12869	1202725542	12862	1202744672	13215	1202781003	12927
50	1202571331	13170	1202650595	13170	1202725844	13164	1202744941	13484	1202781261	13185
51	1202571583	13422	1202650891	13466	1202726140	13460	1202745132	13675	1202781446	13370
52	1202571774	13613	1202651198	13773	1202726445	13765	1202745433	13976	1202781637	13561
53	1202572026	13865	1202651493	14068	1202726746	14066	1202745625	14168	1202781889	13813
54	1202572305	14144	1202651740	14315	1202727052	14372	1202745853	14396	1202782169	14093
55	1202572606	14445	1202651997	14572	1202727343	14663	1202746067	14610	1202782464	14388
56	1202572906	14745	1202652299	14874	1202727648	14968	1202746308	14851	1202782770	14694
57	1202573214	15053	1202652599	15174	1202727955	15275	1202746559	15102	1202783062	14986
58	1202573465	15304	1202652950	15525	1202728278	15598	1202746839	15382	1202783363	15287
59	1202573766	15605	1202653262	15837	1202728525	15845	1202747145	15688	1202783659	15583
60	1202574073	15912	1202653513	16088	1202728776	16096	1202747496	16039	1202783983	15907
61	1202574368	16207	1202653766	16341	1202729022	16342	1202747752	16295	1202784337	16261
				16576	1202720274	16504	1202748004	16547	1202704501	16515
62	1202574658	16497	1202654001	16576	1202729274	16594	1202/48004	16547	1202784591	16515

64	1202574959	16798	1202654296	16871	1202729629	16949	1202748305	16848	1202784907	16831	

BOINC Povrav	y 2 Machines									
Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A
1	1203497198	0	1203553577	0	1203564842	0	1203577570	0	1203592328	0
2	1203497203	5	1203553582	5	1203565042	200	1203577659	89	1203592532	204
3	1203497432	234	1203553811	234	1203565060	218	1203577788	218	1203592608	280
4	1203497530	332	1203553839	262	1203565293	451	1203577907	337	1203592787	459
5	1203497688	490	1203554053	476	1203565295	453	1203578026	456	1203592854	526
6	1203497831	633	1203554128	551	1203565598	756	1203578182	612	1203593074	746
7	1203497930	732	1203554338	761	1203565639	797	1203578265	695	1203593146	818
8	1203498130	932	1203554428	851	1203565895	1053	1203578534	964	1203593314	986
9	1203498244	1046	1203554688	1111	1203565950	1108	1203578559	989	1203593507	1179
10	1203498360	1162	1203554722	1145	1203566196	1354	1203578818	1248	1203593638	1310
11	1203498512	1314	1203555017	1440	1203566201	1359	1203578849	1279	1203593776	1448
12	1203498683	1485	1203555044	1467	1203566495	1653	1203579011	1441	1203593949	1621
13	1203498797	1599	1203555296	1719	1203566544	1702	1203579069	1499	1203594045	1717
14	1203498939	1741	1203555384	1807	1203566737	1895	1203579300	1730	1203594141	1813
15	1203499098	1900	1203555565	1988	1203566847	2005	1203579359	1789	1203594394	2066
16	1203499098	2036	1203555685	2108	1203567006	2164	1203579580	2010	1203594443	2115
17	1203499390	2192	1203555801	2224	1203567137	2295	1203579380	2133	1203594711	2383
18	1203499536	2338	12035556046	2469	1203567257	2415	1203579703	2223	1203594711	2468
19	1203499330	2504	1203556118	2541	1203567487	2645	1203579793	2472	1203594995	2667
20	1203499702	2593	1203556331	2754	1203567542	2700	1203580042	2520	1203595087	2759
21	1203499970	2772	1203556449	2872	1203567782	2940	1203580070	2800	1203595187	2859
22	1203500016	2818	1203556560	2983	1203567854	3012	1203580370	2894	1203595399	3071
23	1203500310	3145	1203556850	3273	1203568180	3338	1203580404	3201	1203595468	3140
24	1203500345	3218	1203556879	3302	1203568227	3385	1203580771	3227	1203595680	3352
25	1203500589	3391	1203550877	3590	1203568535	3693	1203581043	3473	1203595749	3421
26	1203500798	3600	1203557107	3602	1203568542	3700	1203581043	3555	1203595956	3628
27	1203500738	3726	1203557179	3791	1203568748	3906	1203581355	3785	1203595988	3660
28	1203500324	3948	1203557431	3854	1203568862	4020	1203581333	3878	1203596296	3968
29	1203501140	4061	1203557683	4106	1203569093	4251	1203581448	4042	1203596306	3978
30	1203501237	4303	1203557083	4152	1203569165	4323	1203581012	4186	1203596611	4283
31	1203501560	4362	1203557723	4336	1203569375	4533	1203581780	4319	1203596615	4287
32	1203501800	4626	1203557913	4470	1203569416	4574	1203582001	4431	1203596867	4539
33	1203501824	4664	1203558047	4604	1203569565	4723	1203582096	4526	1203596901	4573
34	1203501802	4960	1203558363	4786	1203569732	4890	1203582303	4733	1203597159	4831
35	1203502174	4976	1203558487	4910	1203569801	4959	1203582387	4817	1203597229	4901
36	1203502367	5169	1203558713	5136	1203570046	5204	1203582599	5029	1203597434	5106
37	1203502307	5229	1203558713	5156	1203570040	5219	1203582748	5178	1203597491	5163
38	1203502427	5499	1203558952	5375	1203570001	5500	1203582748	5264	1203597667	5339
39	1203502777	5529	1203558968	5391	1203570342	5509	1203583074	5504	1203597810	5482
40	1203502727	5755	1203559159	5582	1203570531	5692	1203583074	5526	1203597936	5608
41	1203503022	5824	1203559173	5596	1203570534	5832	1203583090	5727	1203598054	5726
42	1203503022	6029	1203559339	5762	1203570868	6026	1203583297	5734	1203598054	5826
43	1203503227	6055	1203559380	5803	1203570808	6077	1203583304	5919	1203598134	5951
44	1203503253	6267	1203559552	5975	1203571087	6245	1203583494	5924	1203598462	6134
45	1203503463	6358	1203539332	6138	1203571087	6308	1203583494	6151	1203598480	6152
46	1203503536	6465	1203559713	6221	1203571130	6432	1203583721	6164	1203598480	6313
47	1203503003	6549	1203559971	6394	1203571274	6499	1203583734	6426	1203598738	6410
48	1203503747	6666	1203559971	6523	1203571548	6706	1203584043	6473	1203598937	6609
49	1203503864	6740	1203560100	6667	1203571548	6733	1203584043	6694	1203598937	6757
50	1203503938	6896	1203560406	6829	1203571373	6942	1203584204	6702	1203599083	6911
51	1203504094	7031	1203560406	6982	1203571784	7030	1203584272	6994	1203599239	7032
52	1203504229	7157	1203560539	7131	1203571872	7190	1203584564	7004	1203599360	7192
53	1203504537 1203504668	7339 7470	1203560837 1203561016	7260 7439	1203572195 1203572343	7353 7501	1203584880 1203584888	7310 7318	1203599682 1203599818	7354 7490
54										

56	1203505007	7809	1203561328	7751	1203572639	7797	1203585203	7633	1203600070	7742
57	1203505090	7892	1203561486	7909	1203572796	7954	1203585484	7914	1203600328	8000
58	1203505284	8086	1203561600	8023	1203572898	8056	1203585538	7968	1203600400	8072
59	1203505337	8139	1203561836	8259	1203573067	8225	1203585785	8215	1203600619	8291
60	1203505556	8358	1203561847	8270	1203573187	8345	1203585883	8313	1203600671	8343
61	1203505611	8413	1203562108	8531	1203573340	8498	1203586039	8469	1203600856	8528
62	1203505677	8479	1203562138	8561	1203573406	8564	1203586104	8534	1203600923	8595
63	1203505825	8627	1203562173	8596	1203573417	8575	1203586156	8586	1203600923	8595
64	1203505890	8692	1203562202	8625	1203573482	8640	1203586221	8651	1203600987	8659

BOINC Povi	ay 4 Machines									
Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A
1	1203630358	0	1203637523	0	1203643638	0	1203650736	0	1203657711	0
2	1203630445	87	1203637534	11	1203643765	127	1203650758	22	1203657715	4
3	1203630448	90	1203637536	13	1203643815	177	1203650773	37	1203657719	8
4	1203630484	126	1203637587	64	1203643843	205	1203650938	202	1203657883	172
5	1203630588	230	1203637792	269	1203643878	240	1203651024	288	1203657951	240
6	1203630701	343	1203637792	269	1203644041	403	1203651045	309	1203658005	294
7	1203630738	380	1203637831	308	1203644134	496	1203651065	329	1203658038	327
8	1203630809	451	1203637850	327	1203644145	507	1203651238	502	1203658114	403
9	1203630830	472	1203638060	537	1203644146	508	1203651255	519	1203658236	525
10	1203630992	634	1203638080	557	1203644349	711	1203651327	591	1203658328	617
11	1203631043	685	1203638116	593	1203644436	798	1203651365	629	1203658348	637
12	1203631136	778	1203638144	621	1203644440	802	1203651528	792	1203658443	732
13	1203631152	794	1203638346	823	1203644462	824	1203651528	792	1203658525	814
14	1203631308	950	1203638370	847	1203644651	1013	1203651666	930	1203658596	885
15	1203631339	981	1203638375	852	1203644715	1077	1203651719	983	1203658648	937
16	1203631459	1101	1203638467	944	1203644786	1148	1203651780	1044	1203658743	1032
17	1203631479	1121	1203638646	1123	1203644812	1174	1203651928	1192	1203658798	1087
18	1203631566	1208	1203638709	1186	1203645006	1368	1203652007	1271	1203658838	1127
19	1203631583	1225	1203638719	1196	1203645031	1393	1203652081	1345	1203658988	1277
20	1203631795	1437	1203638740	1217	1203645031	1393	1203652097	1361	1203659083	1372
21	1203631832	1474	1203638903	1380	1203645145	1507	1203652249	1513	1203659198	1487
22	1203631865	1507	1203639013	1490	1203645251	1613	1203652335	1599	1203659211	1500
23	1203631907	1549	1203639108	1585	1203645257	1619	1203652353	1617	1203659351	1640
24	1203632112	1754	1203639143	1620	1203645341	1703	1203652399	1663	1203659410	1699
25	1203632118	1760	1203639237	1714	1203645443	1805	1203652576	1840	1203659472	1761
26	1203632194	1836	1203639271	1748	1203645485	1847	1203652599	1863	1203659534	1823
27	1203632237	1879	1203639466	1943	1203645613	1975	1203652605	1869	1203659641	1930
28	1203632414	2056	1203639494	1971	1203645657	2019	1203652669	1933	1203659689	1978
29	1203632456	2098	1203639567	2044	1203645740	2102	1203652905	2169	1203659707	1996
30	1203632458	2100	1203639580	2057	1203645890	2252	1203652934	2198	1203659829	2118
31	1203632577	2219	1203639696	2173	1203645975	2337	1203652958	2222	1203659934	2223
32	1203632657	2299	1203639797	2274	1203645997	2359	1203652986	2250	1203659969	2258
33	1203632719	2361	1203639822	2299	1203646002	2364	1203653183	2447	1203659970	2259
34	1203632787	2429	1203639853	2330	1203646219	2581	1203653198	2462	1203660097	2386
35	1203632927	2569	1203639987	2464	1203646238	2600	1203653204	2468	1203660215	2504
36	1203632938	2580	1203639992	2469	1203646256	2618	1203653218	2482	1203660297	2586
37	1203632960	2602	1203640095	2572	1203646293	2655	1203653385	2649	1203660318	2607
38	1203633115	2757	1203640169	2646	1203646473	2835	1203653399	2663	1203660321	2610
39	1203633200	2842	1203640217	2694	1203646506	2868	1203653403	2667	1203660502	2791
40	1203633206	2848	1203640298	2775	1203646525	2887	1203653423	2687	1203660519	2808
41	1203633278	2920	1203640318	2795	1203646619	2981	1203653577	2841	1203660520	2809
42	1203633367	3009	1203640424	2901	1203646659	3021	1203653629	2893	1203660532	2821
43	1203633452	3094	1203640514	2991	1203646703	3065	1203653670	2934	1203660694	2983
44	1203633474	3116	1203640528	3005	1203646729	3091	1203653680	2944	1203660739	3028
45	1203633567	3209	1203640560	3037	1203646848	3210	1203653845	3109	1203660776	3065
46	1203633684	3326	1203640633	3110	1203646896	3258	1203653929	3193	1203660842	3131
47	1203633699	3341	1203640808	3285	1203647004	3366	1203653994	3258	1203660935	3224

48	1203633703	3345	1203640837	3314	1203647050	3412	1203654004	3268	1203661013	3302
49	1203633773	3415	1203640889	3366	1203647171	3533	1203654141	3405	1203661077	3366
50	1203633906	3548	1203640950	3427	1203647200	3562	1203654265	3529	1203661171	3460
51	1203633939	3581	1203641114	3591	1203647332	3694	1203654322	3586	1203661187	3476
52	1203633999	3641	1203641167	3644	1203647367	3729	1203654325	3589	1203661342	3631
53	1203634053	3695	1203641216	3693	1203647487	3849	1203654464	3728	1203661402	3691
54	1203634274	3916	1203641273	3750	1203647549	3911	1203654603	3867	1203661445	3734
55	1203634285	3927	1203641433	3910	1203647576	3938	1203654643	3907	1203661493	3782
56	1203634298	3940	1203641438	3915	1203647646	4008	1203654655	3919	1203661658	3947
57	1203634402	4044	1203641484	3961	1203647738	4100	1203654716	3980	1203661726	4015
58	1203634468	4110	1203641535	4012	1203647804	4166	1203654782	4046	1203661791	4080
59	1203634525	4167	1203641549	4026	1203647817	4179	1203654872	4136	1203661806	4095
60	1203634589	4231	1203641600	4077	1203647838	4200	1203654898	4162	1203661823	4112
61	1203634597	4239	1203641707	4184	1203647883	4245	1203654901	4165	1203661871	4160
62	1203634606	4248	1203641723	4200	1203647903	4265	1203654937	4201	1203661889	4178
63	1203634662	4304	1203641773	4250	1203647940	4302	1203654964	4228	1203661926	4215
64	1203634670	4312	1203641788	4265	1203648005	4367	1203654968	4232	1203661992	4281

BOINC Pov	ray 8 Machines									
Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A
1	1203929020	0	1203931623	0	1203934208	0	1203936790	0	1203939426	0
2	1203929065	45	1203931640	17	1203934241	33	1203936804	14	1203939477	51
3	1203929113	93	1203931656	33	1203934317	109	1203936877	87	1203939479	53
4	1203929121	101	1203931668	45	1203934359	151	1203936899	109	1203939517	91
5	1203929124	104	1203931676	53	1203934366	158	1203936907	117	1203939521	95
6	1203929143	123	1203931697	74	1203934368	160	1203936911	121	1203939607	181
7	1203929233	213	1203931721	98	1203934409	201	1203936963	173	1203939620	194
8	1203929247	227	1203931735	112	1203934424	216	1203936966	176	1203939624	198
9	1203929305	285	1203931918	295	1203934499	291	1203937091	301	1203939684	258
10	1203929377	357	1203931948	325	1203934547	339	1203937122	332	1203939764	338
11	1203929389	369	1203931962	339	1203934591	383	1203937163	373	1203939779	353
12	1203929426	406	1203931967	344	1203934634	426	1203937206	416	1203939810	384
13	1203929442	422	1203931970	347	1203934648	440	1203937216	426	1203939816	390
14	1203929459	439	1203931984	361	1203934667	459	1203937300	510	1203939841	415
15	1203929550	530	1203931994	371	1203934696	488	1203937327	537	1203939857	431
16	1203929561	541	1203932036	413	1203934726	518	1203937364	574	1203939937	511
17	1203929617	597	1203932194	571	1203934839	631	1203937390	600	1203939975	549
18	1203929632	612	1203932201	578	1203934865	657	1203937446	656	1203940072	646
19	1203929649	629	1203932251	628	1203934893	685	1203937477	687	1203940080	654
20	1203929690	670	1203932269	646	1203934937	729	1203937507	717	1203940122	696
21	1203929783	763	1203932309	686	1203934940	732	1203937554	764	1203940161	735
22	1203929799	779	1203932325	702	1203934994	786	1203937620	830	1203940241	815
23	1203929802	782	1203932337	714	1203935033	825	1203937649	859	1203940259	833
24	1203929882	862	1203932395	772	1203935096	888	1203937655	865	1203940297	871
25	1203929908	888	1203932453	830	1203935138	930	1203937709	919	1203940298	872
26	1203929941	921	1203932474	851	1203935146	938	1203937724	934	1203940392	966
27	1203929980	960	1203932569	946	1203935200	992	1203937752	962	1203940417	991
28	1203930009	989	1203932598	975	1203935222	1014	1203937825	1035	1203940464	1038
29	1203930038	1018	1203932654	1031	1203935283	1075	1203937839	1049	1203940479	1053
30	1203930061	1041	1203932665	1042	1203935322	1114	1203937870	1080	1203940485	1059
31	1203930100	1080	1203932687	1064	1203935349	1141	1203937879	1089	1203940569	1143
32	1203930171	1151	1203932744	1121	1203935370	1162	1203937880	1090	1203940576	1150
33	1203930211	1191	1203932780	1157	1203935411	1203	1203937917	1127	1203940627	1201
34	1203930231	1211	1203932797	1174	1203935492	1284	1203937926	1136	1203940651	1225
35	1203930240	1220	1203932813	1190	1203935528	1320	1203937949	1159	1203940668	1242
36	1203930246	1226	1203932870	1247	1203935529	1321	1203938033	1243	1203940731	1305
37	1203930264	1244	1203932912	1289	1203935572	1364	1203938052	1262	1203940755	1329
38	1203930287	1267	1203932945	1322	1203935597	1389	1203938083	1293	1203940777	1351

39	1203930308	1288	1203932994	1371	1203935644	1436	1203938141	1351	1203940809	1383
40	1203930378	1358	1203932998	1375	1203935682	1474	1203938174	1384	1203940840	1414
41	1203930461	1441	1203932998	1375	1203935697	1489	1203938199	1409	1203940842	1416
42	1203930481	1461	1203933006	1383	1203935733	1525	1203938208	1418	1203940850	1424
43	1203930533	1513	1203933067	1444	1203935803	1595	1203938272	1482	1203940871	1445
44	1203930543	1523	1203933151	1528	1203935809	1601	1203938334	1544	1203940938	1512
45	1203930545	1525	1203933192	1569	1203935841	1633	1203938354	1564	1203941028	1602
46	1203930564	1544	1203933204	1581	1203935896	1688	1203938407	1617	1203941095	1669
47	1203930636	1616	1203933244	1621	1203935932	1724	1203938469	1679	1203941107	1681
48	1203930708	1688	1203933272	1649	1203935938	1730	1203938501	1711	1203941143	1717
49	1203930809	1789	1203933306	1683	1203935961	1753	1203938528	1738	1203941171	1745
50	1203930824	1804	1203933334	1711	1203935989	1781	1203938541	1751	1203941185	1759
51	1203930830	1810	1203933372	1749	1203936005	1797	1203938593	1803	1203941194	1768
52	1203930832	1812	1203933390	1767	1203936026	1818	1203938606	1816	1203941238	1812
53	1203930856	1836	1203933399	1776	1203936054	1846	1203938612	1822	1203941251	1825
54	1203930871	1851	1203933455	1832	1203936071	1863	1203938650	1860	1203941260	1834
55	1203930875	1855	1203933474	1851	1203936113	1905	1203938666	1876	1203941282	1856
56	1203930889	1869	1203933479	1856	1203936121	1913	1203938675	1885	1203941302	1876
57	1203930896	1876	1203933508	1885	1203936164	1956	1203938677	1887	1203941347	1921
58	1203930898	1878	1203933513	1890	1203936179	1971	1203938715	1925	1203941366	1940
59	1203930921	1901	1203933539	1916	1203936186	1978	1203938732	1942	1203941374	1948
60	1203930930	1910	1203933544	1921	1203936187	1979	1203938740	1950	1203941379	1953
61	1203930936	1916	1203933545	1922	1203936189	1981	1203938743	1953	1203941388	1962
62	1203930976	1956	1203933573	1950	1203936229	2021	1203938763	1973	1203941439	2013
63	1203930996	1976	1203933578	1955	1203936251	2043	1203938807	2017	1203941444	2018
64	1203931041	2021	1203933610	1987	1203936254	2046	1203938828	2038	1203941454	2028

BOINC Povra	BOINC Povray 16 Machines										
Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A	
1	1202453063	0	1202454796	0	1202456161	0	1202700387	0	1202702060	0	
2	1202453135	72	1202454821	25	1202456210	49	1202700468	81	1202702127	67	
3	1202453144	81	1202454843	47	1202456277	116	1202700482	95	1202702143	83	
4	1202453163	100	1202454887	91	1202456319	158	1202700483	96	1202702149	89	
5	1202453168	105	1202454906	110	1202456326	165	1202700488	101	1202702150	90	
6	1202453190	127	1202454909	113	1202456358	197	1202700506	119	1202702151	91	
7	1202453233	170	1202454909	113	1202456364	203	1202700510	123	1202702158	98	
8	1202453240	177	1202454926	130	1202456367	206	1202700518	131	1202702171	111	
9	1202453262	199	1202454926	130	1202456381	220	1202700518	131	1202702173	113	
10	1202453273	210	1202454930	134	1202456388	227	1202700519	132	1202702182	122	
11	1202453274	211	1202454946	150	1202456395	234	1202700533	146	1202702212	152	
12	1202453281	218	1202454949	153	1202456414	253	1202700547	160	1202702250	190	
13	1202453283	220	1202454971	175	1202456415	254	1202700564	177	1202702251	191	
14	1202453295	232	1202454978	182	1202456417	256	1202700571	184	1202702261	201	
15	1202453300	237	1202455022	226	1202456418	257	1202700602	215	1202702268	208	
16	1202453339	276	1202455044	248	1202456420	259	1202700612	225	1202702347	287	
17	1202453371	308	1202455050	254	1202456460	299	1202700669	282	1202702384	324	
18	1202453391	328	1202455098	302	1202456500	339	1202700712	325	1202702403	343	
19	1202453417	354	1202455115	319	1202456590	429	1202700720	333	1202702427	367	
20	1202453452	389	1202455124	328	1202456593	432	1202700734	347	1202702429	369	
21	1202453459	396	1202455142	346	1202456594	433	1202700792	405	1202702445	385	
22	1202453486	423	1202455147	351	1202456626	465	1202700826	439	1202702452	392	
23	1202453496	433	1202455176	380	1202456640	479	1202700836	449	1202702462	402	
24	1202453497	434	1202455209	413	1202456641	480	1202700861	474	1202702462	402	
25	1202453498	435	1202455223	427	1202456648	487	1202700862	475	1202702468	408	
26	1202453503	440	1202455243	447	1202456655	494	1202700863	476	1202702470	410	
27	1202453541	478	1202455254	458	1202456659	498	1202700864	477	1202702475	415	
28	1202453564	501	1202455259	463	1202456664	503	1202700865	478	1202702476	416	
29	1202453566	503	1202455271	475	1202456669	508	1202700867	480	1202702483	423	
30	1202453571	508	1202455275	479	1202456706	545	1202700872	485	1202702500	440	

31	1202453597	534	1202455300	504	1202456730	569	1202700887	500	1202702543	483
32	1202453639	576	1202455311	515	1202456734	573	1202700892	505	1202702587	527
33	1202453644	581	1202455317	521	1202456735	574	1202701030	643	1202702594	534
34	1202453709	646	1202455323	527	1202456772	611	1202701032	645	1202702612	552
35	1202453714	651	1202455354	558	1202456800	639	1202701033	646	1202702659	599
36	1202453745	682	1202455382	586	1202456837	676	1202701049	662	1202702677	617
37	1202453750	687	1202455387	591	1202456847	686	1202701080	693	1202702685	625
38	1202453758	695	1202455405	609	1202456894	733	1202701095	708	1202702700	640
39	1202453770	707	1202455416	620	1202456912	751	1202701097	710	1202702713	653
40	1202453776	713	1202455419	623	1202456913	752	1202701098	711	1202702722	662
41	1202453780	717	1202455445	649	1202456922	761	1202701114	727	1202702750	690
42	1202453788	725	1202455450	654	1202456927	766	1202701135	748	1202702753	693
43	1202453804	741	1202455455	659	1202456931	770	1202701145	758	1202702753	693
44	1202453810	747	1202455470	674	1202456932	771	1202701146	759	1202702755	695
45	1202453815	752	1202455481	685	1202456933	772	1202701151	764	1202702766	706
46	1202453823	760	1202455510	714	1202456937	776	1202701162	775	1202702769	709
47	1202453833	770	1202455515	719	1202456953	792	1202701164	777	1202702779	719
48	1202453834	771	1202455520	724	1202456959	798	1202701164	777	1202702780	720
49	1202453835	772	1202455539	743	1202456975	814	1202701171	784	1202702788	728
50	1202453836	773	1202455540	744	1202456979	818	1202701179	792	1202702790	730
51	1202453837	774	1202455547	751	1202456981	820	1202701187	800	1202702804	744
52	1202453841	778	1202455552	756	1202456987	826	1202701191	804	1202702806	746
53	1202453853	790	1202455556	760	1202456992	831	1202701200	813	1202702818	758
54	1202453854	791	1202455568	772	1202456996	835	1202701210	823	1202702820	760
55	1202453869	806	1202455588	792	1202456997	836	1202701211	824	1202702820	760
56	1202453883	820	1202455602	806	1202456998	837	1202701216	829	1202702835	775
57	1202453886	823	1202455604	808	1202457001	840	1202701227	840	1202702845	785
58	1202453899	836	1202455605	809	1202457019	858	1202701229	842	1202702853	793
59	1202453900	837	1202455612	816	1202457028	867	1202701230	843	1202702855	795
60	1202453901	838	1202455617	821	1202457040	879	1202701236	849	1202702859	799
61	1202453904	841	1202455621	825	1202457042	881	1202701244	857	1202702870	810
62	1202453919	856	1202455633	837	1202457046	885	1202701252	865	1202702872	812
63	1202453948	885	1202455653	857	1202457093	932	1202701256	869	1202702918	858
64	1202453951	888	1202455667	871	1202457107	946	1202701275	888	1202702925	865

CompTorrent

CompTorrer	nt Povray 1 Machin	e								
Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A
1	1205943096	0	1205975852	0	1205995963	0	1206014453	0	1206031175	0
2	1205942882	214	1205975637	215	1205995747	216	1206014235	218	1206030959	216
3	1205942658	438	1205975413	439	1205995523	440	1206014009	444	1206030731	444
4	1205942421	675	1205975175	677	1205995285	678	1206013771	682	1206030492	683
5	1205942162	934	1205974917	935	1205995025	938	1206013510	943	1206030232	943
6	1205941886	1210	1205974641	1211	1205994748	1215	1206013233	1220	1206029954	1221
7	1205941648	1448	1205974401	1451	1205994510	1453	1206012992	1461	1206029709	1466
8	1205941417	1679	1205974170	1682	1205994277	1686	1206012760	1693	1206029478	1697
9	1205941181	1915	1205973934	1918	1205994040	1923	1206012523	1930	1206029241	1934
10	1205940944	2152	1205973696	2156	1205993803	2160	1206012286	2167	1206029004	2171
11	1205940699	2397	1205973452	2400	1205993557	2406	1206012037	2416	1206028757	2418
12	1205940402	2694	1205973154	2698	1205993260	2703	1206011738	2715	1206028461	2714
13	1205940063	3033	1205972814	3038	1205992914	3049	1206011398	3055	1206028123	3052
14	1205939755	3341	1205972506	3346	1205992606	3357	1206011087	3366	1206027811	3364
15	1205939471	3625	1205972220	3632	1205992316	3647	1206010803	3650	1206027525	3650
16	1205939175	3921	1205971921	3931	1205992020	3943	1206010507	3946	1206027229	3946
17	1205938885	4211	1205971631	4221	1205991730	4233	1206010216	4237	1206026937	4238
18	1205938605	4491	1205971348	4504	1205991449	4514	1206009936	4517	1206026652	4523
19	1205938313	4783	1205971056	4796	1205991156	4807	1206009637	4816	1206026359	4816

20	1205938026	5070	1205970765	5087	1205990869	5094	1206009348	5105	1206026068	5107
	1205937735		1205970703							
21		5361		5379	1205990576	5387	1206009054	5399	1206025773	5402
22	1205937451	5645	1205970188	5664	1205990292	5671	1206008770	5683	1206025489	5686
23	1205937161	5935	1205969897	5955	1205990002	5961	1206008479	5974	1206025187	5988
24	1205936897	6199	1205969631	6221	1205989737	6226	1206008212	6241	1206024921	6254
25	1205936657	6439	1205969391	6461	1205989495	6468	1206007971	6482	1206024677	6498
26	1205936431	6665	1205969165	6687	1205989270	6693	1206007745	6708	1206024452	6723
27	1205936230	6866	1205968963	6889	1205989069	6894	1206007543	6910	1206024251	6924
28	1205936051	7045	1205968784	7068	1205988890	7073	1206007362	7091	1206024072	7103
29	1205935875	7221	1205968608	7244	1205988714	7249	1206007185	7268	1206023895	7280
30	1205935695	7401	1205968428	7424	1205988534	7429	1206007004	7449	1206023715	7460
31	1205935527	7569	1205968261	7591	1205988368	7595	1206006837	7616	1206023548	7627
32	1205935354	7742	1205968087	7765	1205988190	7773	1206006664	7789	1206023375	7800
33	1205935161	7935	1205967893	7959	1205987995	7968	1206006470	7983	1206023180	7995
34	1205934960	8136	1205967691	8161	1205987793	8170	1206006269	8184	1206022979	8196
35	1205934752	8344	1205967484	8368	1205987584	8379	1206006061	8392	1206022770	8405
36	1205934537	8559	1205967268	8584	1205987369	8594	1206005845	8608	1206022555	8620
37	1205934229	8867	1205966961	8891	1205987061	8902	1206005533	8920	1206022246	8929
38	1205933909	9187	1205966639	9213	1205986739	9224	1206005212	9241	1206021919	9256
39	1205933653	9443	1205966383	9469	1205986482	9481	1206004953	9500	1206021664	9511
40	1205933432	9664	1205966162	9690	1205986261	9702	1206004731	9722	1206021442	9733
41	1205933200	9896	1205965928	9924	1205986027	9936	1206004498	9955	1206021209	9966
42	1205932911	10185	1205965576	10276	1205985737	10226	1206004206	10247	1206020918	10257
43	1205932630	10466	1205965241	10611	1205985455	10508	1206003926	10527	1206020635	10540
44	1205932374	10722	1205964983	10869	1205985198	10765	1206003670	10783	1206020377	10798
45	1205932133	10963	1205964738	11114	1205984955	11008	1206003428	11025	1206020134	11041
46	1205931842	11254	1205964447	11405	1205984666	11297	1206003137	11316	1206019844	11331
47	1205931559	11537	1205964161	11691	1205984382	11581	1206002854	11599	1206019561	11614
48	1205931251	11845	1205963853	11999	1205984071	11892	1206002545	11908	1206019254	11921
49	1205930938	12158	1205963539	12313	1205983758	12205	1206002231	12222	1206018939	12236
50	1205930617	12479	1205963217	12635	1205983437	12526	1206001910	12543	1206018617	12558
51	1205930258	12838	1205962857	12995	1205983078	12885	1206001551	12902	1206018257	12918
52	1205929899	13197	1205962497	13355	1205982707	13256	1206001188	13265	1206017898	13277
53	1205929598	13498	1205962194	13658	1205982405	13558	1206000887	13566	1206017596	13579
54	1205929296	13800	1205961889	13963	1205982103	13860	1206000585	13868	1206017290	13885
55	1205929030	14066	1205961622	14230	1205981837	14126	1206000319	14134	1206017024	14151
56	1205928763	14333	1205961354	14498	1205981570	14393	1206000052	14401	1206016756	14419
57	1205928485	14611	1205961076	14776	1205981292	14671	1205999774	14679	1206016478	14697
58	1205928227	14869	1205960816	15036	1205981034	14929	1205999516	14937	1206016218	14957
59	1205927956	15140	1205960541	15311	1205980761	15202	1205999245	15208	1206015947	15228
60	1205927701	15395	1205960284	15568	1205980504	15459	1205998989	15464	1206015691	15484
61	1205927472	15624	1205960055	15797	1205980276	15687	1205998761	15692	1206015462	15713
62	1205927251	15845	1205959832	16020	1205980054	15909	1205998538	15915	1206015402	15936
63	1205927033	16063	1205959612	16240	1205979832	16131	1205998320	16133	1206015020	16155
64	1205927033	16266	1205959408	16444	1205979629	16334	1205998320	16336	1206013020	16360
04	1203920830	10200	1203939408	10444	12009/9029	10334	120399811/	10330	1200014815	10300

CompTorrent I	Povray 2 Machine	es								
Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A
1	1206055207	0	1206065779	0	1206074825	0	1206086446	0	1206096947	0
2	1206055013	194	1206065612	167	1206074630	195	1206086253	193	1206096749	198
3	1206054973	234	1206065565	214	1206074593	232	1206086213	233	1206096714	233
4	1206054789	418	1206065370	409	1206074405	420	1206086029	417	1206096525	422
5	1206054716	491	1206065329	450	1206074335	490	1206085956	490	1206096456	491
6	1206054531	676	1206065071	708	1206074147	678	1206085771	675	1206096267	680
7	1206054416	791	1206065070	709	1206074033	792	1206085656	790	1206096156	791
8	1206054292	915	1206064834	945	1206073908	917	1206085532	914	1206096027	920
9	1206054166	1041	1206064820	959	1206073783	1042	1206085405	1041	1206095905	1042
10	1206054056	1151	1206064598	1181	1206073673	1152	1206085295	1151	1206095790	1157
11	1206053908	1299	1206064563	1216	1206073525	1300	1206085147	1299	1206095648	1299

12	1206052011	1206	1206064252	1426	1206072420	1207	1206095040	1207	1206005546	1401
12	1206053811	1396	1206064353	1426	1206073428	1397	1206085049	1397	1206095546	1401
13	1206053586	1621	1206064242	1537	1206073203	1622	1206084825	1621	1206095326	1621
14	1206053472	1735	1206064017	1762	1206073090	1735	1206084711	1735	1206095208	1739
15	1206053251	1956	1206063908	1871	1206072868	1957	1206084491	1955	1206094991	1956
16	1206053188	2019	1206063732	2047	1206072805	2020	1206084426	2020	1206094922	2025
17	1206052930	2277	1206063588	2191	1206072547	2278	1206084170	2276	1206094670	2277
18	1206052898	2309	1206063443	2336	1206072515	2310	1206084135	2311	1206094632	2315
19	1206052626	2581	1206063284	2495	1206072244	2581	1206083866	2580	1206094366	2581
20	1206052606	2601	1206063152	2627	1206072221	2604	1206083843	2603	1206094339	2608
21	1206052314	2893	1206062743	3036	1206071932	2893	1206083554	2892	1206094052	2895
22	1206052313	2894	1206062627	3152	1206071929	2896	1206083552	2894	1206094050	2897
23	1206052029	3178	1206062434	3345	1206071645	3180	1206083269	3177	1206093766	3181
24	1206052000	3207	1206062337	3442	1206071618	3207	1206083240	3206	1206093736	3211
25	1206051764	3443	1206062146	3633	1206071380	3445	1206083003	3443	1206093501	3446
26	1206051740	3467	1206062097	3682	1206071357	3468	1206082980	3466	1206093476	3471
27	1206051539	3668	1206061902	3877	1206071155	3670	1206082778	3668	1206093276	3671
28	1206051521	3686	1206061896	3883	1206071138	3687	1206082761	3685	1206093257	3690
29	1206051360	3847	1206061716	4063	1206070976	3849	1206082599	3847	1206093096	3851
30	1206051329	3878	1206061711	4068	1206070947	3878	1206082570	3876	1206093066	3881
31	1206051181	4026	1206061536	4243	1206070796	4029	1206082419	4027	1206092917	4030
32	1206051148	4059	1206061530	4249	1206070766	4059	1206082390	4056	1206092885	4062
33	1206051007	4200	1206061363	4416	1206070623	4202	1206082246	4200	1206092744	4203
34	1206050938	4269	1206061320	4459	1206070556	4269	1206082179	4267	1206092675	4272
35	1206050806	4401	1206061162	4617	1206070421	4404	1206082045	4401	1206092542	4405
36	1206050712	4495	1206061095	4684	1206070330	4495	1206081954	4492	1206092449	4498
37	1206050590	4617	1206060947	4832	1206070206	4619	1206081830	4616	1206092327	4620
38	1206050379	4828	1206060761	5018	1206069996	4829	1206081620	4826	1206092116	4831
39	1206050269	4938	1206060624	5155	1206069884	4941	1206081509	4937	1206092005	4942
40	1206050209	5106	1206060483	5296	1206069718	5107	1206081343	5103	1206091838	5109
41	1206050101	5159	1206060403	5377	1206069718	5162	1206081343	5159	1206091784	5163
42	1206049817	5390	1206060232	5547	1206069432	5393	1206081287	5391	1206091553	5394
43	1206049817	5421	1206060232	5667	1206069432	5422	1206081033	5418	1206091533	5424
44	1206049536	5671	1206059928	5851	1206069152	5673	1206080775	5671	1206091272	5675 5702
45	1206049508	5699	1206059854	5925	1206069125	5700	1206080750	5696		
46	1206049247	5960	1206059614	6165	1206068864	5961	1206080489	5957	1206090984	5963
47	1206049246	5961	1206059613	6166	1206068862	5963	1206080485	5961	1206090982	5965
48	1206048964	6243	1206059331	6448	1206068580	6245	1206080203	6243	1206090700	6247
49	1206048915	6292	1206059280	6499	1206068531	6294	1206080156	6290	1206090651	6296
50	1206048650	6557	1206059017	6762	1206068266	6559	1206079889	6557	1206090386	6561
51	1206048567	6640	1206058932	6847	1206068183	6642	1206079808	6638	1206090302	6645
52	1206048292	6915	1206058659	7120	1206067908	6917	1206079531	6915	1206090027	6920
53	1206048177	7030	1206058543	7236	1206067795	7030	1206079420	7026	1206089913	7034
54	1206047990	7217	1206058357	7422	1206067606	7219	1206079229	7217	1206089725	7222
55	1206047849	7358	1206058215	7564	1206067467	7358	1206079092	7354	1206089585	7362
56	1206047725	7482	1206058092	7687	1206067340	7485	1206078963	7483	1206089459	7488
57	1206047560	7647	1206057926	7853	1206067178	7647	1206078803	7643	1206089295	7652
58	1206047444	7763	1206057813	7966	1206067062	7763	1206078685	7761	1206089182	7765
59	1206047280	7927	1206057646	8133	1206066898	7927	1206078523	7923	1206089015	7932
60	1206047173	8034	1206057542	8237	1206066791	8034	1206078414	8032	1206088911	8036
61	1206047002	8205	1206057369	8410	1206066621	8204	1206078245	8201	1206088738	8209
62	1206046944	8263	1206057313	8466	1206066563	8262	1206078186	8260	1206088682	8265
63	1206046762	8445	1206057128	8651	1206066380	8445	1206078003	8443	1206088498	8449
64	1206046726	8481	1206057093	8686	1206066345	8480	1206077967	8479	1206088464	8483
	l	1	I		1	İ	1		1	

CompTorrent Povray 4 Machines											
Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A	
1	1206102846	0	1206145979	0	1206153213	0	1206160835	0	1206167040	0	
2	1206102839	7	1206145884	95	1206153178	35	1206160831	4	1206166868	172	
3	1206102814	32	1206145861	118	1206153148	65	1206160798	37	1206166830	210	

6	1206102603	243	1206145640	339	1206152935	278	1206160588	247	1206166624	416
	1206102538	308		376	1206152872	341	1206160525	310	1206166549	491
8	1206102538	313	1206145603 1206145598	3/6	1206152872	341	1206160525	310	1206166549	491
9	1206102349	497	1206145446	533	1206152721	492	1206160344	491	1206166542	498
10	1206102344	502	1206145380	599	1206152676	537	1206160329	506	1206166364	676
11	1206102283	563	1206145367	612	1206152617	596	1206160275	560	1206166311	729
12	1206102280	566	1206145347	632	1206152615	598	1206160261	574	1206166291	749
13	1206102111	735	1206145189	790	1206152484	729	1206160107	728	1206166291	749
14	1206102078	768	1206145122	857	1206152410	803	1206160063	772	1206166025	1015
15	1206101957	889	1206145055	924	1206152293	920	1206159939	896	1206166013	1027
16	1206101916	930	1206144980	999	1206152246	967	1206159908	927	1206165995	1045
17	1206101803	1043	1206144855	1124	1206152175	1038	1206159798	1037	1206165955	1085
18	1206101768	1078	1206144838	1141	1206152101	1112	1206159754	1081	1206165728	1312
19	1206101634	1212	1206144732	1247	1206151971	1242	1206159618	1217	1206165704	1336
20	1206101602	1244	1206144664	1315	1206151931	1282	1206159592	1243	1206165678	1362
21	1206101523	1323	1206144557	1422	1206151895	1318	1206159517	1318	1206165650	1390
22	1206101451	1395	1206144537	1442	1206151784	1429	1206159437	1398	1206165435	1605
23	1206101321	1525	1206144420	1559	1206151658	1555	1206159304	1531	1206165391	1649
24	1206101285	1561	1206144347	1632	1206151612	1601	1206159275	1560	1206165360	1680
25	1206101239	1607	1206144273	1706	1206151610	1603	1206159233	1602	1206165341	1699
26	1206101137	1709	1206144223	1756	1206151469	1744	1206159122	1713	1206165145	1895
27	1206101033	1813	1206144132	1847	1206151370	1843	1206159016	1819	1206165102	1938
28	1206101024	1822	1206144086	1893	1206151369	1844	1206159015	1820	1206165097	1943
29	1206101013	1833	1206144047	1932	1206151367	1846	1206159007	1828	1206165096	1944
30	1206100917	1929	1206144005	1974	1206151251	1962	1206158904	1931	1206164943	2097
31	1206100838	2008	1206143936	2043	1206151189	2024	1206158828	2007	1206164907	2133
32	1206100837	2009	1206143895	2084	1206151178	2035	1206158825	2010	1206164905	2135
33	1206100829	2017	1206143868	2111	1206151172	2041	1206158819	2016	1206164900	2140
34	1206100736	2110	1206143823	2156	1206151070	2143	1206158722	2113	1206164775	2265
35	1206100664	2182	1206143749	2230	1206151016	2197	1206158655	2180	1206164713	2327
36	1206100627	2219	1206143684	2295	1206150967	2246	1206158614	2221	1206164696	2344
37	1206100611	2235	1206143667	2312	1206150952	2261	1206158601	2234	1206164685	2355
38	1206100510	2336	1206143598	2381	1206150844	2369	1206158496	2339	1206164566	2474
39	1206100448	2398	1206143516	2463	1206150717	2496	1206158368	2467	1206164479	2561
40	1206100292	2554	1206143350	2629	1206150708	2505	1206158346	2489	1206164361	2679
41	1206100263	2583	1206143345	2634	1206150618	2595	1206158265	2570	1206164334	2706
42	1206100232	2614	1206143320	2659	1206150566	2647	1206158218	2617	1206164309	2731
43	1206100227	2619	1206143275	2704	1206150486	2727	1206158125	2710	1206164237	2803
44	1206099995	2851	1206143114	2865	1206150465	2748	1206158116	2719	1206164077	2963
45	1206099976	2870	1206143034	2945	1206150302	2911	1206157950	2885	1206164046	2994
46	1206099959	2887	1206143034	2963	1206150362	2952	1206157914	2921	1206164027	3013
46	1206099959	2891	1206143016	2983	1206150261	2984	1206157914	2921		3013
48	1206099955	3092		3106		3010	1206157869	2966	1206163959	3081
			1206142873		1206150203	3226		3199	1206163835	
49	1206099660	3186	1206142719	3260	1206149987		1206157636		1206163730	3310
50	1206099653	3193	1206142709	3270	1206149955	3258	1206157607	3228	1206163720	3320
51	1206099622	3224	1206142663	3316	1206149923	3290	1206157562	3273	1206163626	3414
52	1206099440	3406	1206142560	3419	1206149863	3350	1206157514	3321	1206163520	3520
53	1206099312	3534	1206142371	3608	1206149638	3575	1206157288	3547	1206163381	3659
54	1206099264	3582	1206142319	3660	1206149566	3647	1206157218	3617	1206163330	3710
55	1206099232	3614	1206142273	3706	1206149564	3649	1206157202	3633	1206163236	3804
56	1206099139	3707	1206142258	3721	1206149536	3677	1206157188	3647	1206163218	3822
57	1206098983	3863	1206142043	3936	1206149310	3903	1206156960	3875	1206163053	3987
58	1206098976	3870	1206142031	3948	1206149297	3916	1206156936	3899	1206163039	4001
59	1206098943	3903	1206141991	3988	1206149276	3937	1206156928	3907	1206162951	4089
60	1206098861	3985	1206141972	4007	1206149234	3979	1206156886	3949	1206162934	4106
61	1206098697	4149	1206141762	4217	1206149029	4184	1206156680	4155	1206162772	4268
01										

63	1206098665	4181	1206141736	4243	1206148999	4214	1206156651	4184	1206162694	4346	ì
64	1206098631	4215	1206141723	4256	1206148985	4228	1206156637	4198	1206162685	4355	ı

CompTorrent	Povray 8 Machin	es								
Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A
1	1206496870	0	1206499559	0	1206503578	0	1206506225	0	1206514357	0
2	1206496859	11	1206499550	9	1206503570	8	1206506217	8	1206514351	6
3	1206496857	13	1206499547	12	1206503564	14	1206506216	9	1206514347	10
4	1206496851	19	1206499545	14	1206503562	16	1206506199	26	1206514340	17
5	1206496833	37	1206499537	22	1206503556	22	1206506192	33	1206514330	27
6	1206496752	118	1206499446	113	1206503479	99	1206506110	115	1206514248	109
7	1206496673	197	1206499396	163	1206503414	164	1206506036	189	1206514174	183
8	1206496654	216	1206499341	218	1206503362	216	1206506000	225	1206514131	226
9	1206496625	245	1206499303	256	1206503346	232	1206505000	234	1206514098	259
10	1206496608	262	1206499303	256	1206503340	257	1206505973	252	1206514096	261
11	1206496599	271	1206499292	267	1206503321	274	1206505940	285	1206514093	264
								289		
12	1206496589	281	1206499279	280	1206503298	280	1206505936		1206514076	281
13	1206496533	337	1206499268	291	1206503271	307	1206505915	310	1206514069	288
14	1206496493	377	1206499188	371	1206503219	359	1206505852	373	1206513989	368
15	1206496423	447	1206499145	414	1206503164	414	1206505785	440	1206513923	434
16	1206496416	454	1206499046	513	1206503089	489	1206505742	483	1206513839	518
17	1206496342	528	1206499030	529	1206503063	515	1206505678	547	1206513831	526
18	1206496322	548	1206499026	533	1206503005	573	1206505669	556	1206513826	531
19	1206496284	586	1206498979	580	1206502999	579	1206505649	576	1206513775	582
20	1206496257	613	1206498974	585	1206502995	583	1206505624	601	1206513764	593
21	1206496199	671	1206498943	616	1206502968	610	1206505604	621	1206513741	616
22	1206496182	688	1206498878	681	1206502909	669	1206505541	684	1206513680	677
23	1206496119	751	1206498820	739	1206502843	735	1206505463	762	1206513601	756
24	1206496108	762	1206498739	820	1206502774	804	1206505425	800	1206513539	818
25	1206496037	833	1206498739	820	1206502757	821	1206505387	838	1206513534	823
26	1206496004	866	1206498709	850	1206502707	871	1206505360	865	1206513508	849
27	1206495970	900	1206498666	893	1206502692	886	1206505335	890	1206513462	895
28	1206495940	930	1206498657	902	1206502679	899	1206505308	917	1206513447	910
29	1206495890	980	1206498634	925	1206502658	920	1206505295	930	1206513431	926
30	1206495867	1003	1206498564	995	1206502593	985	1206505226	999	1206513364	993
31	1206495853	1017	1206498531	1028	1206502555	1023	1206505174	1051	1206513314	1043
32	1206495847	1023	1206498497	1062	1206502513	1065	1206505163	1062	1206513298	1059
33	1206495793	1077	1206498493	1066	1206502511	1067	1206505161	1064	1206513289	1068
34	1206495784	1086	1206498490	1069	1206502505	1073	1206505140	1085	1206513288	1069
35	1206495775	1095	1206498472	1087	1206502497	1073	1206505140	1085	1206513267	1090
36	1206495748	1122	1206498465	1094	1206502488	1090	1206505116	1109	1206513257	1102
37	1206495695	1175	1206498438	1121	1206502462	1116	1206505098	1127	1206513236	1121
38	1206495685	1185	1206498383	1176	1206502412	1166	1206505045	1180	1206513182	1175
39	1206495660	1210	1206498342	1217	1206502368	1210	1206504985	1240	1206513126	1231
40	1206495656	1214	1206498302	1257	1206502312	1266	1206504966	1259	1206513102	1255
41	1206495565	1305	1206498274	1285	1206502293	1285	1206504943	1282	1206513068	1289
42	1206495522	1348	1206498247	1312	1206502271	1307	1206504912	1313	1206513041	1316
43	1206495468	1402	1206498203	1356	1206502228	1350	1206504864	1361	1206513001	1356
44	1206495459	1411	1206498156	1403	1206502179	1399	1206504806	1419	1206512954	1403
45	1206495425	1445	1206498115	1444	1206502139	1439	1206504767	1458	1206512906	1451
46	1206495418	1452	1206498104	1455	1206502134	1444	1206504767	1458	1206512904	1453
47	1206495415	1455	1206498101	1458	1206502127	1451	1206504744	1481	1206512886	1471
48	1206495407	1463	1206498068	1491	1206502080	1498	1206504734	1491	1206512868	1489
49	1206495250	1620	1206497957	1602	1206501976	1602	1206504626	1599	1206512752	1605
50	1206495218	1652	1206497942	1617	1206501967	1611	1206504606	1619	1206512736	1621
51	1206495212	1658	1206497924	1635	1206501949	1629	1206504585	1640	1206512723	1634
52	1206495153	1717	1206497852	1707	1206501872	1706	1206504505	1720	1206512643	1714
53	1206495145	1725	1206497841	1718	1206501865	1713	1206504491	1734	1206512638	1719
54	1206495101	1769	1206497797	1762	1206501821	1757	1206504436	1789	1206512578	1779

55	1206495085	1785	1206497768	1791	1206501806	1772	1206504435	1790	1206512573	1784
56	1206495084	1786	1206497753	1806	1206501766	1812	1206504419	1806	1206512552	1805
57	1206494869	2001	1206497575	1984	1206501600	1978	1206504236	1989	1206512373	1984
58	1206494858	2012	1206497568	1991	1206501585	1993	1206504235	1990	1206512361	1996
59	1206494853	2017	1206497553	2006	1206501577	2001	1206504216	2009	1206512347	2010
60	1206494826	2044	1206497523	2036	1206501545	2033	1206504178	2047	1206512315	2042
61	1206494817	2053	1206497512	2047	1206501537	2041	1206504163	2062	1206512310	2047
62	1206494813	2057	1206497508	2051	1206501533	2045	1206504152	2073	1206512289	2068
63	1206494795	2075	1206497477	2082	1206501516	2062	1206504144	2081	1206512284	2073
64	1206494781	2089	1206497474	2085	1206501487	2091	1206504132	2093	1206512270	2087

1			Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A
	1205882097	0	1205883824	0	1206491033	0	1206492565	0	1206494129	0
2	1205882097	0	1205883823	1	1206491025	8	1206492562	3	1206494117	12
3	1205882094	3	1205883820	4	1206491020	13	1206492552	13	1206494110	19
4	1205882086	11	1205883804	20	1206491020	13	1206492552	13	1206494106	23
5	1205882076	21	1205883798	26	1206491004	29	1206492531	34	1206494103	26
6	1205882006	91	1205883769	55	1206490904	129	1206492474	91	1206494068	61
7	1205882001	96	1205883714	110	1206490894	139	1206492460	105	1206494032	97
8	1205881958	139	1205883698	126	1206490893	140	1206492446	119	1206494007	122
9	1205881929	168	1205883668	156	1206490858	175	1206492439	126	1206493974	155
10	1205881927	170	1205883663	161	1206490856	177	1206492414	151	1206493970	159
11	1205881911	186	1205883656	168	1206490845	188	1206492405	160	1206493969	160
12	1205881910	187	1205883634	190	1206490832	201	1206492390	175	1206493956	173
13	1205881893	204	1205883633	191	1206490826	207	1206492381	184	1206493940	189
14	1205881872	225	1205883622	202	1206490823	210	1206492373	192	1206493939	190
15	1205881867	230	1205883618	206	1206490822	211	1206492371	194	1206493939	190
16	1205881863	234	1205883591	233	1206490821	212	1206492369	196	1206493895	234
17	1205881849	248	1205883589	235	1206490790	243	1206492330	235	1206493894	235
18	1205881836	261	1205883579	245	1206490788	245	1206492307	258	1206493882	247
19	1205881805	292	1205883539	285	1206490744	289	1206492292	273	1206493844	285
20	1205881799	298	1205883539	285	1206490740	293	1206492271	294	1206493829	300
21	1205881767	330	1205883521	303	1206490720	313	1206492260	305	1206493816	313
22	1205881749	348	1205883510	314	1206490655	378	1206492234	331	1206493808	321
23	1205881701	396	1205883461	363	1206490653	380	1206492209	356	1206493780	349
24	1205881670	427	1205883398	426	1206490589	444	1206492182	383	1206493728	401
25	1205881643	454	1205883397	427	1206490588	445	1206492133	432	1206493698	431
26	1205881587	510	1205883366	458	1206490557	476	1206492106	459	1206493672	457
27	1205881560	537	1205883339	485	1206490533	500	1206492081	484	1206493646	483
28	1205881556	541	1205883330	494	1206490526	507	1206492079	486	1206493639	490
29	1205881552	545	1205883330	494	1206490522	511	1206492078	487	1206493637	492
30	1205881551	546	1205883311	513	1206490513	520	1206492059	506	1206493627	502
31	1205881551	546	1205883295	529	1206490509	524	1206492058	507	1206493618	511
32	1205881531	566	1205883287	537	1206490506	527	1206492055	510	1206493590	539
33	1205881530	567	1205883283	541	1206490482	551	1206492023	542	1206493588	541
34	1205881509	588	1205883272	552	1206490472	561	1206491988	577	1206493576	553
35	1205881486	611	1205883226	598	1206490430	603	1206491982	583	1206493531	598
36	1205881481	616	1205883221	603	1206490422	611	1206491970	595	1206493512	617
37	1205881434	663	1205883209	615	1206490411	622	1206491948	617	1206493505	624
38	1205881412	685	1205883194	630	1206490365	668	1206491944	621	1206493492	637
39	1205881407	690	1205883171	653	1206490364	669	1206491921	644	1206493491	638
40	1205881397	700	1205883156	668	1206490347	686	1206491920	645	1206493487	642
41	1205881367	730	1205883152	672	1206490343	690	1206491888	677	1206493454	675
42	1205881360	737	1205883146	678	1206490338	695	1206491887	678	1206493451	678
43	1205881359	738	1205883142	682	1206490337	696	1206491886	679	1206493449	680
44	1205881355	742	1205883137	687	1206490330	703	1206491885	680	1206493444	685

46	1205881342	755	1205883129	695	1206490327	706	1206491876	689	1206493437	692
47	1205881340	757	1205883104	720	1206490325	708	1206491870	695	1206493416	713
48	1205881338	759	1205883090	734	1206490295	738	1206491842	723	1206493416	713
49	1205881301	796	1205883001	823	1206490204	829	1206491766	799	1206493311	818
50	1205881200	897	1205882999	825	1206490202	831	1206491742	823	1206493278	851
51	1205881171	926	1205882974	850	1206490177	856	1206491730	835	1206493271	858
52	1205881136	961	1205882936	888	1206490136	897	1206491687	878	1206493241	888
53	1205881133	964	1205882931	893	1206490131	902	1206491658	907	1206493238	891
54	1205881101	996	1205882891	933	1206490086	947	1206491642	923	1206493212	917
55	1205881094	1003	1205882890	934	1206490085	948	1206491641	924	1206493200	929
56	1205881091	1006	1205882889	935	1206490074	959	1206491632	933	1206493197	932
57	1205881091	1006	1205882876	948	1206490073	960	1206491627	938	1206493196	933
58	1205881080	1017	1205882875	949	1206490066	967	1206491614	951	1206493185	944
59	1205881074	1023	1205882872	952	1206490064	969	1206491609	956	1206493158	971
60	1205881041	1056	1205882839	985	1206490033	1000	1206491578	987	1206493152	977
61	1205881033	1064	1205882829	995	1206490023	1010	1206491572	993	1206493139	990
62	1205881033	1064	1205882821	1003	1206490020	1013	1206491563	1002	1206493130	999
63	1205881014	1083	1205882804	1020	1206490003	1030	1206491551	1014	1206493114	1015
64	1205880979	1118	1205882774	1050	1206489989	1044	1206491535	1030	1206493101	1028

Transcode

Condor

Condor Tra	nscode 1 Machine									
Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A
1	1202119429	0	1202120693	0	1202122155	0	1202123415	0	1202124657	0
2	1202119440	11	1202120704	11	1202122166	11	1202123426	11	1202124668	11
3	1202119453	24	1202120717	24	1202122179	24	1202123439	24	1202124681	24
4	1202119464	35	1202120728	35	1202122189	34	1202123450	35	1202124692	35
5	1202119476	47	1202120740	47	1202122202	47	1202123462	47	1202124704	47
6	1202119488	59	1202120752	59	1202122214	59	1202123474	59	1202124716	59
7	1202119498	69	1202120762	69	1202122224	69	1202123484	69	1202124726	69
8	1202119510	81	1202120774	81	1202122236	81	1202123496	81	1202124739	82
9	1202119522	93	1202120786	93	1202122248	93	1202123509	94	1202124751	94
10	1202119534	105	1202120798	105	1202122261	106	1202123521	106	1202124763	106
11	1202119545	116	1202120809	116	1202122271	116	1202123532	117	1202124775	118
12	1202119558	129	1202120822	129	1202122284	129	1202123545	130	1202124787	130
13	1202119569	140	1202120833	140	1202122294	139	1202123555	140	1202124797	140
14	1202119581	152	1202120845	152	1202122307	152	1202123568	153	1202124810	153
15	1202119594	165	1202120858	165	1202122320	165	1202123580	165	1202124823	166
16	1202119605	176	1202120868	175	1202122330	175	1202123590	175	1202124833	176
17	1202119617	188	1202120880	187	1202122343	188	1202123602	187	1202124845	188
18	1202119629	200	1202120892	199	1202122355	200	1202123614	199	1202124858	201
19	1202119641	212	1202120905	212	1202122367	212	1202123627	212	1202124870	213
20	1202119652	223	1202120916	223	1202122377	222	1202123637	222	1202124880	223
21	1202119664	235	1202120928	235	1202122389	234	1202123649	234	1202124893	236
22	1202119677	248	1202120941	248	1202122402	247	1202123662	247	1202124905	248
23	1202119687	258	1202120952	259	1202122412	257	1202123673	258	1202124916	259
24	1202119699	270	1202120964	271	1202122425	270	1202123685	270	1202124929	272
25	1202119712	283	1202120977	284	1202122438	283	1202123697	282	1202124941	284
26	1202119725	296	1202120990	297	1202122450	295	1202123710	295	1202124954	297
27	1202119735	306	1202121000	307	1202122461	306	1202123721	306	1202124965	308
28	1202119745	316	1202121011	318	1202122471	316	1202123731	316	1202124975	318
29	1202119758	329	1202121023	330	1202122483	328	1202123743	328	1202124987	330
30	1202119769	340	1202121034	341	1202122494	339	1202123754	339	1202124998	341
31	1202119781	352	1202121046	353	1202122506	351	1202123766	351	1202125011	354

84 85 86 87	1202120406 1202120416 1202120429 1202120441 1202120452	977 987 1000 1012 1023	1202121672 1202121683 1202121695 1202121707 1202121718	990 1002 1014 1025	1202123141 1202123153 1202123166 1202123176	986 998 1011 1021	1202124402 1202124414 1202124427 1202124437	987 999 1012 1022	1202125647 1202125660 1202125672 1202125683	990 1003 1015 1026
84 85 86	1202120416 1202120429	987 1000	1202121683 1202121695	990 1002	1202123153	998	1202124414	999	1202125660	1003
84 85	1202120416	987	1202121683	990						
84					1202123141	986	1202124402	987	1202125647	990
	1202120406	977	1202121672			l				
83			1202121772	979	1202123131	976	1202124391	976	1202125636	979
	1202120393	964	1202121660	967	1202123119	964	1202124378	963	1202125624	967
82	1202120381	952	1202121647	954	1202123107	952	1202124366	951	1202125611	954
81	1202120369	940	1202121635	942	1202123095	940	1202124354	939	1202125599	942
80	1202120359	930	1202121625	932	1202123085	930	1202124344	929	1202125589	932
79	1202120347	918	1202121613	920	1202123072	917	1202124331	916	1202125577	920
78	1202120334	905	1202121601	908	1202123060	905	1202124319	904	1202125564	907
77	1202120322	893	1202121588	895	1202123047	892	1202124307	892	1202125552	895
	1202120312	883	1202121578	885	1202123037	882	1202124297	882	1202125542	885
75	1202120299	870	1202121565	872	1202123025	870	1202124285	870	1202125530	873
74	1202120287	858	1202121553	860	1202123012	857	1202124272	857	1202125517	860
73	1202120276	847	1202121543	850	1202123002	847	1202124262	847	1202125507	850
72	1202120264	835	1202121530	837	1202122990	835	1202124250	835	1202125495	838
71	1202120252	823	1202121517	824	1202122978	823	1202124237	822	1202125483	826
70	1202120239	810	1202121505	812	1202122966	811	1202124225	810	1202125470	813
69	1202120229	800	1202121495	802	1202122956	801	1202124215	800	1202125460	803
68	1202120217	788	1202121482	789	1202122943	788	1202124202	787	1202125448	791
67	1202120204	775	1202121470	777	1202122931	776	1202124190	775	1202125435	778
66	1202120192	763	1202121457	764	1202122918	763	1202124177	762	1202125423	766
65	1202120182	753	1202121447	754	1202122908	753	1202124167	752	1202125413	756
	1202120170	741	1202121435	742	1202122896	741	1202124155	740	1202125400	743
	1202120158	729	1202121423	730	1202122884	729	1202124143	728	1202125388	731
	1202120146	717	1202121411	718	1202122871	716	1202124131	716	1202125376	719
	1202120133	704	1202121398	705	1202122859	704	1202124118	703	1202125363	706
	1202120121	692	1202121387	694	1202122847	692	1202124107	692	1202125352	695
	1202120111	682	1202121377	684	1202122837	682	1202124096	681	1202125341	684
	1202120087	670	1202121333	672	1202122812	670	1202124072	669	1202125317	672
	1202120074	658	1202121340	660	1202122812	657	1202124033	657	1202125317	660
	1202120074	645	1202121340	647	1202122800	645	1202124059	644	1202125304	647
	1202120065	636	1202121331	638	1202122770	636	1202124050	635	1202125295	638
	1202120052	623	1202121319	626	1202122778	623	1202124037	622	1202125283	626
	1202120040	611	1202121307	614	1202122767	612	1202124026	611	1202125271	614
	1202120030	601	1202121297	604	1202122757	602	1202124016	601	1202125261	604
	1202120018	589	1202121285	592	1202122744	589	1202124003	588	1202125249	592
50	1202120006	577	1202121272	579	1202122732	577	1202123991	576	1202125236	579
49	1202119993	564	1202121260	567	1202122719	564	1202123979	564	1202125224	567
48	1202119983	554	1202121250	557	1202122709	554	1202123969	554	1202125213	556
47	1202119970	541	1202121237	544	1202122697	542	1202123957	542	1202125201	544
46	1202119958	529	1202121224	531	1202122684	529	1202123944	529	1202125188	531
45	1202119945	516	1202121212	519	1202122672	517	1202123932	517	1202125176	519
44	1202119935	506	1202121202	509	1202122661	506	1202123921	506	1202125165	508
43	1202119922	493	1202121189	496	1202122649	494	1202123909	494	1202125153	496
42	1202119908	479	1202121175	482	1202122635	480	1202123894	479	1202125138	481
41	1202119897	468	1202121163	470	1202122623	468	1202123883	468	1202125127	470
40	1202119884	455	1202121150	457	1202122610	455	1202123870	455	1202125114	457
39	1202119873	444	1202121140	447	1202122599	444	1202123860	445	1202125103	446
38	1202119862	433	1202121128	435	1202122588	433	1202123848	433	1202125091	434
37	1202119852	423	1202121117	424	1202122578	423	1202123838	423	1202125081	424
	1202119840	411	1202121105	412	1202122566	411	1202123826	411	1202125069	412
	1202119828	399	1202121093	400	1202122553	398	1202123813	398	1202125057	400
	1202119816	387	1202121081	388	1202122542	387	1202123802	387	1202125045	388
	1202119804	375	1202121069	376	1202122529	374	1202123789	374	1202125033	376
	1202119791	362	1202121057	364	1202122517	362	1202123777	362	1202125021	364

91	1202120486	1057	1202121753	1060	1202123211	1056	1202124472	1057	1202125717	1060
92	1202120499	1070	1202121765	1072	1202123224	1069	1202124485	1070	1202125730	1073
93	1202120509	1080	1202121776	1083	1202123235	1080	1202124495	1080	1202125741	1084
94	1202120520	1091	1202121788	1095	1202123246	1091	1202124507	1092	1202125752	1095
95	1202120533	1104	1202121800	1107	1202123259	1104	1202124520	1105	1202125765	1108
96	1202120545	1116	1202121812	1119	1202123271	1116	1202124532	1117	1202125777	1120
97	1202120557	1128	1202121824	1131	1202123283	1128	1202124544	1129	1202125789	1132
98	1202120568	1139	1202121835	1142	1202123295	1140	1202124556	1141	1202125800	1143
99	1202120581	1152	1202121848	1155	1202123307	1152	1202124568	1153	1202125813	1156
100	1202120590	1161	1202121857	1164	1202123316	1161	1202124577	1162	1202125822	1165

Condor Transac	ode 2 Machines									
Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A
1	1202179053	0		0	1202179710	0	1202180075	0	1202180457	0
2	1202179055	12	1202179388	4	1202179710	12	1202180073	12	1202180437	12
3	1202179069	16	1202179393	5	1202179724	14	1202180089	14	1202180470	13
4	1202179073	20	1202179399	11	1202179724	18	1202180093	18	1202180475	18
5	1202179073	23	1202179399	15	1202179728	21	1202180093	22	1202180473	21
6	1202179078	25	1202179403	16	1202179731	23	1202180097	23	1202180478	23
7	1202179078			21		25		26		25
8	1202179079	30	1202179409	25	1202179735	29	1202180101	29	1202180482 1202180485	28
9	1202179083	34	1202179413	26	1202179739	32	1202180104	33	1202180489	32
	1202179087	37		29	1202179742	35		36	1202180489	35
10	1202179090	37	1202179417	32	1202179743		1202180111	37	1202180492	
11			1202179420			36	1202180112			36
12	1202179095	42	1202179424	36	1202179750	40	1202180117	42	1202180498	41
13	1202179099	46	1202179425	40	1202179754	44	1202180121	46	1202180501	44
15	1202179102	50	1202179428	45	1202179757	47	1202180124	49	1202180505	48
	1202179103		1202179433		1202179758		1202180124		1202180505	48
16	1202179105	52	1202179437	49	1202179761	51	1202180130	55	1202180511	54
17	1202179109	56	1202179440	52	1202179764	54	1202180134	59	1202180515	58
18	1202179113	60	1202179440	52	1202179767		1202180134			58
19	1202179116	63	1202179446	58	1202179770	60	1202180138	63	1202180519	62
20	1202179116	63	1202179450	62	1202179771	61	1202180142	67	1202180523	66
21	1202179120	67	1202179453	65	1202179774	64	1202180146	71	1202180526	69
22	1202179124	71	1202179453	65	1202179778	68	1202180146	71	1202180527	70
23	1202179127	74	1202179459	71	1202179782	72	1202180150	75	1202180530	73
24	1202179130	77	1202179463	75	1202179786	76	1202180153	78	1202180534	77
25	1202179131	78	1202179464	76	1202179786	76	1202180157	82	1202180537	80
26	1202179136	83	1202179467	79	1202179790	80	1202180160	85	1202180540	83
27	1202179140	87	1202179470	82	1202179794	84	1202180161	86	1202180541	84
28	1202179144	91	1202179474	86	1202179797	87	1202180166	91	1202180546	89
29	1202179144	91	1202179475	87	1202179798	88	1202180170	95	1202180549	92
30	1202179148	95	1202179478	90	1202179801	91	1202180170	95	1202180550	93
31	1202179152	99	1202179482	94	1202179805	95	1202180173	98	1202180552	95
32	1202179155	102	1202179485	97	1202179808	98	1202180178	103	1202180557	100
33	1202179158	105	1202179488	100	1202179811	101	1202180182	107	1202180561	104
34	1202179158	105	1202179489	101	1202179812	102	1202180182	107	1202180561	104
35	1202179163	110	1202179494	106	1202179817	107	1202180186	111	1202180565	108
36	1202179167	114	1202179497	109	1202179820	110	1202180189	114	1202180570	113
37	1202179170	117	1202179499	111	1202179822	112	1202180193	118	1202180574	117
38	1202179171	118	1202179501	113	1202179824	114	1202180196	121	1202180576	119
39	1202179175	122	1202179505	117	1202179828	118	1202180197	122	1202180577	120
40	1202179179	126	1202179509	121	1202179831	121	1202180202	127	1202180582	125
41	1202179181	128	1202179510	122	1202179833	123	1202180205	130	1202180586	129
42	1202179184	131	1202179512	124	1202179834	124	1202180207	132	1202180587	130
43	1202179187	134	1202179517	129	1202179838	128	1202180208	133	1202180590	133
44	1202179191	138	1202179521	133	1202179842	132	1202180212	137	1202180595	138
45	1202179194	141	1202179524	136	1202179845	135	1202180216	141	1202180599	142
46	1202179195	142	1202179526	138	1202179848	138	1202180219	144	1202180601	144
47	1202179198	145	1202179527	139	1202179849	139	1202180220	145	1202180603	146

	1202179328 1202179331 1202179334 1202179336 1202179337 1202179341 1202179345 1202179348	275 278 281 283 284 288 292 295	1202179662 1202179665 1202179669 1202179673 1202179676 1202179676 1202179681 1202179685	274 277 281 285 288 288 293 297 300	1202179982 1202179986 1202179990 1202179992 1202179994 1202179997 1202180001 1202180004 1202180005	272 276 280 282 284 287 291 294 295	1202180352 1202180355 1202180359 1202180360 1202180363 1202180367 1202180370 1202180373	277 280 284 285 288 292 295 298 299	1202180734 1202180737 1202180741 1202180741 1202180747 1202180751 1202180753 1202180754 1202180762	277 280 284 284 290 294 296 297 305
94 95 96 97 98	1202179328 1202179331 1202179334 1202179336 1202179337 1202179341 1202179345	275 278 281 283 284 288 292	1202179665 1202179669 1202179673 1202179676 1202179676 1202179681	277 281 285 288 288 293	1202179986 1202179990 1202179992 1202179994 1202179997 1202180001	276 280 282 284 287 291	1202180355 1202180359 1202180360 1202180363 1202180367 1202180370	280 284 285 288 292 295	1202180737 1202180741 1202180741 1202180747 1202180751 1202180753	280 284 284 290 294 296
94 95 96 97	1202179328 1202179331 1202179334 1202179336 1202179337 1202179341	275 278 281 283 284 288	1202179665 1202179669 1202179673 1202179676 1202179676	277 281 285 288 288	1202179986 1202179990 1202179992 1202179994 1202179997	276 280 282 284 287	1202180355 1202180359 1202180360 1202180363 1202180367	280 284 285 288 292	1202180737 1202180741 1202180741 1202180747 1202180751	280 284 284 290 294
94 95 96	1202179328 1202179331 1202179334 1202179336 1202179337	275 278 281 283 284	1202179665 1202179669 1202179673 1202179676	277 281 285 288	1202179986 1202179990 1202179992 1202179994	276 280 282 284	1202180355 1202180359 1202180360 1202180363	280 284 285 288	1202180737 1202180741 1202180741 1202180747	280 284 284 290
94 95	1202179328 1202179331 1202179334 1202179336	275 278 281 283	1202179665 1202179669 1202179673	277 281 285	1202179986 1202179990 1202179992	276 280 282	1202180355 1202180359 1202180360	280 284 285	1202180737 1202180741 1202180741	280 284 284
94	1202179328 1202179331 1202179334	275 278 281	1202179665 1202179669	277 281	1202179986 1202179990	276 280	1202180355 1202180359	280 284	1202180737 1202180741	280 284
	1202179328 1202179331	275 278	1202179665	277	1202179986	276	1202180355	280	1202180737	280
	1202179328	275								
92			1202170552	274	1202170002	272	1202100252	277		
91	1202179324	271	1202179662	274	1202179981	271	1202180350	275	1202180729	272
90	1202179321	268	1202179658	270	1202179979	269	1202180348	273	1202180726	269
89	1202179321	268	1202179651	263	1202179975	265	1202180344	269	1202180726	269
88	1202179317	264	1202179651	263	1202179970	260	1202180341	266	1202180722	265
87	1202179313	260	1202179648	260	1202179968	258	1202180339	264	1202180718	261
	1202179309	256	1202179645	257	1202179967	257	1202180337	262	1202180714	257
85	1202179308	255	1202179640	252	1202179964	254	1202180333	258	1202180713	256
84	1202179306	253	1202179639	251	1202179960	250	1202180329	254	1202180710	253
83	1202179302	249	1202179636	248	1202179956	246	1202180327	252	1202180706	249
82	1202179297	244	1202179632	244	1202179955	245	1202180326	251	1202180703	246
81	1202179294	241	1202179627	239	1202179952	242	1202180322	247	1202180700	243
80	1202179294	241	1202179627	239	1202179949	239	1202180319	244	1202180699	242
79	1202179291	238	1202179623	235	1202179944	234	1202180316	241	1202180695	238
78	1202179287	234	1202179620	232	1202179943	233	1202180315	240	1202180692	235
77	1202179283	230	1202179615	227	1202179940	230	1202180311	236	1202180688	231
76	1202179281	228	1202179612	224	1202179936	226	1202180305	230	1202180686	229
75	1202179280	227	1202179612	224	1202179932	222	1202180305	230	1202180685	228
74	1202179276	223	1202179608	220	1202179931	221	1202180302	227	1202180681	224
73	1202179273	220	1202179604	216	1202179928	218	1202180298	223	1202180678	221
72	1202179270	217	1202179600	212	1202179925	215	1202180294	219	1202180675	218
	1202179269	216	1202179599	211	1202179920	210	1202180293	218	1202180674	217
70	1202179266	213	1202179596	208	1202179919	209	1202180290	215	1202180670	213
	1202179261	208	1202179593	205	1202179916	206	1202180286	211	1202180666	209
68	1202179258	205	1202179588	200	1202179913	203	1202180283	208	1202180664	207
67	1202179258	205	1202179587	199	1202179908	198	1202180281	206	1202180663	206
66	1202179254	201	1202179584	196	1202179907	197	1202180279	204	1202180660	203
65	1202179250	197	1202179581	193	1202179904	194	1202180275	200	1202180655	198
	1202179249	196	1202179576	188	1202179900	190	1202180271	196	1202180654	197
63	1202179246	193	1202179575	187	1202179896	186	1202180269	194	1202180651	194
62	1202179242	189	1202179572	184	1202179895	185	1202180267	192	1202180647	190
61	1202179238	185	1202179568	180	1202179893	183	1202180264	189	1202180642	185
60	1202179235	182	1202179564	176	1202179890	180	1202180259	184	1202180640	183
59	1202179235	182	1202179563	175	1202179885	175	1202180258	183	1202180639	182
58	1202179231	178	1202179560	172	1202179881	171	1202180255	180	1202180636	179
57	1202179227	174	1202179557	169	1202179881	171	1202180252	177	1202180632	175
56	1202179223	170	1202179553	165	1202179877	167	1202180248	173	1202180628	171
55	1202179220	167	1202179550	162	1202179872	162	1202180244	169	1202180626	169
54	1202179220	167	1202179549	161	1202179872	162	1202180244	169	1202180625	168
53	1202179217	164	1202179545	157	1202179869	159	1202180240	165	1202180622	165
52	1202179213	160	1202179541	153	1202179865	155	1202180235	160	1202180618	161
51	1202179209	156	1202179539	151	1202179861	151	1202180231	156	1202180613	156
50	1202179208	155	1202179538	150	1202179860	150	1202180231	156	1202180613	156
49	1202179205	152	1202179535	147	1202179857	147	1202180227	152	1202180610	153
48	1202179202	149	1202179531	143	1202179853	143	1202180222	147	1202180606	149

Condor Transcode 4 Machines										
Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A
1	1202192644	0	1202193010	0	1202193269	0	1202193624	0	1202193901	0
2	1202192656	12	1202193015	5	1202193281	12	1202193629	5	1202193906	5
3	1202192658	14	1202193016	6	1202193288	19	1202193629	5	1202193907	6

5 6 7 8	1202192666 1202192668 1202192670	22 24 26	1202193020 1202193022	10	1202193293 1202193293	24	1202193632 1202193634	10	1202193913 1202193918	12
6 7 8	1202192670		-				1202173034	10	1202173710	17
7 8			1202193023	13	1202193297	28	1202193634	10	1202193918	17
8	1202192672	28	1202193024	14	1202193297	28	1202193640	16	1202193923	22
	1202192672	28	1202193027	17	1202193300	31	1202193642	18	1202193926	25
9	1202192674	30	1202193031	21	1202193303	34	1202193644	20	1202193927	26
	1202192680	36	1202193035	25	1202193307	38	1202193647	23	1202193931	30
	1202192681	37	1202193036	26	1202193309	40	1202193648	24	1202193932	31
	1202192684	40	1202193038	28	1202193310	41	1202193650	26	1202193933	32
	1202192684	40	1202193040	30	1202193313	44	1202193652	28	1202193937	36
	1202192687	43	1202193041	31	1202193316	47	1202193655	31	1202193940	39
	1202192690	46	1202193045	35	1202193317	48	1202193656	32	1202193941	40
	1202192694	50	1202193049	39	1202193317	51	1202193661	37	1202193945	44
	1202192695	51	1202193050	40	1202193321	52	1202193662	38	1202193945	44
	1202192695	51	1202193052	42	1202193321	53	1202193662	38	1202193945	44
	1202192698	54	1202193053	43	1202193328	59	1202193669	45	1202193952	51
	1202192700	56	1202193055	45	1202193329	60	1202193669	45	1202193954	53
	1202192704	60	1202193058	48	1202193321	62	1202193673	49	1202193955	54
	1202192704	64	1202193058	52	1202193331	63	1202193673	49	1202193955	55
				55				-		59
	1202192709	65	1202193065	56	1202193335	66	1202193678	54	1202193960	59
	1202192712		1202193066		1202193338		1202193680		1202193960	
	1202192712	68	1202193067	57	1202193343	74	1202193683	59	1202193964	63
		70	1202193071	61	1202193343	74	1202193684	60	1202193967	66
	1202192717	73	1202193071	61	1202193345	76	1202193686	62	1202193968	67
	1202192721	77	1202193076	66	1202193347	78	1202193688	64	1202193971	70
	1202192723	79	1202193076	66	1202193348	79	1202193694	70	1202193972	71
	1202192724	80	1202193082	72	1202193352	83	1202193694	70	1202193973	72
	1202192727	83	1202193083	73	1202193357	88	1202193697	73	1202193977	76
	1202192728	84	1202193083	73	1202193357	88	1202193700	76	1202193979	78
	1202192730	86	1202193088	78	1202193360	91	1202193701	77	1202193981	80
	1202192732	88	1202193089	79	1202193363	94	1202193704	80	1202193981	80
	1202192735	91	1202193093	83	1202193364	95	1202193707	83	1202193984	83
	1202192738	94	1202193095	85	1202193367	98	1202193708	84	1202193988	87
	1202192740	96	1202193096	86	1202193369	100	1202193710	86	1202193989	88
	1202192742	98	1202193097	87	1202193371	102	1202193712	88	1202193992	91
	1202192744	100	1202193101	91	1202193373	104	1202193716	92	1202193993	92
	1202192744	100	1202193104	94	1202193374	105	1202193717	93	1202193994	93
	1202192748	104	1202193108	98	1202193377	108	1202193719	95	1202193999	98
	1202192749	105	1202193108	98	1202193379	110	1202193720	96	1202193999	98
	1202192751	107	1202193111	101	1202193383	114	1202193724	100	1202194003	102
	1202192753	109	1202193112	102	1202193383	114	1202193724	100	1202194004	103
	1202192756	112	1202193117	107	1202193387	118	1202193728	104	1202194007	106
	1202192760	116	1202193118	108	1202193388	119	1202193731	107	1202194008	107
	1202192760	116	1202193123	113	1202193389	120	1202193732	108	1202194011	110
	1202192764	120	1202193124	114	1202193395	126	1202193734	110	1202194014	113
	1202192765	121	1202193126	116	1202193397	128	1202193735	111	1202194015	114
	1202192766	122	1202193127	117	1202193397	128	1202193738	114	1202194018	117
	1202192770	126	1202193128	118	1202193400	131	1202193739	115	1202194019	118
	1202192773	129	1202193133	123	1202193403	134	1202193745	121	1202194021	120
	1202192774	130	1202193136	126	1202193405	136	1202193747	123	1202194024	123
	1202192777	133	1202193138	128	1202193409	140	1202193749	125	1202194027	126
	1202192778	134	1202193139	129	1202193409	140	1202193749	125	1202194030	129
	1202192781	137	1202193140	130	1202193412	143	1202193752	128	1202194031	130
	1202192782	138	1202193145	135	1202193414	145	1202193753	129	1202194033	132
	1202192783	139	1202193147	137	1202193415	146	1202193759	135	1202194034	133
	1202192787	143	1202193151	141	1202193420	151	1202193760	136	1202194040	139
	1202192791	147	1202193151	141	1202193422	153	1202193763	139	1202194041	140
	1202192791	147	1202193152	142	1202193423	154	1202193763	139	1202194043	142
62	1202192792	148	1202193157	147	1202193424	155	1202193764	140	1202194044	143

	63	1202192794	150	1202193157	147	1202193428	159	1202193767	143	1202194046	145
65											
66 1202192805 161 1202193165 155 1202193436 167 1202193777 153 1202194055 154 67 1202192807 161 1202193173 161 1202193437 168 1202193778 154 1202194066 155 68 1202192811 167 1202193172 162 1202193437 168 1202193787 154 1202194067 156 69 1202192811 167 1202193172 162 1202193445 176 1202193782 158 1202194060 159 70 1202192812 168 1202193176 166 1202193445 176 1202193787 163 1202194060 159 71 1202192813 171 1202193180 170 1202193449 180 1202193788 164 1202194066 165 72 1202192813 171 1202193181 173 1202193449 180 1202193788 164 1202194066 165 73 1202192821 177 1202193183 173 1202193449 180 1202193790 166 1202194061 170 74 1202192823 181 1202193184 174 1202193454 185 1202193790 166 1202194074 173 75 1202192823 181 1202193188 178 1202193456 187 1202193790 171 1202194074 173 76 1202192823 181 1202193189 179 1202193458 189 1202193800 176 1202194075 174 77 1202192823 188 1202193192 182 1202193462 193 1202193804 180 1202194075 174 78 1202192832 188 1202193192 182 1202193462 193 1202193804 180 120219405 184 80 1202192834 195 1202193304 194 1202193469 200 1202193808 184 1202194087 186 81 1202192840 196 1202193304 194 1202193473 204 1202193808 184 1202194087 186 82 1202192840 196 1202193304 194 1202193473 204 1202193805 184 120219400 190 83 1202192841 196 1202193304 194 1202193473 204 1202193808 184 120219409 190 84 1202192841 196 1202193304 194 1202193482 213 1202193805 195 1202193407 207 1202193807 208 1202193807 208 120219401 209 85 1202192851 207 1202193206 196 1202193482 213 1202193807 208 120219410 209 85 1202192854 207 1202193206 196 1202193487									-		-
1202192807 163	66	1202192805	161	1202193165	155	1202193436	167	1202193777	153	1202194055	154
	67	1202192805	161	1202193168	158	1202193437	168	1202193778	154	1202194056	155
	68	1202192807	163	1202193171	161	1202193439	170	1202193778	154	1202194057	156
	69	1202192811	167	1202193172	162	1202193442	173	1202193782	158	1202194060	159
	70	1202192812	168	1202193176	166	1202193445	176	1202193787	163	1202194061	160
	71	1202192815	171	1202193178	168	1202193448	179	1202193788	164	1202194066	165
74 1202192823 179 1202193184 174 1202193454 185 1202193795 171 1202194074 173 75 1202192825 181 1202193188 178 1202193456 187 1202193796 172 1202194074 173 76 1202192825 181 1202193189 179 1202193458 189 1202193800 176 1202194075 174 77 1202192829 185 1202193192 182 1202193462 193 1202193800 180 1202194079 178 78 1202193323 188 1202193192 182 1202193462 193 1202193804 180 1202194079 178 79 1202193361 192 1202193196 186 1202193460 200 1202193808 184 1202194087 186 81 120219337 193 1202193196 186 1202193473 204 1202193808 184 1202194087 186 81 120219240	72	1202192815	171	1202193180	170	1202193449	180	1202193788	164	1202194068	167
75 1202192825 181 1202193188 178 1202193456 187 1202193796 172 1202194074 173 76 1202192825 181 1202193189 179 1202193458 189 1202193800 176 1202194075 174 77 1202192829 185 1202193192 182 1202193462 193 1202193802 178 1202194079 178 78 1202192836 192 120193194 184 1202193462 193 1202193804 180 1202194085 184 80 1202193361 192 1202193194 184 1202193469 200 1202193808 184 1202194087 186 81 120219337 193 1202193409 186 1202193469 200 1202193808 184 1202194087 186 81 1202192840 196 1202193204 194 1202193473 204 1202193812 188 1202194099 188 82 1202192840	73	1202192821	177	1202193183	173	1202193451	182	1202193790	166	1202194071	170
76 1202192825 181 1202193189 179 1202193488 189 1202193800 176 1202194075 174 77 1202192829 185 1202193192 182 1202193462 193 1202193802 178 1202194079 178 78 1202192832 188 1202193192 182 1202193462 193 1202193804 180 1202194079 178 79 1202192836 192 1202193194 184 1202193462 193 1202193804 180 1202194085 184 80 1202192837 193 1202193196 186 1202193469 200 1202193808 184 1202194087 186 81 1202192840 196 1202193202 192 1202193473 204 1202193812 188 1202194089 188 82 1202192840 196 1202193205 195 1202193474 205 1202193812 188 1202194091 190 84 1202192840	74	1202192823	179	1202193184	174	1202193454	185	1202193795	171	1202194074	173
77 1202192829 185 1202193192 182 1202193462 193 1202193802 178 1202194079 178 78 1202192832 188 1202193192 182 1202193462 193 1202193804 180 1202194079 178 79 1202192836 192 1202193194 184 1202193462 193 1202193804 180 1202194085 184 80 1202192837 193 1202193196 186 1202193469 200 1202193808 184 1202194087 186 81 1202192839 195 1202193202 192 1202193478 204 1202193808 184 1202194089 188 82 1202192840 196 1202193204 194 1202193474 205 1202193812 188 1202194091 190 84 1202192840 196 1202193205 195 1202193474 205 1202193812 188 1202194091 190 85 1202192847	75	1202192825	181	1202193188	178	1202193456	187	1202193796	172	1202194074	173
78 1202192832 188 1202193192 182 1202193462 193 1202193804 180 1202194079 178 79 1202192836 192 1202193194 184 1202193462 193 1202193804 180 1202194085 184 80 1202192837 193 1202193196 186 1202193469 200 1202193808 184 1202194087 186 81 1202192840 196 1202193202 192 1202193473 204 1202193808 184 1202194089 188 82 1202192840 196 1202193205 195 1202193473 204 1202193812 188 1202194091 190 83 1202192840 196 1202193206 196 1202193474 205 1202193816 192 1202194092 191 84 1202192851 207 1202193209 199 1202193478 209 1202193816 192 1202194092 191 86 1202192851	76	1202192825	181	1202193189	179	1202193458	189	1202193800	176	1202194075	174
79 1202192836 192 1202193194 184 1202193462 193 1202193804 180 1202194085 184 80 1202192837 193 1202193196 186 1202193469 200 1202193808 184 1202194087 186 81 1202192839 195 1202193202 192 1202193469 200 1202193808 184 1202194089 188 82 1202192840 196 1202193204 194 1202193473 204 1202193812 188 1202194091 190 83 1202192840 196 1202193205 195 1202193474 205 1202193812 188 1202194091 190 84 1202192847 203 1202193206 196 1202193476 207 1202193816 192 1202194092 191 85 1202192851 207 1202193209 199 1202193478 209 1202193819 195 1202194100 199 86 1202192851	77	1202192829	185	1202193192	182	1202193462	193	1202193802	178	1202194079	178
80 1202192837 193 1202193196 186 1202193469 200 1202193808 184 1202194087 186 81 1202192839 195 1202193202 192 1202193469 200 1202193808 184 1202194089 188 82 1202192840 196 1202193205 195 1202193473 204 1202193812 188 1202194091 190 83 1202192840 196 1202193205 195 1202193478 205 1202193812 188 1202194091 190 84 1202192847 203 1202193206 196 1202193478 207 1202193816 192 1202194092 191 85 1202192851 207 120219320 200 1202193478 209 1202193819 195 1202194100 199 86 1202192852 208 1202193215 205 1202193482 213 1202193820 196 1202194101 200 87 1202192852	78	1202192832	188	1202193192	182	1202193462	193	1202193804	180	1202194079	178
81 1202192839 195 1202193202 192 1202193469 200 1202193808 184 1202194089 188 82 1202192840 196 1202193204 194 1202193473 204 1202193812 188 1202194091 190 83 1202192840 196 1202193205 195 1202193474 205 1202193812 188 1202194091 190 84 1202192847 203 1202193206 196 1202193476 207 1202193816 192 1202194092 191 85 1202192851 207 1202193209 199 1202193478 209 1202193819 195 1202194100 199 86 1202192851 207 1202193210 200 1202193482 213 1202193820 196 1202194101 200 87 1202192852 208 1202193217 207 1202193482 213 1202193822 201 1202194103 202 88 1202192855	79	1202192836	192	1202193194	184	1202193462	193	1202193804	180	1202194085	184
82 1202192840 196 1202193204 194 1202193473 204 1202193812 188 1202194091 190 83 1202192840 196 1202193205 195 1202193474 205 1202193812 188 1202194091 190 84 1202192847 203 1202193206 196 1202193476 207 1202193816 192 1202194092 191 85 1202192851 207 1202193209 199 1202193478 209 1202193819 195 1202194100 199 86 1202192851 207 1202193210 200 1202193482 213 1202193820 196 1202194101 200 87 1202192852 208 1202193215 205 1202193482 213 1202193821 197 1202194103 202 88 1202192854 210 1202193217 207 1202193487 218 1202193825 201 1202194103 204 89 1202192855	80	1202192837	193	1202193196	186	1202193469	200	1202193808	184	1202194087	186
83 1202192840 196 1202193205 195 1202193474 205 1202193812 188 1202194091 190 84 1202192847 203 1202193206 196 1202193476 207 1202193816 192 1202194092 191 85 1202192851 207 1202193209 199 1202193478 209 1202193819 195 1202194100 199 86 1202192851 207 1202193210 200 1202193482 213 1202193820 196 1202194101 200 87 1202192852 208 1202193215 205 1202193482 213 1202193821 197 1202194103 202 88 1202192854 210 1202193217 207 1202193487 218 1202193825 201 1202194105 204 89 1202192860 216 1202193218 208 1202193488 219 1202193822 203 1202194107 206 90 1202192860	81	1202192839	195	1202193202	192	1202193469	200	1202193808	184	1202194089	188
84 1202192847 203 1202193206 196 1202193476 207 1202193816 192 1202194092 191 85 1202192851 207 1202193209 199 1202193478 209 1202193819 195 1202194100 199 86 1202192851 207 1202193210 200 1202193482 213 1202193820 196 1202194101 200 87 1202192852 208 1202193215 205 1202193482 213 1202193821 197 1202194103 202 88 1202192854 210 1202193217 207 1202193487 218 1202193825 201 1202194105 204 89 1202192855 211 1202193218 208 1202193488 219 1202193827 203 1202194107 206 90 1202192860 216 1202193219 209 1202193488 219 1202193828 204 1202194108 207 91 1202192864	82	1202192840	196	1202193204	194	1202193473	204	1202193812	188	1202194091	190
85 1202192851 207 1202193209 199 1202193478 209 1202193819 195 1202194100 199 86 1202192851 207 1202193210 200 1202193482 213 1202193820 196 1202194101 200 87 1202192852 208 1202193215 205 1202193482 213 1202193821 197 1202194103 202 88 1202192854 210 1202193217 207 1202193487 218 1202193825 201 1202194105 204 89 1202192855 211 1202193218 208 1202193488 219 1202193827 203 1202194107 206 90 1202192860 216 1202193219 209 1202193488 219 1202193828 204 1202194108 207 91 1202192864 220 1202193223 213 1202193494 225 1202193832 208 1202194114 213 92 1202192865	83	1202192840	196	1202193205	195	1202193474	205	1202193812	188	1202194091	190
86 1202192851 207 1202193210 200 1202193482 213 1202193820 196 1202194101 200 87 1202192852 208 1202193215 205 1202193482 213 1202193821 197 1202194103 202 88 1202192854 210 1202193217 207 1202193487 218 1202193825 201 1202194105 204 89 1202192855 211 1202193218 208 1202193488 219 1202193827 203 1202194107 206 90 1202192860 216 1202193219 209 1202193488 219 1202193828 204 1202194108 207 91 1202192864 220 1202193223 213 1202193494 225 1202193832 208 1202194114 213 92 1202192865 221 1202193224 214 1202193495 226 1202193837 213 1202194115 214 93 1202192870	84	1202192847	203	1202193206	196	1202193476	207	1202193816	192	1202194092	191
87 1202192852 208 1202193215 205 1202193482 213 1202193821 197 1202194103 202 88 1202192854 210 1202193217 207 1202193487 218 1202193825 201 1202194105 204 89 1202192855 211 1202193218 208 1202193488 219 1202193827 203 1202194107 206 90 1202192860 216 1202193219 209 1202193488 219 1202193828 204 1202194108 207 91 1202192864 220 1202193223 213 1202193494 225 1202193832 208 1202194114 213 92 1202192865 221 1202193224 214 1202193495 226 1202193837 213 1202194115 214 93 1202192870 226 1202193230 220 1202193500 231 1202193837 213 1202194119 218 95 1202192870	85	1202192851	207	1202193209	199	1202193478	209	1202193819	195	1202194100	199
88 1202192854 210 1202193217 207 1202193487 218 1202193825 201 1202194105 204 89 1202192855 211 1202193218 208 1202193488 219 1202193827 203 1202194107 206 90 1202192860 216 1202193219 209 1202193488 219 1202193828 204 1202194108 207 91 1202192864 220 1202193223 213 1202193494 225 1202193832 208 1202194114 213 92 1202192865 221 1202193224 214 1202193495 226 1202193832 208 1202194115 214 93 1202192869 225 1202193228 218 1202193500 231 1202193837 213 1202194119 218 94 1202192870 226 1202193230 220 1202193500 231 1202193837 213 1202194120 219 96 1202192875	86	1202192851	207	1202193210	200	1202193482	213	1202193820	196	1202194101	200
89 1202192855 211 1202193218 208 1202193488 219 1202193827 203 1202194107 206 90 1202192860 216 1202193219 209 1202193488 219 1202193828 204 1202194108 207 91 1202192864 220 1202193223 213 1202193494 225 1202193832 208 1202194114 213 92 1202192865 221 1202193224 214 1202193495 226 1202193832 208 1202194115 214 93 1202192869 225 1202193228 218 1202193500 231 1202193837 213 1202194119 218 94 1202192870 226 1202193230 220 1202193500 231 1202193837 213 1202194120 219 95 1202192870 226 1202193230 220 1202193501 232 1202193838 214 1202194120 219 96 1202192875	87	1202192852	208	1202193215	205	1202193482	213	1202193821	197	1202194103	202
90	88	1202192854	210	1202193217	207	1202193487	218	1202193825	201	1202194105	204
91 1202192864 220 1202193223 213 1202193494 225 1202193832 208 1202194114 213 92 1202192865 221 1202193224 214 1202193495 226 1202193832 208 1202194115 214 93 1202192869 225 1202193228 218 1202193500 231 1202193837 213 1202194119 218 94 1202192870 226 1202193230 220 1202193500 231 1202193837 213 1202194120 219 95 1202192870 226 1202193230 220 1202193501 232 1202193838 214 1202194120 219 96 1202192875 231 1202193232 222 1202193505 236 1202193843 219 1202194123 222 97 1202192878 234 1202193237 227 1202193508 239 1202193846 222 1202194128 227 98 1202192879	89	1202192855	211	1202193218	208	1202193488	219	1202193827	203	1202194107	206
92 1202192865 221 1202193224 214 1202193495 226 1202193832 208 1202194115 214 93 1202192869 225 1202193228 218 1202193500 231 1202193837 213 1202194119 218 94 1202192870 226 1202193230 220 1202193500 231 1202193837 213 1202194120 219 95 1202192870 226 1202193230 220 1202193501 232 1202193838 214 1202194120 219 96 1202192875 231 1202193232 222 1202193505 236 1202193843 219 1202194123 222 97 1202192878 234 1202193237 227 1202193508 239 1202193846 222 1202194128 227 98 1202192879 235 1202193238 228 1202193509 240 1202193847 223 1202194131 230 99 1202192882	90	1202192860	216	1202193219	209	1202193488	219	1202193828	204	1202194108	207
93 1202192869 225 1202193228 218 1202193500 231 1202193837 213 1202194119 218 94 1202192870 226 1202193230 220 1202193500 231 1202193837 213 1202194120 219 95 1202192870 226 1202193230 220 1202193501 232 1202193838 214 1202194120 219 96 1202192875 231 1202193232 222 1202193505 236 1202193843 219 1202194123 222 97 1202192878 234 1202193237 227 1202193508 239 1202193846 222 1202194128 227 98 1202192879 235 1202193238 228 1202193509 240 1202193847 223 1202194131 230 99 1202192882 238 1202193241 231 1202193512 243 1202193850 226 1202194133 232	91	1202192864	220	1202193223	213	1202193494	225	1202193832	208	1202194114	213
94 1202192870 226 1202193230 220 1202193500 231 1202193837 213 1202194120 219 95 1202192870 226 1202193230 220 1202193501 232 1202193838 214 1202194120 219 96 1202192875 231 1202193232 222 1202193505 236 1202193843 219 1202194123 222 97 1202192878 234 1202193237 227 1202193508 239 1202193846 222 1202194128 227 98 1202192879 235 1202193238 228 1202193509 240 1202193847 223 1202194131 230 99 1202192882 238 1202193241 231 1202193512 243 1202193850 226 1202194133 232	92	1202192865	221	1202193224	214	1202193495	226	1202193832	208	1202194115	214
95 1202192870 226 1202193230 220 1202193501 232 1202193838 214 1202194120 219 96 1202192875 231 1202193232 222 1202193505 236 1202193843 219 1202194123 222 97 1202192878 234 1202193237 227 1202193508 239 1202193846 222 1202194128 227 98 1202192879 235 1202193238 228 1202193509 240 1202193847 223 1202194131 230 99 1202192882 238 1202193241 231 1202193512 243 1202193850 226 1202194133 232	93	1202192869	225	1202193228	218	1202193500	231	1202193837	213	1202194119	218
96 1202192875 231 1202193232 222 1202193505 236 1202193843 219 1202194123 222 97 1202192878 234 1202193237 227 1202193508 239 1202193846 222 1202194128 227 98 1202192879 235 1202193238 228 1202193509 240 1202193847 223 1202194131 230 99 1202192882 238 1202193241 231 1202193512 243 1202193850 226 1202194133 232	94	1202192870	226	1202193230	220	1202193500	231	1202193837	213	1202194120	219
97 1202192878 234 1202193237 227 1202193508 239 1202193846 222 1202194128 227 98 1202192879 235 1202193238 228 1202193509 240 1202193847 223 1202194131 230 99 1202192882 238 1202193241 231 1202193512 243 1202193850 226 1202194133 232	95	1202192870	226	1202193230	220	1202193501	232	1202193838	214	1202194120	219
98 1202192879 235 1202193238 228 1202193509 240 1202193847 223 1202194131 230 99 1202192882 238 1202193241 231 1202193512 243 1202193850 226 1202194133 232	96	1202192875	231	1202193232	222	1202193505	236	1202193843	219	1202194123	222
99 1202192882 238 1202193241 231 1202193512 243 1202193850 226 1202194133 232	97	1202192878	234	1202193237	227	1202193508	239	1202193846	222	1202194128	227
	98	1202192879	235	1202193238	228	1202193509	240	1202193847	223	1202194131	230
100 1202192885 241 1202193246 236 1202193518 249 1202193853 229 1202194139 238	99	1202192882	238	1202193241	231	1202193512	243	1202193850	226	1202194133	232
	100	1202192885	241	1202193246	236	1202193518	249	1202193853	229	1202194139	238

Condor Tra	inscode 8 Machines									
Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A
1	1202201411	0	1202201643	0	1202201891	0	1202202129	0	1202202367	0
2	1202201424	13	1202201646	3	1202201894	3	1202202142	13	1202202380	13
3	1202201431	20	1202201657	14	1202201905	14	1202202150	21	1202202381	14
4	1202201434	23	1202201659	16	1202201907	16	1202202152	23	1202202392	25
5	1202201435	24	1202201664	21	1202201909	18	1202202153	24	1202202392	25
6	1202201437	26	1202201666	23	1202201914	23	1202202154	25	1202202394	27
7	1202201437	26	1202201667	24	1202201915	24	1202202157	28	1202202395	28
8	1202201439	28	1202201668	25	1202201915	24	1202202158	29	1202202396	29
9	1202201442	31	1202201669	26	1202201921	30	1202202163	34	1202202400	33
10	1202201444	33	1202201672	29	1202201922	31	1202202164	35	1202202401	34
11	1202201446	35	1202201675	32	1202201924	33	1202202167	38	1202202403	36
12	1202201449	38	1202201676	33	1202201924	33	1202202169	40	1202202404	37
13	1202201450	39	1202201680	37	1202201928	37	1202202169	40	1202202407	40
14	1202201450	39	1202201681	38	1202201929	38	1202202170	41	1202202408	41
15	1202201455	44	1202201683	40	1202201932	41	1202202175	46	1202202411	44
16	1202201456	45	1202201684	41	1202201933	42	1202202177	48	1202202413	46
17	1202201457	46	1202201685	42	1202201934	43	1202202177	48	1202202413	46
18	1202201460	49	1202201690	47	1202201937	46	1202202180	51	1202202417	50
19	1202201463	52	1202201690	47	1202201940	49	1202202182	53	1202202420	53

20	1202201465	54	1202201693	50	1202201943	52	1202202184	55	1202202421	54
						52				
21	1202201466	55	1202201694	51	1202201943		1202202187	58	1202202424	57
22	1202201467	56	1202201697	54	1202201947	56	1202202188	59	1202202425	58
23	1202201471	60	1202201699	56	1202201948	57	1202202190	61	1202202427	60
24	1202201471	60	1202201699	56	1202201949	58	1202202191	62	1202202428	61
25	1202201474	63	1202201703	60	1202201953	62	1202202194	65	1202202431	64
26	1202201475	64	1202201705	62	1202201954	63	1202202196	67	1202202432	65
27	1202201478	67	1202201707	64	1202201955	64	1202202199	70	1202202436	69
28	1202201479	68	1202201708	65	1202201957	66	1202202200	71	1202202436	69
29	1202201481	70	1202201709	66	1202201958	67	1202202202	73	1202202437	70
30	1202201485	74	1202201713	70	1202201963	72	1202202205	76	1202202441	74
31	1202201485	74	1202201714	71	1202201964	73	1202202206	77	1202202441	74
32	1202201488	77	1202201717	74	1202201966	75	1202202209	80	1202202445	78
33	1202201489	78	1202201718	75	1202201966	75	1202202210	81	1202202446	79
34	1202201491	80	1202201720	77	1202201970	79	1202202212	83	1202202449	82
35	1202201495	84	1202201723	80	1202201971	80	1202202216	87	1202202451	84
36	1202201495	84	1202201724	81	1202201973	82	1202202216	87	1202202453	86
37	1202201498	87	1202201728	85	1202201976	85	1202202220	91	1202202456	89
38	1202201500	89	1202201729	86	1202201978	87	1202202221	92	1202202458	91
39	1202201500	89	1202201729	86	1202201978	87	1202202221	92	1202202461	94
40	1202201505	94	1202201732	89	1202201983	92	1202202224	95	1202202462	95
41	1202201506	95	1202201735	92	1202201984	93	1202202225	96	1202202462	95
42	1202201509	98	1202201736	93	1202201986	95	1202202230	101	1202202467	100
43	1202201510	99	1202201737	94	1202201988	97	1202202230	101	1202202467	100
44	1202201514	103	1202201742	99	1202201992	101	1202202233	104	1202202472	105
45	1202201515	104	1202201743	100	1202201994	103	1202202236	107	1202202473	106
46	1202201516	105	1202201745	102	1202201994	103	1202202236	107	1202202473	106
47	1202201519	108	1202201747	104	1202201997	106	1202202230	112	1202202476	109
48	1202201517	111	1202201747	107	1202202000	109	1202202241	112	1202202470	112
49	1202201522	111	1202201750	108	1202202000	110	1202202241	113	1202202479	113
	1202201523				1202202001	113	1202202242	114		
50	1202201526	112	1202201751	108		114			1202202480	113
51		115	1202201754	111	1202202005		1202202246	117		117
52	1202201530	119	1202201757	114	1202202008	117	1202202250	121	1202202487	120
53	1202201531	120	1202201759	116	1202202008	117	1202202250	121	1202202489	122
54	1202201531	120	1202201759	116	1202202009	118	1202202250	121	1202202490	123
55	1202201535	124	1202201764	121	1202202012	121	1202202255	126	1202202493	126
56	1202201536	125	1202201765	122	1202202015	124	1202202257	128	1202202494	127
57	1202201537	126	1202201765	122	1202202015	124	1202202259	130	1202202495	128
58	1202201541	130	1202201770	127	1202202019	128	1202202261	132	1202202499	132
59	1202201543	132	1202201772	129	1202202022	131	1202202263	134	1202202501	134
60	1202201544	133	1202201773	130	1202202023	132	1202202264	135	1202202502	135
61	1202201546	135	1202201775	132	1202202023	132	1202202266	137	1202202503	136
62	1202201549	138	1202201777	134	1202202026	135	1202202269	140	1202202507	140
63	1202201550	139	1202201780	137	1202202031	140	1202202272	143	1202202508	141
64	1202201554	143	1202201783	140	1202202034	143	1202202275	146	1202202513	146
65	1202201555	144	1202201786	143	1202202034	143	1202202276	147	1202202513	146
66	1202201557	146	1202201786	143	1202202037	146	1202202277	148	1202202516	149
67	1202201560	149	1202201787	144	1202202037	146	1202202279	150	1202202517	150
68	1202201560	149	1202201791	148	1202202041	150	1202202281	152	1202202518	151
69	1202201563	152	1202201793	150	1202202041	150	1202202284	155	1202202521	154
70	1202201565	154	1202201795	152	1202202045	154	1202202287	158	1202202523	156
71	1202201566	155	1202201799	156	1202202045	154	1202202290	161	1202202525	158
72	1202201567	156	1202201800	157	1202202049	158	1202202290	161	1202202525	158
73	1202201571	160	1202201802	159	1202202050	159	1202202293	164	1202202530	163
74	1202201574	163	1202201804	161	1202202054	163	1202202294	165	1202202531	164
75	1202201574	163	1202201804	161	1202202054	163	1202202295	166	1202202531	164
76	1202201579	168	1202201807	164	1202202059	168	1202202297	168	1202202535	168
77	1202201579	168	1202201810	167	1202202060	169	1202202301	172	1202202536	169
78	1202201580	169	1202201811	168	1202202060	169	1202202301	172	1202202537	170
		<u> </u>		<u> </u>		1		<u> </u>	1	1

79	1202201585	174	1202201814	171	1202202063	172	1202202305	176	1202202542	175
80	1202201586	175	1202201817	174	1202202066	175	1202202308	179	1202202544	177
81	1202201587	176	1202201818	175	1202202069	178	1202202309	180	1202202545	178
82	1202201588	177	1202201819	176	1202202071	180	1202202310	181	1202202545	178
83	1202201591	180	1202201822	179	1202202072	181	1202202313	184	1202202549	182
84	1202201595	184	1202201825	182	1202202074	183	1202202315	186	1202202553	186
85	1202201595	184	1202201825	182	1202202076	185	1202202316	187	1202202553	186
86	1202201597	186	1202201825	182	1202202078	187	1202202316	187	1202202557	190
87	1202201601	190	1202201830	187	1202202079	188	1202202321	192	1202202558	191
88	1202201603	192	1202201832	189	1202202082	191	1202202324	195	1202202561	194
89	1202201603	192	1202201833	190	1202202082	191	1202202325	196	1202202561	194
90	1202201606	195	1202201835	192	1202202086	195	1202202327	198	1202202563	196
91	1202201607	196	1202201836	193	1202202090	199	1202202327	198	1202202566	199
92	1202201610	199	1202201840	197	1202202090	199	1202202332	203	1202202568	201
93	1202201612	201	1202201841	198	1202202093	202	1202202333	204	1202202570	203
94	1202201612	201	1202201843	200	1202202093	202	1202202335	206	1202202571	204
95	1202201617	206	1202201845	202	1202202095	204	1202202336	207	1202202574	207
96	1202201619	208	1202201849	206	1202202098	207	1202202339	210	1202202576	209
97	1202201621	210	1202201849	206	1202202100	209	1202202342	213	1202202579	212
98	1202201623	212	1202201852	209	1202202103	212	1202202345	216	1202202581	214
99	1202201624	213	1202201853	210	1202202105	214	1202202346	217	1202202582	215
100	1202201625	214	1202201857	214	1202202105	214	1202202349	220	1202202583	216

Count Run I Run I A Run 2 Run 2 Run 3 Run 3 Run 4 Run 5 Run 5 1 1020203356 0 1020203579 1 1020203872 0 1020204972 0 1020204972 0 1020204972 0 1020204972 1 1020204972 1 1020204972 1 1020204972 1 1020204972 1 1020204972 1 1020204072 1 1020204072 1 1020204072 1 1020204072 1 1020204072 1 1 1020204072 1	Condor Tra	nscode 16 Machines	s								
2 12022033538 13 12022035979 13 12022033877 3 12022040972 16 1202204093 14 3 12022033737 23 1202203607 23 1202203890 16 1202204072 16 1202204050 14 4 1202203373 23 12022036071 23 1202203840 16 1202204072 16 1202203070 16 5 1202203377 27 1202203611 27 1202203848 24 1202204068 24 1202204317 26 6 1202203379 29 1202203614 30 1202203852 28 1202204683 27 120220418 27 8 1202203383 33 1202203614 30 1202203852 28 1202204684 28 1202204318 27 10 1202203383 33 1202203620 36 1202203855 31 1202204684 28 1202204312 29 11 1202203383 38 1202	Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A
1	1	1202203350	0	1202203584	0	1202203824	0	1202204056	0	1202204291	0
1	2	1202203363	13	1202203597	13	1202203827	3	1202204059	3	1202204294	3
5 1202203376 26 1202203610 26 1202203847 23 1202204076 20 120220371 26 6 1202203377 27 1202203611 27 1202203848 24 1202204080 24 120220416 25 7 102203379 29 1202203614 30 1202203852 28 1202204084 28 1202204317 26 8 1202203838 33 1202203618 34 1202203852 30 1202204084 28 1202204320 29 10 1202203885 35 1202203620 36 1202203855 31 1202204088 32 1202204323 32 11 1202203386 36 1202203621 37 1202203855 31 1202204088 32 1202204323 32 11 1202203388 38 1202203622 48 1202203858 34 120220409 34 1202204323 32 13 1202203397 47 1202203625	3	1202203373	23	1202203607	23	1202203839	15	1202204072	16	1202204305	14
6 1202203377 27 1202203611 27 1202203848 24 1202204080 24 1202204161 25 7 1202203379 29 1202203614 30 1202203899 25 1202204083 27 1202204131 26 8 1202203382 32 1202203614 30 1202203852 28 1202204084 28 1202204323 29 9 1202203383 33 1202203620 36 1202203854 30 1202204084 28 1202204323 32 10 1202203385 36 1202203620 36 1202203855 31 1202204090 34 1202204323 32 11 1202203386 38 1202203622 38 1202203855 31 1202204090 34 1202204372 36 13 1202203390 42 1202203622 48 1202203853 34 1202204093 37 1202204323 37 14 1202203390 47 120220	4	1202203373	23	1202203607	23	1202203840	16	1202204072	16	1202204305	14
	5	1202203376	26	1202203610	26	1202203847	23	1202204076	20	1202204307	16
8 1202203382 32 1202203614 30 1202203852 28 1202204084 28 1202204312 29 9 1202203383 33 1202203618 34 1202203855 30 1202204084 28 1202204320 29 10 1202203386 35 1202203620 36 1202203855 31 1202204088 32 1202204323 32 11 1202203386 36 1202203621 37 1202203855 31 1202204090 34 1202204322 36 12 1202203392 42 1202203624 40 1202203858 34 1202204091 35 1202204322 36 13 1202203392 42 1202203624 40 1202203863 39 1202204091 35 1202204322 36 15 1202203397 47 1202203623 45 1202203868 4 1202204094 38 1202204333 42 16 1202203402 51 12022	6	1202203377	27	1202203611	27	1202203848	24	1202204080	24	1202204316	25
9 1202203838 33 1202203618 34 1202203855 30 1202204848 28 1202204320 29 10 1202203385 35 1202203620 36 1202203855 31 1202204888 32 1202204323 32 11 1202203886 36 1202203621 37 1202203855 31 1202204090 34 1202204327 36 12 1202203888 38 1202203622 38 1202203855 31 1202204090 34 1202204327 36 13 1202203929 42 1202203622 40 1202203853 35 1202204091 35 1202204332 37 14 1202203397 47 1202203629 45 1202203863 39 1202204094 38 1202204331 40 16 1202203497 47 1202203629 48 1202203868 44 120220410 46 1202204331 40 17 1202203401 51 1202	7	1202203379	29	1202203614	30	1202203849	25	1202204083	27	1202204317	26
100 12022031885 35 1202203620 36 1202203855 31 1202204088 32 1202204323 32	8	1202203382	32	1202203614	30	1202203852	28	1202204084	28	1202204318	27
11 1202203386 36 1202203621 37 1202203855 31 1202204090 34 1202204325 34 12	9	1202203383	33	1202203618	34	1202203854	30	1202204084	28	1202204320	29
12	10	1202203385	35	1202203620	36	1202203855	31	1202204088	32	1202204323	32
13 1202203392 42 1202203624 40 1202203889 35 1202204091 35 1202204328 37 14 120220396 46 1202203625 41 1202203863 39 1202204093 37 1202204330 39 15 1202203397 47 1202203622 48 1202203868 44 1202204004 38 1202204332 41 16 1202203401 51 1202203632 48 1202203872 48 1202204100 44 1202204333 42 18 1202203402 52 1202203637 53 1202203872 48 1202204105 49 1202204333 42 18 1202203402 52 1202203638 54 1202203876 52 1202204105 49 1202204337 46 19 1202203405 55 1202203638 54 1202203877 53 1202204107 51 1202204341 50 21 1202203408 58 120	11	1202203386	36	1202203621	37	1202203855	31	1202204090	34	1202204325	34
14 1202203396 46 1202203625 41 1202203883 39 1202204099 37 1202204330 39 15 1202203397 47 1202203629 45 1202203888 44 1202204094 38 1202204331 40 16 1202203397 47 1202203632 48 1202203888 44 1202204100 44 1202204332 41 17 1202203401 51 1202203635 51 1202203872 48 1202204102 46 1202204333 42 18 1202203402 52 1202203637 53 1202203874 50 1202204105 49 1202204333 42 19 1202203404 54 1202203638 54 1202203876 52 1202204107 51 1202204331 50 21 1202203405 56 1202203633 59 1202203880 56 1202204107 51 1202204341 50 21 1202203408 58 12	12	1202203388	38	1202203622	38	1202203858	34	1202204090	34	1202204327	36
15	13	1202203392	42	1202203624	40	1202203859	35	1202204091	35	1202204328	37
16 1202203397 47 1202203632 48 1202203868 44 1202204100 44 1202203333 42 17 1202203401 51 1202203635 51 1202203872 48 1202204102 46 1202203333 42 18 1202203402 52 1202203637 53 1202203874 50 1202204105 49 1202204337 46 19 1202203404 54 1202203638 54 1202203876 52 1202204107 51 1202204337 48 20 1202203405 55 1202203639 55 1202203877 53 1202204107 51 1202204341 50 21 1202203406 56 1202203643 59 1202203880 56 1202204110 54 1202204346 55 22 1202203408 58 1202203644 60 1202203883 59 1202204112 56 1202204347 56 23 1202203411 61 12	14	1202203396	46	1202203625	41	1202203863	39	1202204093	37	1202204330	39
17	15	1202203397	47	1202203629	45	1202203868	44	1202204094	38	1202204331	40
18 1202203402 52 1202203637 53 1202203874 50 1202204105 49 1202204337 46 19 1202203404 54 1202203638 54 1202203876 52 1202204107 51 1202204339 48 20 1202203405 55 1202203639 55 1202203880 56 1202204107 51 1202204341 50 21 1202203406 56 1202203643 59 1202203880 56 1202204110 54 1202204346 55 22 1202203408 58 1202203644 60 1202203882 58 1202204112 56 1202204347 56 23 1202203411 61 1202203645 61 1202203883 59 1202204112 56 1202204352 61 25 1202203417 67 1202203649 65 1202203887 63 1202204113 57 1202204355 64 26 1202203417 67 12	16	1202203397	47	1202203632	48	1202203868	44	1202204100	44	1202204332	41
19	17	1202203401	51	1202203635	51	1202203872	48	1202204102	46	1202204333	42
20 1202203405 55 1202203639 55 1202203877 53 1202204107 51 1202204341 50 21 1202203406 56 1202203643 59 1202203880 56 1202204110 54 1202204346 55 22 1202203408 58 1202203644 60 1202203882 58 1202204112 56 1202204347 56 23 1202203411 61 1202203644 60 1202203883 59 1202204112 56 1202204349 58 24 1202203411 61 1202203645 61 1202203883 59 1202204113 57 1202204352 61 25 1202203417 67 1202203649 65 1202203887 63 1202204113 57 1202204355 64 26 1202203418 68 1202203653 69 1202203889 65 1202204118 62 1202204357 66 28 1202203420 70 12	18	1202203402	52	1202203637	53	1202203874	50	1202204105	49	1202204337	46
21 1202203406 56 1202203643 59 1202203880 56 1202204110 54 1202204346 55 22 1202203408 58 1202203644 60 1202203882 58 1202204112 56 120220347 56 23 1202203411 61 1202203644 60 1202203883 59 1202204112 56 1202204349 58 24 1202203411 61 1202203645 61 1202203883 59 1202204113 57 1202204352 61 25 1202203417 67 1202203659 65 1202203887 63 1202204117 61 1202204355 64 26 1202203417 67 1202203650 66 1202203889 65 1202204118 62 1202204357 66 27 1202203418 68 1202203654 70 1202203891 67 1202204121 65 1202204357 66 28 1202203424 74 120	19	1202203404	54	1202203638	54	1202203876	52	1202204107	51	1202204339	48
22 1202203408 58 1202203644 60 1202203882 58 1202204112 56 1202204347 56 23 1202203411 61 1202203644 60 1202203883 59 1202204112 56 1202204349 58 24 1202203411 61 1202203645 61 1202203883 59 1202204113 57 1202204352 61 25 1202203417 67 1202203649 65 1202203887 63 1202204117 61 1202204355 64 26 1202203417 67 1202203650 66 1202203889 65 1202204118 62 1202204357 66 27 1202203418 68 1202203653 69 1202203891 67 1202204121 65 1202204357 66 28 1202203420 70 1202203658 70 1202203895 71 1202204126 70 1202204359 68 30 1202203424 74 12	20	1202203405	55	1202203639	55	1202203877	53	1202204107	51	1202204341	50
23 1202203411 61 1202203644 60 1202203883 59 1202204112 56 1202204349 58 24 1202203411 61 1202203645 61 1202203883 59 1202204113 57 1202204552 61 25 1202203417 67 1202203659 66 1202203887 63 1202204117 61 1202204355 64 26 1202203417 67 1202203650 66 1202203891 67 1202204118 62 1202204357 66 27 1202203420 70 1202203654 70 1202203891 67 1202204121 65 1202204358 67 28 1202203424 74 1202203658 74 1202203895 71 1202204122 66 1202204359 68 30 1202203424 74 1202203658 74 1202203896 72 1202204126 70 1202204362 71 31 1202203428 78 12	21	1202203406	56	1202203643	59	1202203880	56	1202204110	54	1202204346	55
24 1202203411 61 1202203645 61 1202203883 59 1202204113 57 1202204352 61 25 1202203417 67 1202203649 65 1202203887 63 1202204117 61 120220355 64 26 1202203417 67 1202203650 66 1202203889 65 1202204118 62 1202204357 66 27 1202203418 68 1202203653 69 1202203891 67 1202204121 65 1202204357 66 28 1202203420 70 1202203654 70 1202203892 68 1202204122 66 1202204358 67 29 1202203424 74 1202203658 74 1202203895 71 1202204126 70 1202204362 71 31 1202203428 78 1202203661 77 1202203899 75 1202204129 73 1202204365 74 32 1202203428 78 120	22	1202203408	58	1202203644	60	1202203882	58	1202204112	56	1202204347	56
25 1202203417 67 1202203649 65 1202203887 63 1202204117 61 1202204355 64 26 1202203417 67 1202203650 66 1202203889 65 1202204118 62 1202204357 66 27 1202203418 68 1202203653 69 1202203891 67 1202204121 65 1202204357 66 28 1202203420 70 1202203654 70 1202203892 68 1202204122 66 1202204358 67 29 1202203424 74 1202203658 74 1202203895 71 1202204126 70 1202204359 68 30 1202203424 74 1202203658 74 1202203896 72 1202204126 70 1202204362 71 31 1202203428 78 1202203661 77 1202203899 75 1202204129 73 1202204365 74 32 1202203428 78 12	23	1202203411	61	1202203644	60	1202203883	59	1202204112	56	1202204349	58
26 1202203417 67 1202203650 66 1202203889 65 1202204118 62 1202204357 66 27 1202203418 68 1202203653 69 1202203891 67 1202204121 65 1202204357 66 28 1202203420 70 1202203654 70 1202203892 68 1202204122 66 120220458 67 29 1202203424 74 1202203658 74 1202203895 71 1202204126 70 1202204359 68 30 1202203424 74 1202203658 74 1202203896 72 1202204126 70 1202204362 71 31 1202203428 78 1202203661 77 1202203899 75 1202204129 73 1202204363 72 32 1202203428 78 1202203663 79 1202203900 76 1202204131 75 1202204365 74 33 1202203429 79 120	24	1202203411	61	1202203645	61	1202203883	59	1202204113	57	1202204352	61
27 1202203418 68 1202203653 69 1202203891 67 1202204121 65 1202204357 66 28 1202203420 70 1202203654 70 1202203892 68 1202204122 66 1202204358 67 29 1202203424 74 1202203658 74 1202203895 71 1202204126 70 1202204359 68 30 1202203424 74 1202203658 74 1202203896 72 1202204126 70 1202204362 71 31 1202203428 78 1202203661 77 1202203899 75 1202204129 73 1202204363 72 32 1202203428 78 1202203663 79 1202203900 76 1202204131 75 1202204365 74 33 1202203429 79 1202203667 83 1202203904 80 1202204134 78 1202204370 79	25	1202203417	67	1202203649	65	1202203887	63	1202204117	61	1202204355	64
28 1202203420 70 1202203654 70 1202203892 68 1202204122 66 1202204358 67 29 1202203424 74 1202203658 74 1202203895 71 1202204126 70 1202204359 68 30 1202203424 74 1202203658 74 1202203896 72 1202204126 70 1202204362 71 31 1202203428 78 1202203661 77 1202203899 75 1202204129 73 1202204363 72 32 1202203428 78 1202203663 79 1202203900 76 1202204131 75 1202204365 74 33 1202203429 79 1202203667 83 1202203904 80 1202204134 78 1202204370 79	26	1202203417	67	1202203650	66	1202203889	65	1202204118	62	1202204357	66
29 1202203424 74 1202203658 74 1202203895 71 1202204126 70 1202204359 68 30 1202203424 74 1202203658 74 1202203896 72 1202204126 70 1202204362 71 31 1202203428 78 1202203661 77 1202203899 75 1202204129 73 1202204363 72 32 1202203428 78 1202203663 79 1202203900 76 1202204131 75 1202204365 74 33 1202203429 79 1202203667 83 1202203904 80 1202204134 78 1202204370 79	27	1202203418	68	1202203653	69	1202203891	67	1202204121	65	1202204357	66
30 1202203424 74 1202203658 74 1202203896 72 1202204126 70 1202204362 71 31 1202203428 78 1202203661 77 1202203899 75 1202204129 73 1202204363 72 32 1202203428 78 1202203663 79 1202203900 76 1202204131 75 1202204365 74 33 1202203429 79 1202203667 83 1202203904 80 1202204134 78 1202204370 79	28	1202203420	70	1202203654	70	1202203892	68	1202204122	66	1202204358	67
31 1202203428 78 1202203661 77 1202203899 75 1202204129 73 1202204363 72 32 1202203428 78 1202203663 79 1202203900 76 1202204131 75 1202204365 74 33 1202203429 79 1202203667 83 1202203904 80 1202204134 78 1202204370 79	29	1202203424	74	1202203658	74	1202203895	71	1202204126	70	1202204359	68
32 1202203428 78 1202203663 79 1202203900 76 1202204131 75 1202204365 74 33 1202203429 79 1202203667 83 1202203904 80 1202204134 78 1202204370 79	30	1202203424	74	1202203658	74	1202203896	72	1202204126	70	1202204362	71
33 1202203429 79 1202203667 83 1202203904 80 1202204134 78 1202204370 79	31	1202203428	78	1202203661	77	1202203899	75	1202204129	73	1202204363	72
	32	1202203428	78	1202203663	79	1202203900	76	1202204131	75	1202204365	74
34 1202203432 82 1202203667 83 1202203906 82 1202204136 80 1202204372 81	33	1202203429	79	1202203667	83	1202203904	80	1202204134	78	1202204370	79
	34	1202203432	82	1202203667	83	1202203906	82	1202204136	80	1202204372	81

25	1202202424		120220266	0.5	120220202	02	1202204127	01	1202204272	02
35	1202203434	84	1202203669	85	1202203907	83	1202204137	81	1202204373	82
36	1202203436 1202203440	90	1202203670	90	1202203908	84	1202204138	82	1202204377	86
38	1202203440	90		91	1202203910	86	1202204142	86	1202204378	88
39	1202203440	93	1202203675	94	1202203912	88	1202204142	86	1202204379	90
40	1202203443	95	1202203679	95	1202203912	92	1202204142	90	1202204381	91
41	1202203448	98	1202203679	100	1202203910	97	1202204140	95	1202204385	94
42	1202203449	99	1202203684	100	1202203921	97	1202204151	96	1202204383	96
43	1202203447	101	1202203689	105	1202203921	102	1202204155	99	1202204387	96
44	1202203451	103	1202203689	105	1202203926	102	1202204155	100	1202204387	101
45	1202203454	104	1202203690	106	1202203920	103	1202204150	101	1202204392	102
46	1202203454	104	1202203691	107	1202203927	105	1202204157	104	1202204396	105
47	1202203459	109	1202203692	108	1202203929	107	1202204160	105	1202204398	107
48	1202203462	112	1202203695	111	1202203932	108	1202204163	107	1202204400	109
49	1202203462	112	1202203697	113	1202203934	110	1202204164	108	1202204400	109
50	1202203467	117	1202203697	113	1202203934	110	1202204164	108	1202204404	113
51	1202203469	119	1202203705	121	1202203938	114	1202204168	112	1202204407	116
52	1202203470	120	1202203705	121	1202203942	118	1202204172	116	1202204407	116
53	1202203470	120	1202203705	121	1202203943	119	1202204173	117	1202204409	118
54	1202203474	124	1202203709	125	1202203946	122	1202204175	120	1202204410	119
55	1202203474	125	1202203709	125	1202203947	123	1202204170	120	1202204410	121
56	1202203475	125	1202203703	127	1202203949	125	1202204178	122	1202204414	123
57	1202203476	126	1202203716	132	1202203953	129	1202204183	127	1202204415	124
58	1202203479	129	1202203716	132	1202203955	131	1202204185	129	1202204421	130
59	1202203483	133	1202203717	133	1202203955	131	1202204185	129	1202204422	131
60	1202203484	134	1202203718	134	1202203957	133	1202204186	130	1202204423	132
61	1202203487	137	1202203720	136	1202203957	133	1202204187	131	1202204427	136
62	1202203491	141	1202203722	138	1202203961	137	1202204190	134	1202204429	138
63	1202203492	142	1202203727	143	1202203963	139	1202204194	138	1202204431	140
64	1202203493	143	1202203728	144	1202203967	143	1202204196	140	1202204432	141
65	1202203496	146	1202203731	147	1202203968	144	1202204198	142	1202204435	144
66	1202203496	146	1202203731	147	1202203970	146	1202204199	143	1202204436	145
67	1202203499	149	1202203736	152	1202203973	149	1202204204	148	1202204436	145
68	1202203501	151	1202203737	153	1202203975	151	1202204205	149	1202204439	148
69	1202203503	153	1202203739	155	1202203977	153	1202204207	151	1202204441	150
70	1202203505	155	1202203739	155	1202203979	155	1202204208	152	1202204444	153
71	1202203506	156	1202203740	156	1202203980	156	1202204210	154	1202204447	156
72	1202203512	162	1202203743	159	1202203980	156	1202204210	154	1202204449	158
73	1202203513	163	1202203745	161	1202203983	159	1202204213	157	1202204450	159
74	1202203513	163	1202203748	164	1202203985	161	1202204215	159	1202204452	161
75	1202203518	168	1202203751	167	1202203985	161	1202204216	160	1202204455	164
76	1202203518	168	1202203754	170	1202203991	167	1202204220	164	1202204457	166
77	1202203519	169	1202203754	170	1202203991	167	1202204221	165	1202204457	166
78	1202203523	173	1202203758	174	1202203996	172	1202204225	169	1202204461	170
79	1202203524	174	1202203760	176	1202203996	172	1202204226	170	1202204462	171
80	1202203525	175	1202203761	177	1202203998	174	1202204229	173	1202204464	173
81	1202203527	177	1202203762	178	1202203999	175	1202204229	173	1202204464	173
82	1202203527	177	1202203766	182	1202204003	179	1202204233	177	1202204465	174
83	1202203532	182	1202203767	183	1202204004	180	1202204233	177	1202204469	178
84	1202203536	186	1202203770	186	1202204006	182	1202204235	179	1202204473	182
85	1202203536	186	1202203771	187	1202204007	183	1202204237	181	1202204474	183
86	1202203539	189	1202203771	187	1202204008	184	1202204238	182	1202204477	186
87	1202203541	191	1202203776	192	1202204011	187	1202204241	185	1202204479	188
88	1202203542	192	1202203777	193	1202204012	188	1202204243	187	1202204480	189
89	1202203544	194	1202203780	196	1202204016	192	1202204247	191	1202204482	191
90	1202203545	195	1202203782	198	1202204017	193	1202204247	191	1202204483	192
91	1202203548	198	1202203785	201	1202204022	198	1202204252	196	1202204486	195
		1		i .					-	
92	1202203549	199	1202203787	203	1202204023	199	1202204253	197	1202204487	196

94	1202203552	202	1202203790	206	1202204026	202	1202204256	200	1202204490	199
95	1202203554	204	1202203793	209	1202204028	204	1202204258	202	1202204492	201
96	1202203559	209	1202203793	209	1202204029	205	1202204259	203	1202204497	206
97	1202203561	211	1202203795	211	1202204031	207	1202204261	205	1202204498	207
98	1202203562	212	1202203798	214	1202204032	208	1202204263	207	1202204500	209
99	1202203564	214	1202203799	215	1202204034	210	1202204264	208	1202204501	210
100	1202203566	216	1202203800	216	1202204035	211	1202204266	210	1202204504	213

BOINC

BOINC Tra	anscode 1 Machine									
Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A
1	1203667516	0	1203670289	0	1203685961	0	1203739108	0	1203749532	0
2	1203667534	18	1203670310	21	1203685981	20	1203739129	21	1203749558	26
3	1203667555	39	1203670331	42	1203686003	42	1203739150	42	1203749580	48
4	1203667576	60	1203670357	68	1203686025	64	1203739170	62	1203749601	69
5	1203667599	83	1203670379	90	1203686046	85	1203739193	85	1203749622	90
6	1203667619	103	1203670404	115	1203686066	105	1203739212	104	1203749643	111
7	1203667642	126	1203670431	142	1203686087	126	1203739235	127	1203749664	132
8	1203667662	146	1203670452	163	1203686108	147	1203739259	151	1203749683	151
9	1203667686	170	1203670474	185	1203686128	167	1203739280	172	1203749704	172
10	1203667708	192	1203670494	205	1203686150	189	1203739302	194	1203749725	193
11	1203667727	211	1203670514	225	1203686170	209	1203739322	214	1203749749	217
12	1203667753	237	1203670540	251	1203686192	231	1203739342	234	1203749768	236
13	1203667775	259	1203670568	279	1203686211	250	1203739363	255	1203749791	259
14	1203667796	280	1203670592	303	1203686234	273	1203739383	275	1203749810	278
15	1203667817	301	1203670612	323	1203686253	292	1203739404	296	1203749831	299
16	1203667837	321	1203670634	345	1203686276	315	1203739429	321	1203749852	320
17	1203667859	343	1203670655	366	1203686296	335	1203739451	343	1203749873	341
18	1203667881	365	1203670676	387	1203686316	355	1203739476	368	1203749893	361
19	1203667901	385	1203670695	406	1203686337	376	1203739498	390	1203749914	382
20	1203667922	406	1203670716	427	1203686358	397	1203739522	414	1203749939	407
21	1203667943	427	1203670737	448	1203686378	417	1203739541	433	1203749962	430
22	1203667964	448	1203670758	469	1203686404	443	1203739564	456	1203749986	454
23	1203667984	468	1203670782	493	1203686426	465	1203739583	475	1203750441	909
24	1203668005	489	1203670805	516	1203686447	486	1203739606	498	1203750441	909
25	1203668026	510	1203670826	537	1203686467	506	1203739625	517	1203750484	952
26	1203668046	530	1203670847	558	1203686489	528	1203739648	540	1203750503	971
27	1203668067	551	1203670868	579	1203686509	548	1203739667	559	1203750523	991
28	1203668088	572	1203670888	599	1203686531	570	1203739692	584	1203750543	1011
29	1203668107	591	1203670916	627	1203686552	591	1203739714	606	1203750569	1037
30	1203668129	613	1203670938	649	1203686573	612	1203739734	626	1203750592	1060
31	1203668156	640	1203670959	670	1203686593	632	1203739755	647	1203750612	1080
32	1203668180	664	1203670980	691	1203686619	658	1203739776	668	1203750638	1106
33	1203668202	686	1203671002	713	1203686640	679	1203739796	688	1203750659	1127
34	1203668223	707	1203671023	734	1203686664	703	1203739817	709	1203750679	1147
35	1203668238	722	1203671043	754	1203687115	1154	1203739838	730	1203750700	1168
36	1203668258	742	1203671063	774	1203687115	1154	1203739859	751	1203750725	1193
37	1203668279	763	1203671083	794	1203687157	1196	1203739880	772	1203750747	1215
38	1203668301	785	1203671104	815	1203687183	1222	1203739900	792	1203750768	1236
39	1203668321	805	1203671128	839	1203687204	1243	1203739926	818	1203750789	1257
40	1203668342	826	1203671154	865	1203687225	1264	1203739948	840	1203750810	1278
41	1203668364	848	1203671176	887	1203687246	1285	1203739968	860	1203750832	1300
42	1203668384	868	1203671196	907	1203687266	1305	1203739986	878	1203750856	1324
43	1203668405	889	1203671217	928	1203687287	1326	1203740007	899	1203750879	1347
44	1203668429	913	1203671238	949	1203687309	1348	1203740032	924	1203750898	1366
45	1203668452	936	1203671258	969	1203687330	1369	1203740059	951	1203750919	1387

16	1202668474	050	1202/71270	000	1202/07254	1202	1203740079	071	1202750040	1400
46	1203668474	958	1203671279	990	1203687354	1393		971	1203750940	1408
47	1203668493	977	1203671301	1012	1203687377	1416	1203740105	997	1203750963	1431
48	1203668518	1002	1203671323	1034	1203687397	1436	1203740132	1024	1203750983	1451
49	1203668539	1023	1203671343	1054	1203687420	1459	1203740154	1046	1203751005	1473
50	1203668561	1045	1203671363	1074	1203687439	1478	1203740174	1066	1203751029	1497
51	1203668582	1066	1203671384	1095	1203687464	1503	1203740195	1087	1203751046	1514
52	1203668602	1086	1203671405	1116	1203687486	1525	1203740216	1108	1203751068	1536
53	1203668624	1108	1203671425	1136	1203687511	1550	1203740238	1130	1203751092	1560
54	1203668644	1128	1203671445	1156	1203687534	1573	1203740258	1150	1203751114	1582
55	1203668666	1150	1203671467	1178	1203687556	1595	1203740279	1171	1203751139	1607
56	1203668685	1169	1203671488	1199	1203687580	1619	1203740300	1192	1203751166	1634
57	1203668708	1192	1203671509	1220	1203687602	1641	1203740320	1212	1203751186	1654
58	1203668727	1211	1203671529	1240	1203687622	1661	1203740341	1233	1203751207	1675
59	1203668748	1232	1203671550	1261	1203687643	1682	1203740366	1258	1203751233	1701
60	1203668768	1252	1203671570	1281	1203687664	1703	1203740388	1280	1203751255	1723
61	1203668789	1273	1203671592	1303	1203687684	1723	1203740411	1303	1203751276	1744
62	1203668809	1293	1203671612	1323	1203687706	1745	1203740431	1323	1203751298	1766
63	1203668831	1315	1203671639	1350	1203687726	1765	1203740456	1348	1203751321	1789
64	1203668852	1336	1203671660	1371	1203687748	1787	1203740482	1374	1203751347	1815
65	1203668872	1356	1203671680	1391	1203687769	1808	1203740503	1395	1203751368	1836
66	1203668892	1376	1203671705	1416	1203687789	1828	1203740526	1418	1203751392	1860
67	1203668913	1397	1203671727	1438	1203687811	1850	1203740545	1437	1203751415	1883
68	1203668934	1418	1203671748	1459	1203687831	1870	1203740571	1463	1203751438	1906
69	1203668957	1441	1203671770	1481	1203687856	1895	1203740594	1486	1203751461	1929
70	1203668978	1462	1203671790	1501	1203687878	1917	1203740613	1505	1203751482	1950
71	1203668997	1481	1203671810	1521	1203687899	1938	1203740634	1526	1203751502	1970
72	1203669019	1503	1203671831	1542	1203687920	1959	1203740654	1546	1203751524	1992
73	1203669040	1524	1203671856	1567	1203687941	1980	1203740675	1567	1203751550	2018
74	1203669064	1548	1203671878	1589	1203687963	2002	1203740700	1592	1203751571	2039
75	1203669083	1567	1203671898	1609	1203687982	2021	1203740727	1619	1203751592	2060
76	1203669103	1587	1203671920	1631	1203688003	2042	1203740749	1641	1203751614	2082
77	1203669125	1609	1203671941	1652	1203688029	2068	1203740769	1661	1203751639	2107
78	1203669146	1630	1203671961	1672	1203688055	2094	1203740791	1683	1203751662	2130
79	1203669166	1650	1203671983	1694	1203688033	2117	1203740771	1703	1203751681	2149
80	1203669192	1676	1203672003	1714	1203688098	2137	1203740811	1705	1203751702	2170
81	1203669214	1698	1203672003	1740	1203688038	2157	1203740854	1746	1203751702	2625
82	1203669234	1718	1203672047	1758	1203688139	2178	1203740876	1768	1203752157	2625
83	1203669254	1738	1203672067	1778	1203688161	2200	1203740895	1787	1203752198	2666
84	1203669280	1764	1203672089	1800	1203688182	2221	1203740918	1810	1203752220	2688
85	1203669301	1785	1203672109	1820	1203688206	2245	1203740938	1830	1203752241	2709
86	1203669322	1806	1203672130	1841	1203688227	2266	1203740960	1852	1203752261	2729
87	1203669342	1826	1203672154	1865	1203688248	2287	1203740981	1873	1203752282	2750
88	1203669364	1848	1203672177	1888	1203688270	2309	1203741003	1895	1203752304	2772
89	1203669384	1868	1203672198	1909	1203688290	2329	1203741022	1914	1203752757	3225
90	1203669406	1890	1203672218	1929	1203688311	2350	1203741047	1939	1203752757	3225
91	1203669425	1909	1203672244	1955	1203688331	2370	1203741069	1961	1203752803	3271
92	1203669446	1930	1203672265	1976	1203688352	2391	1203741089	1981	1203752828	3296
93	1203669467	1951	1203672286	1997	1203688805	2844	1203741111	2003	1203752848	3316
94	1203669488	1972	1203672307	2018	1203688806	2845	1203741131	2023	1203752873	3341
95	1203669509	1993	1203672327	2038	1203688851	2890	1203741151	2043	1203752894	3362
96	1203669536	2020	1203672350	2061	1203688872	2911	1203741171	2063	1203752917	3385
97	1203669561	2045	1203672375	2086	1203688893	2932	1203741192	2084	1203752938	3406
98	1203669581	2065	1203672397	2108	1203688915	2954	1203741214	2106	1203752957	3425
99	1203669607	2091	1203672418	2129	1203688934	2973	1203741234	2126	1203752983	3451
100	1203669673	2157	1203672483	2194	1203689001	3040	1203741299	2191	1203753049	3517

BOINC Transc	BOINC Transcode 2 Machines										
Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A	
1	1203969736	0	1203971662	0	1203974907	0	1203977162	0	1203981135	0	
2	1203969743	7	1203971677	15	1203974922	15	1203977174	12	1203981155	20	

3	1203969757	21	1203971682	20	1203974932	25	1203977182	20	1203981157	22
4	1203969757	28	1203971682	41	1203974932	36	1203977182	34	1203981137	41
5	1203969764	42	1203971703	49	1203974943	46	1203977196	41	1203981176	41
	1203969778									
7	1203969783	49	1203971733 1203971744	71 82	1203974967 1203974974	60	1203977221 1203977224	59 62	1203981197 1203981198	62
		66				67				63
9	1203969810	74	1203971763	101	1203974990	83	1203977242	80	1203981217	82
	1203969824	88	1203971776	114	1203974995	88	1203977244	82	1203981218	83
10	1203969832	96	1203971793	131	1203975011	104	1203977267	105	1203981238	103
11	1203969848	112	1203971813	151	1203975015	108	1203977269	107	1203981239	104
12	1203969858	122	1203971819	157	1203975032	125	1203977289	127	1203981258	123
13	1203969870	134	1203971844	182	1203975036	129	1203977290	128	1203981264	129
14	1203969880	144	1203971845	183	1203975053	146	1203977309	147	1203981279	144
15	1203969890	154	1203971872	210	1203975056	149	1203977311	149	1203981287	152
16	1203969902	166	1203971873	211	1203975073	166	1203977331	169	1203981300	165
17	1203969911	175	1203971894	232	1203975076	169	1203977332	170	1203981309	174
18	1203969924	188	1203971904	242	1203975094	187	1203977351	189	1203981320	185
19	1203969932	196	1203971920	258	1203975099	192	1203977352	190	1203981330	195
20	1203969945	209	1203971948	286	1203975114	207	1203977373	211	1203981341	206
21	1203969952	216	1203971975	313	1203975140	233	1203977376	214	1203981352	217
22	1203969964	228	1203972002	340	1203975166	259	1203977393	231	1203981361	226
23	1203969973	237	1203972029	367	1203975188	281	1203977398	236	1203981372	237
24	1203969991	255	1203972052	390	1203975209	302	1203977414	252	1203981383	248
25	1203969998	262	1203972079	417	1203975233	326	1203977420	258	1203981395	260
26	1203970013	277	1203972109	447	1203975255	348	1203977435	273	1203981403	268
27	1203970020	284	1203972137	475	1203975276	369	1203977441	279	1203981415	280
28	1203970035	299	1203972163	501	1203975298	391	1203977456	294	1203981424	289
29	1203970040	304	1203972192	530	1203975319	412	1203977461	299	1203981437	302
30	1203970060	324	1203972214	552	1203975340	433	1203977477	315	1203981444	309
31	1203970061	325	1203972236	574	1203975362	455	1203977486	324	1203981461	326
32	1203970081	345	1203972266	604	1203975387	480	1203977497	335	1203981469	334
33	1203970083	347	1203972292	630	1203975409	502	1203977512	350	1203981482	347
34	1203970102	366	1203972313	651	1203975430	523	1203977518	356	1203981487	352
35	1203970104	368	1203972337	675	1203975452	545	1203977534	372	1203981507	372
36	1203970122	386	1203972359	697	1203975472	565	1203977538	376	1203981507	372
37	1203970129	393	1203972359	697	1203975493	586	1203977560	398	1203981528	393
38	1203970143	407	1203972360	698	1203975519	612	1203977564	402	1203981529	394
39	1203970151	415	1203972381	719	1203975544	637	1203977582	420	1203981548	413
40	1203970163	427	1203972401	739	1203975553	646	1203977586	424	1203981549	414
41	1203970172	436	1203972402	740	1203975553	646	1203977608	446	1203981569	434
42	1203970185	449	1203972423	761	1203975596	689	1203977628	466	1203981571	436
43	1203970204	468	1203972424	762	1203975614	707	1203977629	467	1203981590	455
44	1203970226	490	1203972446	784	1203975637	730	1203977651	489	1203981590	455
45	1203970246	510	1203972446	784	1203975656	749	1203977653	491	1203981610	475
46	1203970267	531	1203972465	803	1203975677	770	1203977671	509	1203981612	477
47	1203970287	551	1203972466	804	1203975697	790	1203977676	514	1203981632	497
48	1203970307	571	1203972487	825	1203975718	811	1203977693	531	1203981633	498
49	1203970307	592	1203972490	828	1203975740	833	1203977695	533	1203981652	517
50	1203970349	613	1203972511	849	1203975763	856	1203977716	554	1203981659	524
51	1203970373	637	1203972513	851	1203975786	879	1203977718	556	1203981672	537
52	1203970395	659	1203972533	871	1203975807	900	1203977737	575	1203981680	545
53	1203970415	679	1203972537	875	1203975828	921	1203977739	577	1203981693	558
54	1203970437	701	1203972553	891	1203975848	941	1203977759	597	1203981702	567
55	1203970437	720	1203972558	896	1203975873	966	1203977767	605	1203981702	578
56	1203970436	741	1203972573	911	1203975894	987	1203977780	618	1203981713	588
	1203970477	761		917		1008		627		599
57	1203970497		1203972579	931	1203975915 1203975934	1008	1203977789	638	1203981734	
58		783	1203972593				1203977800		1203981747	612
59	1203970539	803	1203972599	937	1203975934	1027	1203977809	647	1203981755	620
60	1203970560	824	1203972619	957	1203975936	1029	1203977820	658	1203981769	634
61	1203970586	850	1203972621	959	1203975957	1050	1203977830	668	1203981775	640

	62	1203970610	874	1203972642	980	1203975978	1071	1203977840	678	1203981794	659
64 1203970627 891 1203972662 1000 1203975981 101 1203977862 70 120397618 68 1203976033 897 1203972682 100 1203976018 1111 1203978777 715 1203981816 68 66 1203970671 91 120397202 1041 1203976010 113 1203979783 721 120398187 702 68 1203970671 93 1203972702 1041 1203976001 113 120397914 79 120398187 722 69 1203970673 95 1203972742 1081 1203976061 1154 1203979714 79 120398182 72 70 1203970713 97 1203972781 119 1203976061 1154 1203977914 79 1203981806 78 71 1203970713 97 1203972825 116 1203976061 1194 1203978081 82 1203981967 78 72 1203970713 97 1203972862											
65 1203970633 897 1203977688 1006 1203970619 111 1203978787 71 120398187 70 66 1203970672 91 1203972682 100 1203976091 112 1203978783 72 120398187 702 67 1203970673 375 1203972723 1061 1203976601 133 1203979191 757 120398187 72 66 1203970604 95 1203972745 1061 1203976601 133 1203979191 757 1203981807 72 70 1203970701 95 1203972780 1141 1203976080 173 1203979002 80 120398190 78 71 1203970713 97 1203972803 141 1203976000 173 1203978003 83 1203981907 78 72 1203970775 98 1203972884 112 1203976102 1214 1203978033 81 1203981907 78 73 1203970775 10											
66 1203970652 916 1203972682 103 1203976691 1112 1203977883 721 1203981837 702 67 1203970673 935 1203927293 1041 1203976040 1133 1203977888 736 1203981837 702 68 1203970673 397 1203972245 1061 120397661 1144 1203977904 77 1203981887 72 70 1203970694 958 1203972766 1104 120397661 1154 1203977904 80 1203981902 78 71 1203970713 977 1203972281 1119 1203976061 1174 1203970734 82 1203972881 119 1203976061 1194 1203970734 82 1203972821 1163 120397613 80 1203981907 83 73 1203970734 98 1203972844 1182 120397613 1206 120397823 101 1203981966 83 74 1203970735 1021 120397											
67 1 203970071 93 1 20397073 104 1 203970604 133 1 20397087 76 1 20397183 72 68 1 203970673 397 1 203972723 1061 1 203970601 113 1 203977907 757 1 203981837 722 60 1 203970601 95 1 203972766 1104 1 203970601 154 1 20397707 70 1 203981907 78 1 203972761 1104 1 203970801 182 1 20397073 77 1 203972781 1119 1 203970801 182 2 20397173 172 1 20397283 1141 1 203970801 143 2 20397183 141 1 203970801 142 2 20397073 89 1 20397284 1182 1 20397181 140 2 20397073 89 1 20397284 1182 1 20397610 140 2 20398839 1 20397284 182 1 20397610 140 2 20398200 82 1 20397184 1 20397184 1 20397884 182 2 20397184 1 20398829 1 20398829 1 20397184											
68 1203970675 937 1203972723 1061 1203970601 1133 1203977947 772 1203981857 722 69 1203970691 958 1203972765 1043 1203976061 1154 1203977942 779 1203981800 765 70 1203970713 978 1203972803 1141 1203976061 1154 1203977804 822 1203981900 765 72 1203970733 978 1203972835 1141 1203976101 1194 1203978035 843 1203981905 785 73 1203970733 978 1203972844 1182 1203976131 1204 1203978035 881 1203981965 813 74 1203970757 1011 203972874 123 120397611 1204 1203978035 881 1203981963 813 75 1203970757 1011 203972877 1235 1203976121 1215 1203978321 110 1203982007 872 75 120											
69 1203970694 955 1203972745 1083 12039706161 1144 1203977941 779 1203981882 747 70 1203970644 958 1203972766 1104 1203970610 1154 1203977924 800 1203981900 75 71 1203970714 978 1203972781 1119 1203976080 1173 1203978058 43 1203981940 810 72 1203970714 978 1203972845 1141 1203976086 1179 1203978050 843 1203981948 810 74 1203970734 998 1203972844 1182 1203976132 1206 1203978053 891 1203981987 82 75 1203970758 1021 1203972887 1225 1203976161 1225 1203978207 1108 1203982033 87 76 1203970758 1040 1203972897 1245 1203976161 1224 1203978207 1108 1203982033 91 77 1											
70 1203970644 958 120397266 1104 120397661 1154 1203977196 80 1203981900 76 71 1203970713 977 120397281 119 1203976080 1173 1203977984 822 1203981945 81 72 1203970733 977 1203972832 1141 1203976103 1179 1203978031 843 1203981945 810 73 1203970733 978 1203972842 1182 1203976113 1206 1203978033 891 1203981987 82 74 1203970785 1021 1203972862 123 120397612 1215 1203978727 108 1203981987 82 75 1203970785 1021 1203972887 1225 1203976163 1229 1203978727 113 1203976163 125 1203978322 113 1203976163 125 1203978333 117 1203982033 98 70 1203970822 1086 120397293 133 120											
71 1203970713 977 120397281 1119 1203976080 1173 1203977984 822 1203981940 78 72 1203970714 978 1203972833 1411 1203976086 1179 1203978085 843 1203981946 810 73 1203970733 978 1203972823 1163 1203976101 1194 1203978085 891 1203981966 811 74 1203970737 1021 1203972887 1223 1203976122 1215 1203978285 108 1203982073 87 76 1203970758 1022 1203972872 1245 1203976162 1223 1203978270 1108 1203982037 87 77 1203970762 1040 120397290 1245 1203976161 1254 1203978320 1108 1203982037 99 79 1203970822 1060 120397293 1288 1203976161 1254 1203978338 117 1203982036 91 81 120											
72 1203970714 978 1203972803 1141 1203976086 1179 1203978003 843 1203981945 810 73 1203970733 997 1203972825 1163 1203976110 1194 1203978031 869 1203981968 831 74 1203970754 998 1203972865 1203 1203976131 1206 12039780703 914 1203981987 852 75 1203970757 1021 1203972865 1223 1203976122 1215 1203978270 1108 1203982007 872 76 1203970758 1040 1203972907 1248 1203976161 1254 1203978720 130 1203982007 99 78 1203970782 1046 1203972907 1288 1203976161 1254 1203978333 1171 1203982007 99 79 1203970782 1066 1203972931 133 1203976161 1254 1203978338 1171 1203982016 90 81 <											
73 1203970733 997 1203972825 1163 1203976101 1194 1203978031 869 120381966 831 74 1203970734 98 1203972844 1182 1203976113 1206 1203978053 891 120398070 852 75 1203970757 1021 1203972865 1203 1203976122 1215 120397820 1108 120398007 872 76 1203970782 1046 120397293 1268 1203976161 1254 120397822 1108 1203982073 999 79 1203970782 1046 120397290 1288 1203976161 1254 1203978333 1171 1203982094 999 79 1203970832 1066 1203972909 1238 1203976184 1277 1203978333 1171 1203982016 1001 81 1203970834 1107 1203972017 1350 1203976184 1277 1203978333 1171 1203982115 1001 82 <									-		
74 1203970734 998 1203972844 1182 1203976113 1206 1203978055 891 120391987 852 75 1203970757 1021 1203972865 1203 1203976122 1215 1203978270 1108 1203982070 872 76 1203970776 1040 1203972877 1225 1203976142 1235 1203978290 1108 1203982032 897 77 1203970776 1040 1203972930 1268 1203976161 1254 1203978312 1150 1203982074 939 79 1203970789 1062 1203972975 1313 1203976163 1256 1203978333 1171 1203982015 980 81 1203970820 1086 1203972975 1313 1203976184 1277 1203978333 1717 1203982115 980 81 1203970831 1107 1203973737 1375 120397625 1298 1203978340 1217 1203982122 1067 82											
75 1203970757 1021 1203972865 1203 1203976122 1215 1203978270 1108 120392003 872 76 1203970758 1022 1203972887 1225 1203976136 1229 1203978270 1108 1203982032 897 77 1203970782 1046 1203972970 1245 1203976161 1254 1203978312 1150 1203982074 939 78 1203970782 1046 1203972950 1288 1203976161 1254 1203978333 1171 1203982079 961 80 1203970820 1062 1203972951 1313 1203976184 1277 1203978333 1171 120398215 980 81 1203970830 1044 1203972971 1329 1203976188 1281 1203978303 1217 1203982166 0101 82 1203970843 1118 1203973078 1375 1203976218 1298 1203978400 1238 1203982126 1073 83											
76 1203970758 1022 1203972887 1225 1203976136 1229 1203978270 1108 1203982032 897 77 1203970776 1040 1203972971 1245 1203976142 1235 1203978292 1130 1203982053 918 78 1203970782 1046 1203972990 1288 1203976163 1256 1203978333 1171 1203982074 939 79 1203970822 1086 1203972975 1313 1203976184 1277 1203978338 1196 1203982115 980 80 1203970830 1094 1203973012 1350 1203976188 1281 1203978379 1217 1203982136 1001 82 1203970843 1107 1203973012 1350 1203976212 1340 1203978400 1238 1203982126 1025 83 1203970843 1116 1203973038 1396 1203976221 130 1203978401 1258 1203982120 1047 85 <td></td>											
77 1203970776 1040 1203972907 1245 1203976142 1235 1203978292 1130 1203982053 918 78 1203970782 1046 1203972930 1268 1203976161 1254 1203978312 1150 1203982074 939 79 120397082 1086 1203972950 1288 1203976184 1277 1203978333 1171 1203982015 980 80 1203970830 1094 120397291 1329 1203976188 1281 1203978338 1196 1203982115 980 81 1203970843 1107 1203973012 1350 1203976255 1298 1203978400 1238 1203982160 1025 83 1203970843 1118 1203973037 1375 1203976211 1304 1203978400 1238 1203982122 1067 84 1203970843 1118 1203973038 1366 1203976224 1304 1203978401 1239 1203982222 1067 84 <td></td>											
78 1203970782 1046 120397290 1268 1203976161 1254 1203978312 1150 1203982074 939 79 1203970798 1062 1203972950 1288 1203976163 1256 1203978333 1171 120398206 961 80 1203970822 1086 1203972975 1313 1203976184 1277 1203978388 1196 1203982115 980 81 1203970830 1094 1203972012 1350 1203976205 1298 1203978400 1238 1203982160 1025 83 1203970854 1118 1203973037 1375 1203976211 1304 1203978400 1238 1203982182 1047 84 1203970864 1128 1203973088 1396 1203976227 1320 1203978401 1279 1203982223 1088 86 1203970889 1162 1203973084 1422 1203976256 1349 1203978501 1339 1203982244 1109 87 <td></td> <td></td> <td></td> <td></td> <td>-</td> <td></td> <td>-</td> <td></td> <td></td> <td></td> <td></td>					-		-				
79 1203970798 1062 1203972950 1288 1203976163 1256 1203978333 1171 1203982096 961 80 1203970822 1086 1203972975 1313 1203976184 1277 1203978388 1196 1203982115 980 81 1203970830 1094 1203972991 1329 1203976188 1281 1203978379 1217 1203982136 1001 82 1203970843 1107 1203973012 1350 1203976205 1298 1203978400 1238 1203982160 1025 83 1203970864 1118 1203973088 1396 1203976227 1320 120397840 1228 1203982122 1047 85 1203970864 1140 1203973088 1396 12039796227 1320 1203978407 1305 1203982223 1088 86 1203970889 1162 1203973034 1422 1203976264 1349 1203978501 1339 1203982224 1199 87											
80 1203970822 1086 1203972975 1313 1203976184 1277 1203978358 1196 1203982115 980 81 1203970830 1094 120397291 1329 1203976205 1298 1203978400 1238 1203982136 1001 82 1203970843 1107 120397307 1375 1203976205 1298 1203978400 1238 1203982160 1025 83 1203970844 1118 1203973073 1375 1203976271 1320 1203978441 1279 1203982122 1067 84 1203970876 1140 1203973083 1396 1203976271 1320 1203978467 1305 1203982223 1088 86 1203970889 1153 1203973083 1401 1203976247 1340 1203978501 1339 1203982244 1109 87 1203970898 1162 1203973184 1422 1203976268 1361 1203978501 1339 1203982244 1109 89<											
81 1203970830 1094 120397291 1329 1203976188 1281 1203978379 1217 1203982136 1001 82 1203970843 1107 1203973012 1350 1203976205 1298 1203978400 1238 1203982136 1001 83 1203970844 1118 1203973087 1375 1203976227 1320 1203978420 1258 1203982120 1067 84 1203970864 1128 1203973088 1396 1203976227 1320 1203978441 1279 1203982223 1067 85 1203970876 1140 1203973083 1396 1203976247 1340 1203978467 1305 1203982224 1109 86 1203970898 1162 1203973084 1422 1203976268 1349 1203978501 1339 1203982284 1129 88 1203970910 1174 1203973105 1443 120397628 1371 1203978590 1407 1203982284 1159 90	1 1 1								·		
82 1203970843 1107 1203973012 1350 1203976205 1298 1203978400 1238 1203982160 1025 83 1203970854 1118 1203973037 1375 1203976211 1304 1203978420 1258 1203982182 1047 84 1203970864 1128 1203973058 1396 1203976227 1320 1203978441 1279 1203982202 1067 85 1203970876 1140 1203973058 1396 1203976231 1324 1203978467 1305 1203982223 1088 86 1203970889 1162 1203973031 1401 1203976256 1349 1203978501 1339 1203982244 1109 87 1203970898 1162 1203973104 1442 1203976268 1361 1203978501 1339 1203982285 1150 88 1203970910 1174 1203973105 1443 1203976278 1371 1203978501 1339 1203982285 1150											
1203970854 1118											
84 1203970864 1128 120397058 1396 1203976227 1320 1203978441 1279 1203982202 1067 85 1203970876 1140 1203973058 1396 1203976231 1324 1203978467 1305 1203982223 1088 86 1203970898 1153 1203973063 1401 1203976256 1349 1203978501 1339 1203982244 1109 87 1203970910 1174 1203973104 1442 1203976268 1361 1203978547 1385 1203982285 1150 89 1203970925 1189 1203973105 1443 1203976289 1382 1203978569 1407 1203982289 1154 90 1203970911 1195 1203973125 1463 1203976289 1382 1203978569 1407 1203982289 1154 91 1203970946 1210 1203973127 1465 1203976289 1391 1203978616 1454 1203982325 1190 9											
85 1203970876 1140 1203973058 1396 1203976231 1324 1203978467 1305 1203982223 1088 86 1203970889 1153 1203973063 1401 1203976247 1340 1203978501 1339 1203982244 1109 87 1203970898 1162 1203973084 1422 1203976256 1349 1203978501 1339 1203982244 1109 88 120397010 1174 1203973104 1442 1203976268 1361 1203978547 1385 1203982285 1150 89 1203970925 1189 1203973105 1443 1203976289 1382 1203978569 1407 1203982289 1154 90 1203970931 1195 1203973127 1465 1203976298 1391 1203978616 1454 1203982289 1154 91 1203970952 1216 1203973145 1483 1203976319 1412 1203978681 1454 1203982325 1190 9											
86 1203970889 1153 1203973063 1401 1203976247 1340 1203978501 1339 1203982244 1109 87 1203970898 1162 1203973084 1422 1203976256 1349 1203978501 1339 1203982264 1129 88 1203970910 1174 1203973104 1442 1203976268 1361 1203978547 1385 1203982285 1150 89 1203970925 1189 1203973105 1443 1203976278 1371 1203978569 1407 1203982289 1154 90 1203970931 1195 1203973125 1463 1203976289 1382 1203978569 1428 1203982289 1154 91 1203970946 1210 1203973127 1465 1203976298 1391 1203978616 1454 1203982325 1170 92 1203970972 1231 1203973145 1483 1203976310 1403 1203978668 1506 1203982328 1193											
87 1203970898 1162 1203973084 1422 1203976256 1349 1203978501 1339 1203982264 1129 88 1203970910 1174 1203973104 1442 1203976268 1361 1203978547 1385 1203982285 1150 89 1203970925 1189 1203973105 1443 1203976278 1371 1203978569 1407 1203982289 1154 90 1203970931 1195 1203973125 1463 1203976289 1382 1203978590 1428 1203982289 1154 91 1203970946 1210 1203973127 1465 1203976398 1391 1203978616 1454 1203982325 1190 93 1203970967 1231 1203973166 1504 1203976319 1412 1203978688 1506 1203982328 1193 94 1203970972 1236 1203973169 1507 1203976331 1424 1203978691 1529 1203982347 1212											
88 1203970910 1174 1203973104 1442 1203976268 1361 1203978547 1385 1203982285 1150 89 1203970925 1189 1203973105 1443 1203976278 1371 1203978569 1407 1203982289 1154 90 1203970931 1195 1203973125 1463 1203976289 1382 1203978590 1428 1203982289 1154 91 1203970946 1210 1203973127 1465 1203976298 1391 1203978616 1454 1203982305 1170 92 1203970952 1216 1203973145 1483 1203976310 1403 1203978642 1480 1203982325 1190 93 1203970967 1231 1203973166 1504 1203976319 1412 1203978668 1506 1203982328 1193 94 1203970972 1236 1203973187 1525 1203976331 1424 1203978691 1529 1203982347 1212											
89 1203970925 1189 1203973105 1443 1203976278 1371 1203978569 1407 1203982289 1154 90 1203970931 1195 1203973125 1463 1203976289 1382 1203978590 1428 1203982289 1154 91 1203970946 1210 1203973127 1465 1203976298 1391 1203978616 1454 1203982305 1170 92 1203970952 1216 1203973145 1483 1203976310 1403 1203978642 1480 1203982325 1190 93 1203970967 1231 1203973166 1504 1203976319 1412 120397868 1506 1203982328 1193 94 1203970972 1236 1203973169 1507 1203976331 1424 1203978691 1529 1203982346 1211 95 1203970997 1261 1203973164 1532 1203976339 1432 1203978732 1570 1203982374 1212 9											
90 1203970931 1195 1203973125 1463 1203976289 1382 1203978590 1428 1203982289 1154 91 1203970946 1210 1203973127 1465 1203976298 1391 1203978616 1454 1203982305 1170 92 1203970952 1216 1203973145 1483 1203976310 1403 1203978642 1480 1203982325 1190 93 1203970967 1231 1203973166 1504 1203976319 1412 1203978668 1506 1203982328 1193 94 1203970972 1236 1203973169 1507 1203976331 1424 1203978691 1529 1203982346 1211 95 1203970989 1253 1203973187 1525 1203976331 1424 1203978712 1550 1203982374 1212 96 1203970997 1261 1203973194 1532 1203976352 1445 1203978753 1591 1203982374 1239											
91 1203970946 1210 1203973127 1465 1203976298 1391 1203978616 1454 1203982305 1170 92 1203970952 1216 1203973145 1483 1203976310 1403 1203978642 1480 1203982325 1190 93 1203970967 1231 1203973166 1504 1203976319 1412 1203978668 1506 1203982328 1193 94 1203970972 1236 1203973169 1507 1203976331 1424 1203978691 1529 1203982346 1211 95 1203970989 1253 1203973187 1525 1203976331 1424 1203978712 1550 1203982347 1212 96 1203970997 1261 1203973194 1532 1203976339 1432 1203978732 1570 1203982374 1239 97 1203971009 1273 1203973208 1546 1203976352 1445 1203978753 1591 1203982397 1262											
92 1203970952 1216 1203973145 1483 1203976310 1403 1203978642 1480 1203982325 1190 93 1203970967 1231 1203973166 1504 1203976319 1412 1203978668 1506 1203982328 1193 94 1203970972 1236 1203973169 1507 1203976331 1424 1203978691 1529 1203982346 1211 95 1203970989 1253 1203973187 1525 1203976331 1424 1203978712 1550 1203982347 1212 96 1203970997 1261 1203973194 1532 1203976339 1432 1203978732 1570 1203982374 1239 97 1203971009 1273 1203973208 1546 1203976352 1445 1203978818 1656 1203982397 1262 98 1203971062 1326 1203973260 1598 1203976360 1453 1203978818 1656 1203982377 1602											
93 1203970967 1231 1203973166 1504 1203976319 1412 1203978668 1506 1203982328 1193 94 1203970972 1236 1203973169 1507 1203976331 1424 1203978691 1529 1203982346 1211 95 1203970989 1253 1203973187 1525 1203976331 1424 1203978712 1550 1203982347 1212 96 1203970997 1261 1203973194 1532 1203976339 1432 1203978732 1570 1203982374 1239 97 1203971009 1273 1203973208 1546 1203976362 1445 1203978753 1591 1203982397 1262 98 1203971062 1326 1203973260 1598 1203976360 1453 1203978818 1656 1203982463 1328 99 1203971074 1338 1203973272 1610 1203976417 1510 1203978851 1689 1203982737 1602	-		-						-		
94 1203970972 1236 1203973169 1507 1203976331 1424 1203978691 1529 1203982346 1211 95 1203970989 1253 1203973187 1525 1203976331 1424 1203978712 1550 1203982347 1212 96 1203970997 1261 1203973194 1532 1203976339 1432 1203978732 1570 1203982374 1239 97 1203971009 1273 1203973208 1546 1203976352 1445 1203978753 1591 1203982397 1262 98 1203971062 1326 1203973260 1598 1203976360 1453 1203978818 1656 1203982463 1328 99 1203971074 1338 1203973272 1610 1203976417 1510 1203978851 1689 1203982737 1602	93										
95 1203970989 1253 1203973187 1525 1203976331 1424 1203978712 1550 1203982347 1212 96 1203970997 1261 1203973194 1532 1203976339 1432 1203978732 1570 1203982374 1239 97 1203971009 1273 1203973208 1546 1203976352 1445 1203978753 1591 1203982397 1262 98 1203971062 1326 1203973260 1598 1203976360 1453 1203978818 1656 1203982463 1328 99 1203971074 1338 1203973272 1610 1203976417 1510 1203978851 1689 1203982737 1602											
96 1203970997 1261 1203973194 1532 1203976339 1432 1203978732 1570 1203982374 1239 97 1203971009 1273 1203973208 1546 1203976352 1445 1203978753 1591 1203982397 1262 98 1203971062 1326 1203973260 1598 1203976360 1453 1203978818 1656 1203982463 1328 99 1203971074 1338 1203973272 1610 1203976417 1510 1203978851 1689 1203982737 1602											
97 1203971009 1273 1203973208 1546 1203976352 1445 1203978753 1591 1203982397 1262 98 1203971062 1326 1203973260 1598 1203976360 1453 1203978818 1656 1203982463 1328 99 1203971074 1338 1203973272 1610 1203976417 1510 1203978851 1689 1203982737 1602	96	1203970997	1261						1570		1239
99 1203971074 1338 1203973272 1610 1203976417 1510 1203978851 1689 1203982737 1602											
99 1203971074 1338 1203973272 1610 1203976417 1510 1203978851 1689 1203982737 1602	98	1203971062	1326	1203973260	1598	1203976360	1453	1203978818	1656	1203982463	1328
100 1203971210 1474 1203973614 1952 1203976425 1518 1203978851 1689 1203982737 1602											
	100	1203971210	1474	1203973614	1952	1203976425	1518	1203978851	1689	1203982737	1602

BOINC Tran	scode 4 Machines									
Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A
1	1203486150	0	1203487752	0	1203488870	0	1203492308	0	1203493404	0
2	1203486150	0	1203487754	2	1203488870	0	1203492309	1	1203493405	1
3	1203486157	7	1203487755	3	1203488872	2	1203492317	9	1203493408	4
4	1203486170	20	1203487772	20	1203488891	21	1203492329	21	1203493425	21
5	1203486172	22	1203487777	25	1203488894	24	1203492332	24	1203493426	22
6	1203486178	28	1203487780	28	1203488894	24	1203492339	31	1203493429	25
7	1203486192	42	1203487792	40	1203488912	42	1203492349	41	1203493446	42
8	1203486193	43	1203487799	47	1203488914	44	1203492352	44	1203493447	43
9	1203486199	49	1203487803	51	1203488917	47	1203492359	51	1203493450	46
10	1203486211	61	1203487813	61	1203488932	62	1203492371	63	1203493467	63
11	1203486214	64	1203487823	71	1203488940	70	1203492373	65	1203493470	66
12	1203486219	69	1203487824	72	1203488943	73	1203492384	76	1203493471	67
13	1203486232	82	1203487834	82	1203488954	84	1203492391	83	1203493489	85
14	1203486239	89	1203487846	94	1203488962	92	1203492393	85	1203493489	85
15	1203486243	93	1203487847	95	1203488965	95	1203492407	99	1203493497	93
16	1203486255	105	1203487858	106	1203488974	104	1203492411	103	1203493509	105
17	1203486262	112	1203487866	114	1203488986	116	1203492413	105	1203493510	106
18	1203486265	115	1203487868	116	1203488989	119	1203492433	125	1203493519	115

10	1202496276	126	1202407000	129	1202400005	125	1202402422	125	1203493529	125
19	1203486276	126	1203487880	128	1203488995	125	1203492433	125		125
20	1203486282	132	1203487890	138	1203489010	140	1203492438	130	1203493530	126
21	1203486285	135	1203487893	141	1203489014	144	1203492452	144	1203493541	137
22	1203486297	147	1203487900	148	1203489020	150	1203492454	146	1203493550	146
23	1203486302	152	1203487912	160	1203489032	162	1203492460	152	1203493551	147
24	1203486307	157	1203487915	163	1203489036	166	1203492473	165	1203493563	159
25	1203486317	167	1203487921	169	1203489042	172	1203492478	170	1203493571	167
26	1203486323	173	1203487937	185	1203489052	182	1203492480	172	1203493572	168
27	1203486327	177	1203487937	185	1203489057	187	1203492494	186	1203493585	181
28	1203486341	191	1203487941	189	1203489066	196	1203492501	193	1203493592	188
29	1203486343	193	1203487958	206	1203489076	206	1203492505	197	1203493596	192
30	1203486349	199	1203487958	206	1203489078	208	1203492518	210	1203493607	203
31	1203486364	214	1203487963	211	1203489089	219	1203492526	218	1203493617	213
32	1203486364	214	1203487978	226	1203489098	228	1203492527	219	1203493617	213
33	1203486369	219	1203487980	228	1203489102	232	1203492539	231	1203493632	228
34	1203486385	235	1203487983	231	1203489109	239	1203492546	238	1203493639	235
35	1203486388	238	1203487999	247	1203489119	249	1203492547	239	1203493640	236
36	1203486390	240	1203488002	250	1203489125	255	1203492561	253	1203493659	255
37	1203486407	257	1203488004	252	1203489129	259	1203492567	259	1203493659	255
38		260		268	1203489129	259		262	-	256
39	1203486410		1203488020	268		259	1203492570	262	1203493660	256
	1203486411	261	1203488026		1203489143		1203492580		1203493680	
40	1203486431	281	1203488027	275	1203489146	276	1203492585	277	1203493680	276
41	1203486432	282	1203488027	275	1203489149	279	1203492590	282	1203493680	276
42	1203486434	284	1203488042	290	1203489165	295	1203492593	285	1203493686	282
43	1203486437	287	1203488046	294	1203489167	297	1203492602	294	1203493701	297
44	1203486452	302	1203488049	297	1203489170	300	1203492612	304	1203493701	297
45	1203486455	305	1203488064	312	1203489188	318	1203492618	310	1203493702	298
46	1203486460	310	1203488068	316	1203489190	320	1203492622	314	1203493722	318
47	1203486473	323	1203488072	320	1203489194	324	1203492633	325	1203493724	320
48	1203486479	329	1203488085	333	1203489208	338	1203492639	331	1203493725	321
49	1203486483	333	1203488088	336	1203489216	346	1203492646	338	1203493742	338
50	1203486494	344	1203488094	342	1203489216	346	1203492654	346	1203493745	341
51	1203486501	351	1203488108	356	1203489230	360	1203492662	354	1203493752	348
52	1203486502	352	1203488110	358	1203489236	366	1203492668	360	1203493766	362
53	1203486515	365	1203488116	364	1203489238	368	1203492676	368	1203493768	364
54	1203486523	373	1203488132	380	1203489252	382	1203492686	378	1203493773	369
55	1203486524	374	1203488133	381	1203489257	387	1203492688	380	1203493786	382
56	1203486543	393	1203488137	385	1203489260	390	1203492697	389	1203493788	384
57	1203486548	398	1203488153	401	1203489272	402	1203492708	400	1203493793	389
58	1203486565	415	1203488155	403	1203489277	407	1203492709	401	1203493809	405
59	1203486570	420	1203488157	405	1203489283	413	1203492719	411	1203493812	408
60	1203486586	436	1203488174	422	1203489294	424	1203492729	421	1203493814	410
61	1203486590	440	1203488176	424	1203489299	429	1203492729	421	1203493829	425
62	1203486607	457	1203488177	425	1203489305	435	1203492741	433	1203493830	426
63	1203486611	461	1203488197	445	1203489319	449	1203492750	442	1203493834	430
64	1203486627	477	1203488197	445	1203489320	450	1203492751	443	1203493850	446
65	1203486633	483	1203488199	447	1203489330	460	1203492765	457	1203493854	450
66	1203486648	498	1203488218	466	1203489332	462	1203492770	462	1203493856	452
67	1203486653	503	1203488219	467	1203489340	470	1203492771	463	1203493870	466
68	1203486655	505	1203488225	473	1203489342	472	1203492783	475	1203493877	473
69	1203486668	518	1203488240	488	1203489351	481	1203492791	483	1203493880	476
70	1203486679	529	1203488240	488	1203489362	492	1203492794	486	1203493881	477
71	1203486693	543	1203488242	490	1203489365	495	1203492805	497	1203493891	487
72	1203486699	549	1203488248	496	1203489372	502	1203492805	497	1203493903	499
73	1203486716	566	1203488262	510	1203489382	512	1203492817	509	1203493903	499
74	1203486716	575	1203488264	512	1203489382	517	1203492817	509	1203493903	508
75	1203486737	587	1203488268	516	1203489395	525	1203492826	518	1203493924	520
76	1203486748	598	1203488282	530	1203489403	533	1203492837	529	1203493927	523
77	1203486759	609	1203488284	532	1203489408	538	1203492838	530	1203493933	529

78	1203486773	623	1203488289	537	1203489414	544	1203492846	538	1203493946	542
79	1203486780	630	1203488303	551	1203489423	553	1203492858	550	1203493949	545
80	1203486795	645	1203488305	553	1203489430	560	1203492859	551	1203493952	548
81	1203486801	651	1203488314	562	1203489435	565	1203492871	563	1203493967	563
82	1203486815	665	1203488326	574	1203489448	578	1203492878	570	1203493969	565
83	1203486836	686	1203488326	574	1203489450	580	1203492880	572	1203493974	570
84	1203486857	707	1203488339	587	1203489456	586	1203492894	586	1203493991	587
85	1203486882	732	1203488341	589	1203489471	601	1203492901	593	1203493996	592
86	1203486907	757	1203488346	594	1203489471	601	1203492903	595	1203494010	606
87	1203486929	779	1203488362	610	1203489481	611	1203492914	606	1203494015	611
88	1203486951	801	1203488362	610	1203489491	621	1203492923	615	1203494032	628
89	1203486968	818	1203488367	615	1203489492	622	1203492924	616	1203494036	632
90	1203486968	818	1203488381	629	1203489504	634	1203492935	627	1203494051	647
91	1203486971	821	1203488384	632	1203489512	642	1203492944	636	1203494056	652
92	1203486978	828	1203488387	635	1203489517	647	1203492946	638	1203494077	673
93	1203486996	846	1203488402	650	1203489526	656	1203492956	648	1203494077	673
94	1203487010	860	1203488409	657	1203489532	662	1203492965	657	1203494097	693
95	1203487022	872	1203488410	658	1203489538	668	1203492965	657	1203494098	694
96	1203487034	884	1203488423	671	1203489546	676	1203492981	673	1203494118	714
97	1203487044	894	1203488428	676	1203489554	684	1203492986	678	1203494123	719
98	1203487051	901	1203488432	680	1203489560	690	1203492987	679	1203494161	757
99	1203487109	959	1203488488	736	1203489592	722	1203493046	738	1203494183	779
100	1203487117	967	1203488493	741	1203489610	740	1203493051	743	1203494189	785

BOINC Tra	nscode 8 Machines									
Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A
1	1203924185	0	1203924963	0	1203925945	0	1203926736	0	1203927672	0
2	1203924186	1	1203924966	3	1203925949	4	1203926737	1	1203927676	4
3	1203924188	3	1203924967	4	1203925949	4	1203926737	1	1203927679	7
4	1203924188	3	1203924968	5	1203925952	7	1203926738	2	1203927679	7
5	1203924188	3	1203924968	5	1203925953	8	1203926739	3	1203927679	7
6	1203924188	3	1203924970	7	1203925954	9	1203926740	4	1203927679	7
7	1203924189	4	1203924988	25	1203925957	12	1203926741	5	1203927680	8
8	1203924205	20	1203924989	26	1203925965	20	1203926742	6	1203927680	8
9	1203924209	24	1203924989	26	1203925970	25	1203926758	22	1203927697	25
10	1203924209	24	1203924990	27	1203925973	28	1203926758	22	1203927698	26
11	1203924209	24	1203924993	30	1203925975	30	1203926759	23	1203927701	29
12	1203924209	24	1203925009	46	1203925977	32	1203926759	23	1203927701	29
13	1203924210	25	1203925012	49	1203925977	32	1203926760	24	1203927701	29
14	1203924211	26	1203925013	50	1203925981	36	1203926766	30	1203927702	30
15	1203924212	27	1203925014	51	1203925989	44	1203926767	31	1203927704	32
16	1203924226	41	1203925014	51	1203925991	46	1203926778	42	1203927704	32
17	1203924229	44	1203925029	66	1203925995	50	1203926779	43	1203927718	46
18	1203924229	44	1203925032	69	1203925995	50	1203926779	43	1203927719	47
19	1203924229	44	1203925035	72	1203925999	54	1203926779	43	1203927721	49
20	1203924230	45	1203925035	72	1203926000	55	1203926782	46	1203927722	50
21	1203924231	46	1203925040	77	1203926002	57	1203926788	52	1203927723	51
22	1203924234	49	1203925055	92	1203926011	66	1203926789	53	1203927726	54
23	1203924242	57	1203925055	92	1203926015	70	1203926800	64	1203927727	55
24	1203924248	63	1203925057	94	1203926016	71	1203926800	64	1203927731	59
25	1203924250	65	1203925062	99	1203926017	72	1203926801	65	1203927740	68
26	1203924251	66	1203925075	112	1203926020	75	1203926802	66	1203927742	70
27	1203924254	69	1203925077	114	1203926021	76	1203926803	67	1203927743	71
28	1203924254	69	1203925081	118	1203926027	82	1203926809	73	1203927744	72
29	1203924255	70	1203925084	121	1203926031	86	1203926813	77	1203927744	72
30	1203924256	71	1203925097	134	1203926036	91	1203926820	84	1203927746	74
31	1203924267	82	1203925097	134	1203926036	91	1203926821	85	1203927747	75
32	1203924269	84	1203925101	138	1203926041	96	1203926822	86	1203927753	81
33	1203924272	87	1203925105	142	1203926042	97	1203926824	88	1203927759	87

24	1202024274	90	1202025115	1.54	1202020040	104	1202024020	l 02	1202027762	91
34	1203924274	92	1203925117 1203925118	154	1203926049 1203926053	104	1203926829	93	1203927763	92
35	1203924277 1203924277	92	1203925118	155	1203926057	108	1203926834	99	1203927764 1203927765	93
37	1203924277	92	1203925122	163	1203926057	112	1203926841	105	1203927766	94
38	1203924277	96	1203925120	174	1203926063	118	1203926841	106	1203927700	100
39	1203924281	103	1203925138	175	1203926063	118	1203926843	107	1203927772	100
40	1203924288	107	1203925147	184	1203926072	127	1203926846	110	1203927772	101
41	1203924292	107	1203925147	184	1203926072	128	1203926851	115	1203927773	109
42	1203924296	111	1203925160	197	1203926077	132	1203926856	120	1203927781	113
43	1203924296	111	1203925164	201	1203926083	138	1203926862	126	1203927785	113
44	1203924298	113	1203925167	204	1203926083	138	1203926862	126	1203927786	114
45	1203924290	117	1203925169	206	1203926083	138	1203926864	128	1203927788	116
46	1203924313	128	1203925179	216	1203926093	148	1203926866	130	1203927794	122
47	1203924318	133	1203925188	225	1203926096	151	1203926867	131	1203927795	123
48	1203924319	134	1203925189	226	1203926098	153	1203926871	135	1203927797	125
49	1203924321	136	1203925194	231	1203926104	159	1203926876	140	1203927801	129
50	1203924322	137	1203925200	237	1203926104	159	1203926884	148	1203927806	134
51	1203924334	149	1203925208	245	1203926108	163	1203926886	150	1203927809	137
52	1203924339	154	1203925211	248	1203926113	168	1203926887	151	1203927811	139
53	1203924342	157	1203925215	252	1203926118	173	1203926888	152	1203927811	139
54	1203924343	158	1203925220	257	1203926119	174	1203926889	153	1203927819	147
55	1203924344	159	1203925230	267	1203926124	179	1203926892	156	1203927819	147
56	1203924354	169	1203925235	272	1203926125	180	1203926905	169	1203927820	148
57	1203924364	179	1203925236	273	1203926130	185	1203926908	172	1203927822	150
58	1203924364	179	1203925245	282	1203926134	189	1203926909	173	1203927827	155
59	1203924365	180	1203925251	288	1203926138	193	1203926909	173	1203927831	159
60	1203924367	182	1203925257	294	1203926139	194	1203926911	175	1203927832	160
61	1203924379	194	1203925258	295	1203926146	201	1203926912	176	1203927833	161
62	1203924387	202	1203925267	304	1203926147	202	1203926927	191	1203927840	168
63	1203924387	202	1203925275	312	1203926156	211	1203926928	192	1203927841	169
64	1203924389	204	1203925276	313	1203926159	214	1203926930	194	1203927842	170
65	1203924390	205	1203925288	325	1203926164	219	1203926931	195	1203927842	170
66	1203924406	221	1203925294	331	1203926165	220	1203926932	196	1203927848	176
67	1203924409	224	1203925297	334	1203926167	222	1203926934	198	1203927852	180
68	1203924413	228	1203925301	338	1203926176	231	1203926947	211	1203927852	180
69	1203924428	243	1203925313	350	1203926180	235	1203926949	213	1203927854	182
70	1203924430	245	1203925315	352	1203926187	242	1203926951	215	1203927863	191
71	1203924438	253	1203925318	355	1203926187	242	1203926952	216	1203927864	192
72	1203924448	263	1203925322	359	1203926191	246	1203926953	217	1203927866	194
73	1203924456	271	1203925324	361	1203926196	251	1203926953	217	1203927866	194
74	1203924459	274	1203925334	371	1203926201	256	1203926968	232	1203927869	197
75	1203924469	284	1203925339	376	1203926206	261	1203926972	236	1203927874	202
76	1203924481	296	1203925341	378	1203926214	269	1203926973	237	1203927875	203
77	1203924481	296	1203925345	382	1203926214	269	1203926973	237	1203927878	206
78	1203924490	305	1203925354	391	1203926218	273	1203926974	238	1203927883	211
79	1203924502	317	1203925360	397	1203926221	276	1203926979	243	1203927884	212
80	1203924503	318	1203925367	404	1203926229	284	1203926986	250	1203927888	216
81	1203924515	330	1203925371	408	1203926233	288	1203926993	257	1203927888	216
82	1203924522	337	1203925376	413	1203926235	290	1203926995	259	1203927895	223
83	1203924526	341	1203925376	413	1203926240	295	1203926995	259	1203927896	224
84	1203924537	352	1203925383	420	1203926241	296	1203926997	261	1203927900	228
85	1203924544	359	1203925387	424	1203926253	308	1203927000	264	1203927904	232
86	1203924548	363	1203925393	430	1203926256	311	1203927014	278	1203927905	233
87	1203924559	374	1203925397	434	1203926257	312	1203927015	279	1203927908	236
88	1203924608	423	1203925402	439	1203926259	314	1203927019	283	1203927909	237
89	1203924613	428	1203925416	453	1203926263	318	1203927020	284	1203927916	244
90	1203924625	440	1203925416	453	1203926278	333	1203927065	329	1203927917	245
91	1203924730	545	1203925453	490	1203926321	376	1203927079	343	1203927920	248
92	1203924730	545	1203925457	494	1203926322	377	1203927080	344	1203927969	297
		1		l		l .		l .		L

93	1203924742	557	1203925462	499	1203926325	380	1203927085	349	1203927969	297
94	1203924742	557	1203925463	500	1203926327	382	1203927085	349	1203927973	301
95	1203924745	560	1203925466	503	1203926344	399	1203927195	459	1203927973	301
96	1203924745	560	1203925486	523	1203926361	416	1203927195	459	1203927981	309
97	1203924775	590	1203925486	523	1203926406	461	1203927261	525	1203927982	310
98	1203924775	590	1203925652	689	1203926406	461	1203927261	525	1203927984	312
99	1203924834	649	1203925767	804	1203926578	633	1203927441	705	1203928322	650
100	1203924834	649	1203925767	804	1203926578	633	1203927441	705	1203928322	650

DOING T	andale Markin								·	
	code16 Machines	I	D 2	D 2 4	D 2	D 2 4	D 4	D 4 *	D 5	D 5 *
Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A
1	1202630770	0	1202632831	0	1202633679	0	1202634285	0	1202634857	0
2	1202630771	1	1202632831	0	1202633684	5	1202634286	1	1202634858	1
3	1202630771	1	1202632835	4	1202633684	5	1202634286	1	1202634862	5
4	1202630773	3	1202632836	5	1202633684	5	1202634287	2	1202634863	6
5	1202630775	5	1202632836	5	1202633684	5	1202634289	4	1202634864	7
6	1202630777	7	1202632836	5	1202633685	6	1202634289	4	1202634864	7
7	1202630777	7	1202632836	5	1202633685	6	1202634291	6	1202634864	7
8	1202630777	7	1202632837	6	1202633687	8	1202634291	6	1202634865	8
9	1202630777	7	1202632839	8	1202633687	8	1202634291	6	1202634866	9
10	1202630779	9	1202632839	8	1202633688	9	1202634293	8	1202634866	9
11	1202630781	11	1202632840	9	1202633691	12	1202634294	9	1202634866	9
12	1202630782	12	1202632841	10	1202633692	13	1202634294	9	1202634867	10
13	1202630783	13	1202632842	11	1202633692	13	1202634295	10	1202634867	10
14	1202630784	14	1202632851	20	1202633702	23	1202634295	10	1202634867	10
15	1202630787	17	1202632853	22	1202633704	25	1202634302	17	1202634882	25
16	1202630790	20	1202632855	24	1202633705	26	1202634306	21	1202634883	26
17	1202630792	22	1202632856	25	1202633706	27	1202634310	25	1202634884	27
18	1202630793	23	1202632856	25	1202633706	27	1202634310	25	1202634884	27
19	1202630794	24	1202632857	26	1202633706	27	1202634310	25	1202634884	27
20	1202630795	25	1202632858	27	1202633707	28	1202634311	26	1202634885	28
21	1202630797	27	1202632859	28	1202633708	29	1202634311	26	1202634886	29
22	1202630798	28	1202632859	28	1202633709	30	1202634313	28	1202634886	29
23	1202630798	28	1202632860	29	1202633711	32	1202634313	28	1202634887	30
24	1202630800	30	1202632860	29	1202633712	33	1202634315	30	1202634887	30
25	1202630803	33	1202632860	29	1202633715	36	1202634316	31	1202634888	31
26	1202630803	33	1202632863	32	1202633715	36	1202634316	31	1202634888	31
27	1202630804	34	1202632864	33	1202633721	42	1202634317	32	1202634903	46
28	1202630805	35	1202632872	41	1202633725	46	1202634318	33	1202634904	47
29	1202630805	35	1202632872	41	1202633725	46	1202634322	37	1202634905	48
30	1202630808	38	1202632874	43	1202633726	47	1202634326	41	1202634905	48
31	1202630810	40	1202632876	45	1202633726	47	1202634331	46	1202634905	48
32	1202630813	43	1202632878	47	1202633728	49	1202634331	46	1202634906	49
33	1202630817	47	1202632878	47	1202633729	50	1202634331	46	1202634906	49
34	1202630819	49	1202632879	48	1202633731	52	1202634331	46	1202634907	50
35	1202630819	49	1202632880	49	1202633732	53	1202634333	48	1202634907	50
36	1202630819	49	1202632880	49	1202633733	54	1202634334	49	1202634909	52
37	1202630819	49	1202632880	49	1202633735	56	1202634336	51	1202634910	53
38	1202630820	50	1202632882	51	1202633736	57	1202634337	52	1202634912	55
39	1202630822	52	1202632882	51	1202633736	57	1202634337	52	1202634912	55
40	1202630824	54	1202632885	54	1202633742	63	1202634337	52	1202634924	67
41	1202630825	55	1202632885	54	1202633746	67	1202634338	53	1202634926	69
42	1202630827	57	1202632892	61	1202633747	68	1202634340	55	1202634926	69
43	1202630828	58	1202632893	62	1202633747	68	1202634344	59	1202634926	69
44	1202630830	60	1202632895	64	1202633747	68	1202634348	63	1202634926	69
45	1202630831	61	1202632898	67	1202633748	69	1202634348	63	1202634927	70
46	1202630832	62	1202632898	67	1202633750	71	1202634351	66	1202634927	70
47	1202630834	64	1202632900	69	1202633751	72	1202634352	67	1202634928	71
48	1202630836	66	1202632901	70	1202633753	74	1202634353	68	1202634929	72
49	1202630839	69	1202632901	70	1202633756	77	1202634353	68	1202634930	73

50	1202(20040	70	1202622002	l	1202622776		1202624255	1.70	1202624020	
50	1202630840	70	1202632902	71	1202633756	77	1202634355	70	1202634930	73
51	1202630840	70	1202632902	71	1202633756	77	1202634357	72	1202634933	76
52	1202630841	71	1202632903	72	1202633757	78	1202634357	72	1202634933	76
53	1202630842	72	1202632904	73	1202633762	83	1202634358	73	1202634945	88
54	1202630844	74	1202632904	73	1202633766	87	1202634359	74	1202634947	90
55	1202630845	75	1202632905	74	1202633767	88	1202634362	77	1202634947	90
56	1202630850	80	1202632912	81	1202633769	90	1202634362	77	1202634947	90
57	1202630851	81	1202632914	83	1202633769	90	1202634364	79	1202634948	91
58	1202630852	82	1202632915	84	1202633771	92	1202634368	83	1202634949	92
59	1202630852	82	1202632919	88	1202633774	95	1202634372	87	1202634949	92
60	1202630853	83	1202632921	90	1202633775	96	1202634373	88	1202634950	93
61	1202630854	84	1202632921	90	1202633776	97	1202634374	89	1202634951	94
62	1202630857	87	1202632922	91	1202633777	98	1202634374	89	1202634954	97
63	1202630860	90	1202632922	91	1202633777	98	1202634374	89	1202634954	97
64	1202630862	92	1202632923	92	1202633779	100	1202634376	91	1202634954	97
65	1202630862	92	1202632924	93	1202633783	104	1202634378	93	1202634967	110
66	1202630862	92	1202632924	93	1202633784	105	1202634378	93	1202634968	111
67	1202630865	95	1202632927	96	1202633788	109	1202634379	94	1202634969	112
68	1202630866	96	1202632929	98	1202633789	110	1202634379	94	1202634969	112
69	1202630866	96	1202632929	98	1202633789	110	1202634384	99	1202634971	114
70	1202630872	102	1202632933	102	1202633790	111	1202634384	99	1202634972	115
71	1202630873	103	1202632935	104	1202633793	114	1202634385	100	1202634973	116
72	1202630875	105	1202632935	104	1202633794	115	1202634390	105	1202634974	117
73	1202630876	106	1202632940	109	1202633798	119	1202634393	108	1202634974	117
74	1202630879	109	1202632941	110	1202633798	119	1202634393	108	1202634975	118
75	1202630880	110	1202632942	111	1202633799	120	1202634394	109	1202634977	120
76	1202630881	111	1202632943	112	1202633800	121	1202634395	110	1202634988	131
77	1202630882	112	1202632943	112	1202633802	123	1202634395	110	1202634989	132
78	1202630884	114	1202632948	117	1202633804	125	1202634397	112	1202634991	134
79	1202630886	116	1202632948	117	1202633805	126	1202634399	114	1202634992	135
80	1202630886	116	1202632949	118	1202633809	130	1202634400	115	1202635037	180
81	1202630887	117	1202632995	164	1202633809	130	1202634400	115	1202635039	182
82	1202630887	117	1202632995	164	1202633810	131	1202634405	120	1202635039	182
83	1202630937	167	1202632998	167	1202633811	132	1202634405	120	1202635039	182
84	1202630938	168	1202632999	168	1202633858	179	1202634409	124	1202635041	184
85	1202630939	169	1202633000	169	1202633859	180	1202634449	164	1202635042	185
86	1202630941	171	1202633005	174	1202633862	183	1202634455	170	1202635053	196
87	1202630941	171	1202633006	175	1202633863	184	1202634458	173	1202635054	197
88	1202630943	173	1202633007	176	1202633863	184	1202634458	173	1202635056	199
89	1202630946	176	1202633007	176	1202633865	186	1202634459	174	1202635058	201
90	1202630946	176	1202633012	181	1202633868	189	1202634460	175	1202635180	323
91	1202630948	178	1202633014	183	1202633869	190	1202634460	175	1202635272	415
92	1202630949	179	1202633015	184	1202633869	190	1202634462	177	1202635274	417
93	1202630951	181	1202633134	303	1202633874	195	1202634464	179	1202635317	460
94	1202630951	181	1202633203	372	1202633874	195	1202634464	179	1202635317	460
95	1202630951	181	1202633206	375	1202633875	196	1202634465	180	1202635317	484
96	1202630953	183	1202633239	408	1202633875	196	1202634470	185	1202635341	484
97	1202631079	309	1202633295	464	1202633988	309	1202634471	186	1202635341	484
98	1202631079	455	1202633295	464	1202634048	369	1202634475	190	1202635357	500
99	1202631223	508	1202633273	540	1202634048	369	1202634595	310	1202635357	500
100	1202631278	508	1202633371	540	1202634048	402	1202634593	405	1202635337	523
100	1202031270	200	1202033371	540	1202034001	.02	1202034070	.03	1202033300	525

CompTorrent

CompTo	rrent Trans	scode 1 M	achine							
Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A

1	1206172266	ا ا	1206174728	ا	1206177022	١	1206190062	١,	1206191952	١
2	1206172366	0		0	1206177032	0	1206180062	0	1206181853	0
2	1206172359	7	1206174720	8	1206177025	7	1206180054	8	1206181846	7
3	1206172346	20	1206174708	20	1206177013	19	1206180042	20	1206181834	19
4	1206172336	30	1206174697	31	1206177003	29	1206180033	29	1206181823	30
5	1206172325	41	1206174686	42	1206176991	41	1206180021	41	1206181812	41
6	1206172313	53	1206174674	54	1206176979	53	1206180009	53	1206181800	53
7	1206172301	65	1206174663	65	1206176967	65	1206179997	65	1206181788	65
8	1206172290	76	1206174652	76	1206176956	76	1206179987	75	1206181777	76
9	1206172280	86	1206174642	86	1206176946	86	1206179977	85	1206181767	86
10	1206172268	98	1206174631	97	1206176934	98	1206179965	97	1206181755	98
11	1206172256	110	1206174619	109	1206176922	110	1206179954	108	1206181744	109
12	1206172247	119	1206174610	118	1206176914	118	1206179944	118	1206181734	119
13	1206172235	131	1206174598	130	1206176901	131	1206179933	129	1206181723	130
14	1206172226	140	1206174589	139	1206176892	140	1206179925	137	1206181714	139
15	1206172214	152	1206174577	151	1206176880	152	1206179913	149	1206181701	152
16	1206172202	164	1206174566	162	1206176868	164	1206179901	161	1206181689	164
17	1206172193	173	1206174558	170	1206176860	172	1206179892	170	1206181680	173
18	1206172181	185	1206174545	183	1206176847	185	1206179880	182	1206181668	185
19	1206172169	197	1206174534	194	1206176835	197	1206179869	193	1206181656	197
20	1206172158	208	1206174522	206	1206176823	209	1206179857	205	1206181644	209
21	1206172149	217	1206174513	215	1206176816	216	1206179848	214	1206181636	217
22	1206172137	229	1206174502	226	1206176802	230	1206179836	226	1206181624	229
23	1206172125	241	1206174490	238	1206176791	241	1206179824	238	1206181612	241
24	1206172113	253	1206174479	249	1206176779	253	1206179812	250	1206181600	253
25	1206172105	261	1206174470	258	1206176771	261	1206179804	258	1206181593	260
26	1206172092	274	1206174458	270	1206176778	274	1206179792	270	1206181582	271
27	1206172080	286	1206174446	282	1206176746	286	1206179780	282	1206181569	284
28	1206172071	295	1206174437	291	1206176746	296	1206179700	291	1206181560	293
29	1206172071	307	1206174426	302	1206176736	307	1206179771	303	1206181548	305
30	1206172047	319	1206174414	314	1206176723	320	1206179747	315	1206181546	317
		330				331		326		329
31	1206172036		1206174402	326	1206176701		1206179736		1206181524	
32	1206172028	338	1206174393	335	1206176691	341	1206179728	334	1206181517	336
33	1206172016	350	1206174381	347	1206176679	353	1206179715	347	1206181504	349
34	1206172004	362	1206174369	359	1206176667	365	1206179704	358	1206181492	361
35	1206171992	374	1206174358	370	1206176655	377	1206179694	368	1206181480	373
36	1206171983	383	1206174349	379	1206176648	384	1206179684	378	1206181471	382
37	1206171971	395	1206174337	391	1206176635	397	1206179672	390	1206181459	394
38	1206171960	406	1206174326	402	1206176624	408	1206179661	401	1206181448	405
39	1206171948	418	1206174314	414	1206176612	420	1206179649	413	1206181438	415
40	1206171936	430	1206174303	425	1206176600	432	1206179638	424	1206181425	428
41	1206171925	441	1206174292	436	1206176588	444	1206179627	435	1206181413	440
42	1206171917	449	1206174282	446	1206176580	452	1206179617	445	1206181404	449
43	1206171905	461	1206174272	456	1206176568	464	1206179606	456	1206181393	460
44	1206171893	473	1206174260	468	1206176556	476	1206179594	468	1206181380	473
45	1206171881	485	1206174248	480	1206176544	488	1206179583	479	1206181368	485
46	1206171873	493	1206174240	488	1206176536	496	1206179575	487	1206181362	491
47	1206171860	506	1206174228	500	1206176524	508	1206179563	499	1206181348	505
48	1206171849	517	1206174217	511	1206176513	519	1206179552	510	1206181337	516
49	1206171840	526	1206174208	520	1206176505	527	1206179543	519	1206181330	523
50	1206171829	537	1206174197	531	1206176492	540	1206179531	531	1206181317	536
51	1206171817	549	1206174185	543	1206176481	551	1206179519	543	1206181305	548
52	1206171805	561	1206174173	555	1206176468	564	1206179507	555	1206181293	560
53	1206171796	570	1206174166	562	1206176459	573	1206179500	562	1206181284	569
54	1206171784	582	1206174153	575	1206176448	584	1206179487	575	1206181272	581
55	1206171771	595	1206174141	587	1206176435	597	1206179475	587	1206181260	593
56	1206171760	606	1206174130	598	1206176423	609	1206179463	599	1206181249	604
57	1206171750	616	1206174120	608	1206176414	618	1206179454	608	1206181239	614
58	1206171738	628	1206174108	620	1206176402	630	1206179442	620	1206181227	626
59	1206171725	641	1206174094	634	1206176388	644	1206179428	634	1206181213	640
		<u> </u>		<u> </u>		1	1	<u> </u>	1	1

	1206171716	650	1206174004	Lau	1206176270	652	1206170410	L 642	1206101202	
60	1206171716	650	1206174084	644	1206176379	653	1206179419	643	1206181202	651
61	1206171703	663	1206174071	657	1206176366	666	1206179405	657	1206181190	663
62	1206171692	674	1206174061	667	1206176355	677	1206179395	667	1206181181	672
63	1206171681	685	1206174050	678	1206176345	687	1206179384	678	1206181171	682
64	1206171672	694	1206174043	685	1206176335	697	1206179375	687	1206181161	692
65	1206171661	705	1206174030	698	1206176324	708	1206179364	698	1206181150	703
66	1206171649	717	1206174018	710	1206176312	720	1206179351	711	1206181138	715
67	1206171638	728	1206174008	720	1206176301	731	1206179341	721	1206181127	726
68	1206171626	740	1206173996	732	1206176289	743	1206179329	733	1206181115	738
69	1206171616	750	1206173984	744	1206176276	756	1206179317	745	1206181103	750
70	1206171606	760	1206173976	752	1206176269	763	1206179310	752	1206181094	759
71	1206171594	772	1206173964	764	1206176256	776	1206179297	765	1206181082	771
72	1206171583	783	1206173954	774	1206176246	786	1206179286	776	1206181071	782
73	1206171572	794	1206173943	785	1206176234	798	1206179275	787	1206181060	793
74	1206171562	804	1206173933	795	1206176224	808	1206179265	797	1206181050	803
75	1206171552	814	1206173924	804	1206176215	817	1206179257	805	1206181040	813
76	1206171540	826	1206173911	817	1206176202	830	1206179244	818	1206181030	823
77	1206171528	838	1206173899	829	1206176190	842	1206179232	830	1206181017	836
78	1206171516	850	1206173887	841	1206176178	854	1206179219	843	1206181004	849
79	1206171508	858	1206173878	850	1206176168	864	1206179209	853	1206180994	859
80	1206171495	871	1206173866	862	1206176156	876	1206179197	865	1206180982	871
81	1206171482	884	1206173854	874	1206176144	888	1206179185	877	1206180970	883
82	1206171473	893	1206173844	884	1206176136	896	1206179177	885	1206180962	891
83	1206171461	905	1206173833	895	1206176123	909	1206179164	898	1206180951	902
84	1206171449	917	1206173821	907	1206176111	921	1206179152	910	1206180938	915
85	1206171437	929	1206173809	919	1206176099	933	1206179140	922	1206180926	927
86	1206171429	937	1206173800	928	1206176089	943	1206179132	930	1206180916	937
87	1206171416	950	1206173788	940	1206176077	955	1206179119	943	1206180904	949
88	1206171404	962	1206173775	953	1206176064	968	1206179107	955	1206180891	962
89	1206171396	970	1206173766	962	1206176055	977	1206179099	963	1206180883	970
90	1206171383	983	1206173754	974	1206176043	989	1206179086	976	1206180870	983
91	1206171372	994	1206173744	984	1206176033	999	1206179076	986	1206180862	991
92	1206171361	1005	1206173733	995	1206176021	1011	1206179064	998	1206180849	1004
93	1206171349	1017	1206173721	1007	1206176009	1023	1206179053	1009	1206180837	1016
94	1206171337	1029	1206173709	1019	1206175997	1035	1206179041	1021	1206180825	1028
95	1206171330	1036	1206173700	1028	1206175988	1044	1206179031	1031	1206180818	1035
96	1206171317	1049	1206173688	1040	1206175976	1056	1206179019	1043	1206180805	1048
97	1206171305	1061	1206173676	1052	1206175965	1067	1206179007	1055	1206180793	1060
98	1206171295	1071	1206173666	1062	1206175955	1077	1206178999	1063	1206180785	1068
99	1206171282	1084	1206173654	1074	1206175942	1090	1206178988	1074	1206180772	1081
100	1206171272	1094	1206173643	1085	1206175932	1100	1206178978	1084	1206180761	1092
<u> </u>	· · · · · · · ·	<u> </u>						_		

CompTo	rrent Tran	scode 2 M	achines							
Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A
1	1206243070	0	1206244180	0	1206246608	0	1206248217	0	1206249604	0
2	1206243065	5	1206244174	6	1206246607	1	1206248210	7	1206249598	6
3	1206243058	12	1206244172	8	1206246601	7	1206248204	13	1206249596	8
4	1206243058	12	1206244161	19	1206246594	14	1206248202	15	1206249586	18
5	1206243045	25	1206244161	19	1206246588	20	1206248191	26	1206249585	19
6	1206243044	26	1206244149	31	1206246583	25	1206248186	31	1206249574	30
7	1206243033	37	1206244147	33	1206246575	33	1206248176	41	1206249572	32
8	1206243031	39	1206244136	44	1206246571	37	1206248171	46	1206249562	42
9	1206243022	48	1206244135	45	1206246565	43	1206248161	56	1206249560	44
10	1206243020	50	1206244125	55	1206246561	47	1206248158	59	1206249552	52
11	1206243010	60	1206244122	58	1206246550	58	1206248148	69	1206249547	57
12	1206243006	64	1206244113	67	1206246549	59	1206248144	73	1206249539	65
13	1206243000	70	1206244111	69	1206246541	67	1206248136	81	1206249536	68
14	1206242994	76	1206244102	78	1206246537	71	1206248129	88	1206249528	76
15	1206242991	79	1206244101	79	1206246531	77	1206248125	92	1206249526	78

16	120/242222	00	120/244222	02	120/24/22:	0.4	1206249:::	102	1206240515	00
16	1206242980	90	1206244088	92	1206246524	84	1206248114	103	1206249515	91
17	1206242979	101	1206244088	102	1206246520 1206246510	98	1206248112 1206248103	114	1206249513 1206249502	102
		-								-
19	1206242966	104	1206244076	104	1206246510	98	1206248098	119	1206249500	104
20	1206242956	114	1206244064	116	1206246497	111	1206248087	130	1206249488	116
21	1206242955	115	1206244064	116	1206246497	111	1206248084	133	1206249487	117
22	1206242945	125	1206244055	125	1206246488	120	1206248074	143	1206249481	123
23	1206242942	128	1206244050	130	1206246483	125	1206248070	147	1206249474	130
24	1206242933	137	1206244043	137	1206246475	133	1206248058	159	1206249468	136
25	1206242930	140	1206244038	142	1206246470	138	1206248056	161	1206249461	143
26	1206242926	144	1206244035	145	1206246468	140	1206248047	170	1206249459	145
27	1206242916	154	1206244026	154	1206246456	152	1206248040	177	1206249447	157
28	1206242913	157	1206244022	158	1206246455	153	1206248032	185	1206249446	158
29	1206242906	164	1206244015	165	1206246446	162	1206248028	189	1206249437	167
30	1206242903	167	1206244011	169	1206246443	165	1206248019	198	1206249434	170
31	1206242893	177	1206244002	178	1206246433	175	1206248012	205	1206249423	181
32	1206242890	180	1206243999	181	1206246431	177	1206248006	211	1206249423	181
33	1206242881	189	1206243991	189	1206246423	185	1206248001	216	1206249413	191
34	1206242879	191	1206243990	190	1206246419	189	1206247991	226	1206249410	194
35	1206242869	201	1206243978	202	1206246410	198	1206247984	233	1206249400	204
36	1206242866	204	1206243976	204	1206246408	200	1206247977	240	1206249398	206
37	1206242858	212	1206243968	212	1206246400	208	1206247972	245	1206249389	215
38	1206242857	213	1206243966	214	1206246396	212	1206247962	255	1206249387	217
39	1206242845	225	1206243955	225	1206246388	220	1206247959	258	1206249377	227
40	1206242843	227	1206243953	227	1206246384	224	1206247948	269	1206249374	230
41	1206242832	238	1206243942	238	1206246374	234	1206247943	274	1206249365	239
42	1206242831	239	1206243941	239	1206246373	235	1206247935	282	1206249364	240
43	1206242822	248	1206243932	248	1206246363	245	1206247930	287	1206249355	249
44	1206242819	251	1206243929	251	1206246361	247	1206247923	294	1206249351	253
45	1206242810	260	1206243920	260	1206246351	257	1206247915	302	1206249343	261
46	1206242805	265	1206243916	264	1206246348	260	1206247908	309	1206249338	266
47	1206242801	269	1206243912	268	1206246343	265	1206247905	312	1206249335	269
48	1206242793	277	1206243902	278	1206246335	273	1206247894	323	1206249325	279
49	1206242790	280	1206243901	279	1206246332	276	1206247890	327	1206249323	281
50	1206242781	289	1206243892	288	1206246324	284	1206247882	335	1206249314	290
51	1206242779	291	1206243890	290	1206246322	286	1206247880	337	1206249311	293
52	1206242768	302	1206243880	300	1206246311	297	1206247867	350	1206249302	302
53	1206242766	304	1206243877	303	1206246309	299	1206247867	350	1206249298	306
54	1206242760	310	1206243871	309	1206246303	305	1206247854	363	1206249293	311
55	1206242753	317	1206243864	316	1206246296	312	1206247851	366	1206249285	319
56	1206242747	323	1206243859	321	1206246290	318	1206247839	378	1206249280	324
57	1206242741	329	1206243851	329	1206246283	325	1206247835	382	1206249272	332
58	1206242737	333	1206243849	331	1206246280	328	1206247828	389	1206249270	334
59	1206242727	343	1206243838	342	1206246270	338	1206247820	397	1206249258	346
60	1206242723	347	1206243836	344	1206246266	342	1206247812	405	1206249256	348
61	1206242716	354	1206243826	354	1206246259	349	1206247805	412	1206249245	359
62	1206242710	360	1206243823	357	1206246253	355	1206247797	420	1206249244	360
63	1206242716	366	1206243815	365	1206246247	361	1206247792	425	1206249234	370
64	1206242700	370	1206243813	367	1206246247	365	1206247792	432	1206249233	371
65	1206242700	376	1206243806	374	1206246237	371	1206247783	435	1206249224	380
66	1206242694	382	1206243800	379	1206246237	377	1206247782	447	1206249224	384
67	1206242688	390	1206243801	388	1206246231	384	1206247776	452	1206249220	393
		390								395
68	1206242678		1206243790	390	1206246221	387	1206247757	460	1206249209	
69	1206242667	403	1206243779	401	1206246210	398	1206247750	467	1206249199	405
70	1206242665	405	1206243778	402	1206246209	399	1206247744	473	1206249197	407
71	1206242655	415	1206243767	413	1206246198	410	1206247739	478	1206249191	413
72	1206242653	417	1206243766	414	1206246197	411	1206247730	487	1206249185	419
73	1206242643	427	1206243755	425	1206246186	422	1206247724	493	1206249180	424
74	1206242641	429	1206243753	427	1206246185	423	1206247716	501	1206249171	433

75	1206242635	435	1206243745	435	1206246176	432	1206247712	505	1206249169	435
76	1206242633	437	1206243743	437	1206246174	434	1206247704	513	1206249160	444
77	1206242621	449	1206243732	448	1206246163	445	1206247696	521	1206249158	446
78	1206242618	452	1206243729	451	1206246160	448	1206247689	528	1206249147	457
79	1206242608	462	1206243720	460	1206246150	458	1206247680	537	1206249145	459
80	1206242607	463	1206243718	462	1206246149	459	1206247677	540	1206249136	468
81	1206242596	474	1206243707	473	1206246138	470	1206247664	553	1206249131	473
82	1206242594	476	1206243705	475	1206246136	472	1206247663	554	1206249124	480
83	1206242587	483	1206243700	480	1206246129	479	1206247652	565	1206249122	482
84	1206242581	489	1206243692	488	1206246123	485	1206247647	570	1206249112	492
85	1206242574	496	1206243686	494	1206246116	492	1206247639	578	1206249108	496
86	1206242568	502	1206243678	502	1206246110	498	1206247632	585	1206249100	504
87	1206242565	505	1206243677	503	1206246107	501	1206247628	589	1206249100	504
88	1206242554	516	1206243664	516	1206246097	511	1206247616	601	1206249087	517
89	1206242552	518	1206243664	516	1206246094	514	1206247613	604	1206249085	519
90	1206242544	526	1206243656	524	1206246085	523	1206247603	614	1206249075	529
91	1206242541	529	1206243650	530	1206246083	525	1206247599	618	1206249075	529
92	1206242533	537	1206243644	536	1206246072	536	1206247589	628	1206249064	540
93	1206242530	540	1206243638	542	1206246071	537	1206247584	633	1206249061	543
94	1206242521	549	1206243632	548	1206246059	549	1206247576	641	1206249052	552
95	1206242516	554	1206243625	555	1206246057	551	1206247569	648	1206249048	556
96	1206242512	558	1206243623	557	1206246052	556	1206247564	653	1206249044	560
97	1206242503	567	1206243611	569	1206246041	567	1206247553	664	1206249035	569
98	1206242500	570	1206243611	569	1206246039	569	1206247551	666	1206249032	572
99	1206242490	580	1206243600	580	1206246029	579	1206247538	679	1206249022	582
100	1206242490	580	1206243598	582	1206246026	582	1206247533	684	1206249019	585

Comp	Γorrent Tran	scode 4 N	Machines							
Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A
1	1206251506	0	1206252819	0	1206254073	0	1206254927	0	1206256460	0
2	1206251503	3	1206252816	3	1206254070	3	1206254923	4	1206256460	0
3	1206251501	5	1206252813	6	1206254068	5	1206254922	5	1206256456	4
4	1206251497	9	1206252812	7	1206254067	6	1206254921	6	1206256453	7
5	1206251496	10	1206252810	9	1206254062	11	1206254917	10	1206256450	10
6	1206251492	14	1206252802	17	1206254054	19	1206254914	13	1206256447	13
7	1206251490	16	1206252801	18	1206254052	21	1206254910	17	1206256443	17
8	1206251482	24	1206252798	21	1206254052	21	1206254907	20	1206256440	20
9	1206251482	24	1206252796	23	1206254052	21	1206254904	23	1206256435	25
10	1206251478	28	1206252788	31	1206254038	35	1206254900	27	1206256433	27
11	1206251476	30	1206252785	34	1206254037	36	1206254895	32	1206256429	31
12	1206251470	36	1206252784	35	1206254036	37	1206254894	33	1206256427	33
13	1206251468	38	1206252780	39	1206254030	43	1206254892	35	1206256420	40
14	1206251465	41	1206252775	44	1206254026	47	1206254887	40	1206256419	41
15	1206251463	43	1206252774	45	1206254022	51	1206254885	42	1206256417	43
16	1206251458	48	1206252769	50	1206254021	52	1206254879	48	1206256410	50
17	1206251454	52	1206252768	51	1206254014	59	1206254877	50	1206256407	53
18	1206251453	53	1206252759	60	1206254012	61	1206254876	51	1206256404	56
19	1206251445	61	1206252757	62	1206254006	67	1206254873	54	1206256402	58
20	1206251445	61	1206252756	63	1206254005	68	1206254866	61	1206256395	65
21	1206251443	63	1206252753	66	1206254002	71	1206254863	64	1206256393	67
22	1206251439	67	1206252745	74	1206253994	79	1206254861	66	1206256390	70
23	1206251433	73	1206252742	77	1206253992	81	1206254859	68	1206256386	74
24	1206251431	75	1206252741	78	1206253990	83	1206254854	73	1206256381	79
25	1206251429	77	1206252740	79	1206253988	85	1206254851	76	1206256379	81
26	1206251426	80	1206252732	87	1206253980	93	1206254846	81	1206256378	82
27	1206251418	88	1206252727	92	1206253978	95	1206254845	82	1206256371	89
28	1206251418	88	1206252727	92	1206253978	95	1206254841	86	1206256367	93
29	1206251415	91	1206252725	94	1206253972	101	1206254838	89	1206256364	96
30	1206251412	94	1206252718	101	1206253965	108	1206254832	95	1206256363	97

21	1206251406	100	1206252712	106	1206252062	110	1206254922	05	1206256254	106
31		100	1206252713 1206252713	106	1206253963 1206253962	110	1206254832 1206254827	95	1206256354	106
	1206251403								1206256354	106
33	1206251401	105	1206252710	109	1206253955	118	1206254826	101	1206256351	109
34	1206251398	108	1206252705	114	1206253953	120	1206254820	107	1206256347	113
35	1206251391	115	1206252700	119	1206253945	128	1206254819	108	1206256340	120
36	1206251389	117	1206252697	122	1206253945	128	1206254813	114	1206256338	122
37	1206251389	117	1206252694	125	1206253943	130	1206254810	117	1206256335	125
38	1206251384	122	1206252691	128	1206253937	136	1206254806	121	1206256332	128
39	1206251380	126	1206252689	130	1206253936	137	1206254806	121	1206256328	132
40	1206251377	129	1206252681	138	1206253926	147	1206254803	124	1206256325	135
41	1206251375	131	1206252679	140	1206253926	147	1206254794	133	1206256323	137
42	1206251370	136	1206252677	142	1206253923	150	1206254791	136	1206256317	143
43	1206251366	140	1206252675	144	1206253919	154	1206254791	136	1206256313	147
44	1206251364	142	1206252668	151	1206253912	161	1206254791	136	1206256311	149
45	1206251362	144	1206252664	155	1206253910	163	1206254779	148	1206256311	149
46	1206251360	146	1206252662	157	1206253910	163	1206254779	148	1206256302	158
47	1206251352	154	1206252659	160	1206253901	172	1206254777	150	1206256298	162
48	1206251351	155	1206252654	165	1206253899	174	1206254775	152	1206256296	164
49	1206251350	156	1206252652	167	1206253897	176	1206254769	158	1206256296	164
50	1206251345	161	1206252649	170	1206253892	181	1206254766	161	1206256293	167
51	1206251339	167	1206252646	173	1206253888	185	1206254762	165	1206256285	175
52	1206251339	167	1206252640	179	1206253886	187	1206254762	165	1206256282	178
53	1206251337	169	1206252636	183	1206253882	191	1206254760	167	1206256280	180
54	1206251331	175	1206252635	184	1206253877	196	1206254754	173	1206256279	181
55	1206251325	181	1206252631	188	1206253875	198	1206254748	179	1206256271	189
56	1206251325	181	1206252624	195	1206253871	202	1206254747	180	1206256268	192
57	1206251323	183	1206252623	196	1206253869	204	1206254744	183	1206256266	194
58	1206251317	189	1206252620	199	1206253862	211	1206254741	186	1206256263	197
59	1206251317	193	1206252615	204	1206253862	213	1206254741	194	1206256257	203
60	1206251313	195	1206252612	207	1206253854	219	1206254733	194	1206256257	203
61	1206251309	197	1206252607	212	1206253852	221	1206254731	196	1206256248	212
62	1206251302	204	1206252602	217	1206253849	224	1206254725	202	1206256247	213
63	1206251300	206	1206252602	217	1206253847	226	1206254722	205	1206256242	218
64	1206251299	207	1206252598	221	1206253840	233	1206254719	208	1206256242	218
65	1206251296	210	1206252592	227	1206253839	234	1206254718	209	1206256233	227
66	1206251288	218	1206252589	230	1206253834	239	1206254711	216	1206256232	228
67	1206251286	220	1206252588	231	1206253829	244	1206254711	216	1206256230	230
68	1206251285	221	1206252584	235	1206253827	246	1206254706	221	1206256227	233
69	1206251282	224	1206252578	241	1206253823	250	1206254702	225	1206256219	241
70	1206251274	232	1206252575	244	1206253819	254	1206254697	230	1206256217	243
71	1206251272	234	1206252575	244	1206253818	255	1206254694	233	1206256217	243
72	1206251271	235	1206252570	249	1206253812	261	1206254694	233	1206256212	248
73	1206251264	242	1206252564	255	1206253811	262	1206254687	240	1206256204	256
74	1206251262	244	1206252562	257	1206253806	267	1206254686	241	1206256203	257
75	1206251261	245	1206252561	258	1206253806	267	1206254680	247	1206256200	260
76	1206251258	248	1206252555	264	1206253796	277	1206254677	250	1206256200	260
77	1206251252	254	1206252550	269	1206253794	279	1206254675	252	1206256192	268
78	1206251249	257	1206252546	273	1206253794	279	1206254673	254	1206256189	271
79	1206251247	259	1206252546	273	1206253789	284	1206254665	262	1206256184	276
80	1206251247	259	1206252540	279	1206253780	293	1206254663	264	1206256183	277
81	1206251236	270	1206252534	285	1206253778	295	1206254661	266	1206256176	284
82	1206251235	271	1206252531	288	1206253778	295	1206254661	266	1206256175	285
83	1206251233	273	1206252530	289	1206253771	302	1206254651	276	1206256169	291
84	1206251233	273	1206252528	291	1206253771	302	1206254648	279	1206256166	294
85	1206251225	281	1206252520	299	1206253762	311	1206254647	280	1206256162	298
86	1206251221	285	1206252525	304	1206253761	312	1206254643	284	1206256161	299
87	1206251221	287	1206252514	305	1206253761	317	1206254639	288	1206256155	305
88	1206251217	289	1206252514	307	1206253750	321	1206254633	294	1206256153	309
89	1206251215	291	1206252507	312	1206253748	325	1206254632	295	1206256150	310

90	1206251206	300	1206252499	320	1206253745	328	1206254629	298	1206256144	316
91	1206251204	302	1206252498	321	1206253744	329	1206254627	300	1206256140	320
92	1206251204	302	1206252496	323	1206253736	337	1206254620	307	1206256138	322
93	1206251202	304	1206252494	325	1206253735	338	1206254619	308	1206256133	327
94	1206251191	315	1206252484	335	1206253729	344	1206254615	312	1206256128	332
95	1206251190	316	1206252483	336	1206253728	345	1206254611	316	1206256127	333
96	1206251189	317	1206252481	338	1206253725	348	1206254606	321	1206256123	337
97	1206251184	322	1206252481	338	1206253717	356	1206254605	322	1206256117	343
98	1206251180	326	1206252470	349	1206253713	360	1206254596	331	1206256117	343
99	1206251177	329	1206252458	361	1206253701	372	1206254594	333	1206256103	357
100	1206251174	332	1206252458	361	1206253699	374	1206254593	334	1206256100	360

Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A
1	1206257747	0	1206258162	0	1206258746	0	1206259144	0	1206260258	0
2	1206257743	4	1206258159	3	1206258744	2	1206259142	2	1206260254	4
3	1206257743	4	1206258158	4	1206258743	3	1206259140	4	1206260252	6
4	1206257741	6	1206258156	6	1206258743	3	1206259139	5	1206260252	6
5	1206257739	8	1206258151	11	1206258743	3	1206259138	6	1206260250	8
6	1206257738	9	1206258149	13	1206258740	6	1206259135	9	1206260248	10
7	1206257735	12	1206258149	13	1206258738	8	1206259135	9	1206260246	12
8	1206257733	14	1206258147	15	1206258734	12	1206259135	9	1206260246	12
9	1206257732	15	1206258144	18	1206258734	12	1206259132	12	1206260243	15
10	1206257730	17	1206258143	19	1206258731	15	1206259127	17	1206260240	18
11	1206257728	19	1206258136	26	1206258729	17	1206259126	18	1206260238	20
12	1206257727	20	1206258136	26	1206258728	18	1206259126	18	1206260236	22
13	1206257722	25	1206258135	27	1206258727	19	1206259126	18	1206260234	24
14	1206257722	25	1206258133	29	1206258727	19	1206259122	22	1206260233	25
15	1206257722	25	1206258130	32	1206258722	24	1206259121	23	1206260231	27
16	1206257719	28	1206258129	33	1206258722	24	1206259120	24	1206260229	29
17	1206257717	30	1206258128	34	1206258719	27	1206259118	26	1206260229	29
18	1206257717	30	1206258122	40	1206258717	29	1206259111	33	1206260225	33
19	1206257716	31	1206258119	43	1206258717	29	1206259109	35	1206260224	34
20	1206257713	34	1206258119	43	1206258714	32	1206259109	35	1206260221	37
21	1206257708	39	1206258118	44	1206258714	32	1206259109	35	1206260220	38
22	1206257705	42	1206258113	49	1206258713	33	1206259105	39	1206260219	39
23	1206257703	44	1206258112	50	1206258709	37	1206259103	41	1206260217	41
24	1206257703	44	1206258108	54	1206258708	38	1206259102	42	1206260211	47
25	1206257700	47	1206258106	56	1206258704	42	1206259101	43	1206260211	47
26	1206257699	48	1206258105	57	1206258704	42	1206259097	47	1206260209	49
27	1206257697	50	1206258104	58	1206258701	45	1206259095	49	1206260208	50
28	1206257694	53	1206258098	64	1206258700	46	1206259093	51	1206260206	52
29	1206257692	55	1206258097	65	1206258698	48	1206259092	52	1206260205	53
30	1206257691	56	1206258093	69	1206258696	50	1206259091	53	1206260202	56
31	1206257689	58	1206258091	71	1206258694	52	1206259090	54	1206260199	59
32	1206257689	58	1206258089	73	1206258693	53	1206259088	56	1206260196	62
33	1206257685	62	1206258086	76	1206258692	54	1206259087	57	1206260196	62
34	1206257683	64	1206258084	78	1206258690	56	1206259083	61	1206260193	65
35	1206257680	67	1206258084	78	1206258688	58	1206259083	61	1206260192	66
36	1206257679	68	1206258082	80	1206258683	63	1206259078	66	1206260191	67
37	1206257676	71	1206258081	81	1206258683	63	1206259076	68	1206260185	73
38	1206257674	73	1206258077	85	1206258681	65	1206259076	68	1206260184	74
39	1206257674	73	1206258074	88	1206258679	67	1206259075	69	1206260181	77
40	1206257674	73	1206258072	90	1206258678	68	1206259072	72	1206260180	78
41	1206257670	77	1206258070	92	1206258677	69	1206259072	72	1206260179	79
42	1206257667	80	1206258068	94	1206258675	71	1206259071	73	1206260176	82
43	1206257664	83	1206258066	96	1206258673	73	1206259065	79	1206260173	85
44	1206257663	84	1206258062	100	1206258672	74	1206259063	81	1206260171	87

45	120/25755	96	120/250052	1,00	120/250555	l 00	120/250252	01	120/2/0:5:	0.7
45	1206257661	86	1206258062	100	1206258666	80	1206259063	81	1206260171	87
46	1206257660	87	1206258058	104	1206258666	80	1206259060	84	1206260167	91
47	1206257658	89	1206258057	105	1206258664	82	1206259058	86	1206260167	91
48	1206257658	89	1206258053	109	1206258663	83	1206259058	86	1206260162	96
49	1206257655	92	1206258053	109	1206258662	84	1206259056	88	1206260162	96
50	1206257653	94	1206258052	110	1206258661	85	1206259054	90	1206260160	98
51	1206257651	96	1206258045	117	1206258657	89	1206259051	93	1206260159	99
52	1206257648	99	1206258044	118	1206258657	89	1206259050	94	1206260159	99
53	1206257647	100	1206258043	119	1206258653	93	1206259048	96	1206260154	104
54	1206257644	103	1206258039	123	1206258652	94	1206259045	99	1206260152	106
55	1206257644	103	1206258038	124	1206258651	95	1206259044	100	1206260151	107
56	1206257642	105	1206258038	124	1206258650	96	1206259042	102	1206260145	113
57	1206257640	107	1206258037	125	1206258648	98	1206259041	103	1206260145	113
58	1206257637	110	1206258030	132	1206258646	100	1206259038	106	1206260142	116
59	1206257634	113	1206258028	134	1206258642	104	1206259035	109	1206260142	116
60	1206257631	116	1206258022	140	1206258641	105	1206259034	110	1206260141	117
61	1206257630	117	1206258022	140	1206258640	106	1206259032	112	1206260136	122
62	1206257629	118	1206258018	144	1206258637	109	1206259030	114	1206260135	123
63	1206257627	120	1206258018	144	1206258635	111	1206259028	116	1206260133	125
64	1206257625	122	1206258016	146	1206258635	111	1206259027	117	1206260131	127
65	1206257622	125	1206258014	148	1206258629	117	1206259026	118	1206260129	129
66	1206257620	127	1206258013	149	1206258628	118	1206259022	122	1206260128	130
67	1206257619	128	1206258005	157	1206258626	120	1206259020	124	1206260126	132
68	1206257619	128	1206258003	159	1206258624	122	1206259019	125	1206260121	137
69	1206257616	131	1206258003	159	1206258622	124	1206259017	127	1206260119	139
70	1206257615	132	1206258003	159	1206258621	125	1206259017	131	1206260118	140
71	1206257613	135	1206258000	162	1206258618	128	1206259013	131	1206260116	142
72	1206257608	139	1206258000	162	1206258613	133	1206259011	133	1206260115	143
73		141	1206257999	163		133	1206259011	135		145
	1206257606				1206258613				1206260113	
74	1206257605	142	1206257994	168	1206258611	135	1206259009	135	1206260112	146
75	1206257602	145	1206257991	171	1206258610	136	1206259005	139	1206260111	147
76	1206257602	145	1206257988	174	1206258610	136	1206259003	141	1206260109	149
77	1206257602	145	1206257984	178	1206258607	139	1206259003	141	1206260109	149
78	1206257598	149	1206257984	178	1206258607	139	1206259002	142	1206260102	156
79	1206257597	150	1206257979	183	1206258602	144	1206258996	148	1206260100	158
80	1206257592	155	1206257977	185	1206258597	149	1206258994	150	1206260100	158
81	1206257592	155	1206257975	187	1206258597	149	1206258993	151	1206260097	161
82	1206257589	158	1206257974	188	1206258594	152	1206258990	154	1206260095	163
83	1206257588	159	1206257973	189	1206258594	152	1206258988	156	1206260092	166
84	1206257585	162	1206257972	190	1206258591	155	1206258987	157	1206260091	167
85	1206257585	162	1206257965	197	1206258591	155	1206258985	159	1206260087	171
86	1206257582	165	1206257962	200	1206258589	157	1206258985	159	1206260086	172
87	1206257581	166	1206257960	202	1206258589	157	1206258982	162	1206260084	174
88	1206257581	166	1206257955	207	1206258581	165	1206258979	165	1206260082	176
89	1206257581	166	1206257953	209	1206258581	165	1206258977	167	1206260080	178
90	1206257573	174	1206257951	211	1206258579	167	1206258975	169	1206260078	180
91	1206257571	176	1206257951	211	1206258578	168	1206258973	171	1206260077	181
92	1206257571	176	1206257949	213	1206258578	168	1206258973	171	1206260071	187
93	1206257567	180	1206257947	215	1206258575	171	1206258971	173	1206260069	189
94	1206257564	183	1206257943	219	1206258572	174	1206258969	175	1206260069	189
95	1206257564	183	1206257942	220	1206258572	174	1206258966	178	1206260068	190
96	1206257557	190	1206257938	224	1206258566	180	1206258963	181	1206260065	193
97	1206257555	192	1206257936	226	1206258564	182	1206258960	184	1206260063	195
98	1206257555	192	1206257932	230	1206258563	183	1206258959	185	1206260060	198
99	1206257552	195	1206257924	238	1206258561	185	1206258959	185	1206260055	203
100	1206257552	195	1206257923	239	1206258560	186	1206258956	188	1206260055	203

CompTo	rrent Trans	scode 16 N	Machines							
Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A

1.	1206525	L	1207221:-:	١	1207222	١	1207222	L	1207222	١,
1	1206527649	0	1207201404	0	1207202139	0	1207202535	0	1207203278	0
2	1206527647	2	1207201401	3	1207202138	1	1207202534	1	1207203276	2
3	1206527645	4	1207201399	5	1207202137	2	1207202531	4	1207203275	3
4	1206527643	6	1207201395	9	1207202136	3	1207202529	6	1207203274	4
5	1206527641	8	1207201393	11	1207202134	5	1207202527	8	1207203266	12
6	1206527639	10	1207201392	12	1207202131	8	1207202526	9	1207203264	14
7	1206527637	12	1207201391	13	1207202130	9	1207202523	12	1207203263	15
8	1206527634	15	1207201388	16	1207202128	11	1207202521	14	1207203261	17
9	1206527630	19	1207201386	18	1207202127	12	1207202520	15	1207203260	18
10	1206527626	23	1207201381	23	1207202124	15	1207202520	15	1207203256	22
11	1206527624	25	1207201377	27	1207202124	15	1207202518	17	1207203255	23
12	1206527619	30	1207201375	29	1207202117		1207202513		1207203252	26
13	1206527618	31	1207201373	31	1207202116	23	1207202512	23	1207203248	30
14	1206527614	35	1207201371	33	1207202113	26	1207202509	26	1207203241	37
15	1206527610	39	1207201370	34	1207202113	26	1207202508	27	1207203239	39
16	1206527608	41	1207201366	38	1207202110	29	1207202507	28	1207203231	47
17	1206527605	50	1207201365	39	1207202109	30	1207202507	28	1207203229	49
18	1206527599	50	1207201364	40	1207202103	36	1207202498	37	1207203227	51
19	1206527597	52	1207201364	40	1207202103	36	1207202498	37	1207203225	53
20	1206527595	54	1207201362	42	1207202094	45	1207202497	38	1207203214	62
21	1206527594	55	1207201362	42	1207202092	47	1207202494	41	1207203214	64
22	1206527594	55	1207201360	44	1207202091	48	1207202493	42	1207203212	66
23	1206527593	56	1207201359	45	1207202088	51	1207202489	46	1207203211	67
24	1206527592	57	1207201354	50	1207202083	56	1207202488	47	1207203208	70
25	1206527584	65	1207201353	51	1207202079	60	1207202486	49	1207203202	76
26	1206527583	66	1207201351	53	1207202077	62	1207202484	51	1207203201	77
27	1206527582	67	1207201350	54	1207202075	64	1207202484	51	1207203201	77
28	1206527578	71	1207201345	59	1207202073	66	1207202483	52	1207203197	81
29	1206527578	71	1207201344	60	1207202072	67	1207202483	52	1207203195	83
30	1206527578	71	1207201342	62	1207202070	74	1207202477	58	1207203194	84
31	1206527576 1206527571	73	1207201340	65	1207202065	75	1207202477	58	1207203189	90
33	1206527570	79	1207201339	67	1207202064	75	1207202477	62	1207203188	90
34	1206527570	82	1207201337	69	1207202064	77	1207202473	63	1207203186	92
35	1206527563	86	1207201333	69	1207202062	81	1207202472	66	1207203186	94
36	1206527562	87	1207201333	71	1207202056	83	1207202469	66	1207203184	98
37	1206527559	90	1207201333	74	1207202055	84	1207202469	72	1207203180	100
38	1206527558	91	1207201330	75	1207202055	84	1207202463	72	1207203176	100
39	1206527558	91	1207201329	83	1207202033	90	1207202463	72	1207203176	102
40	1206527556	93	1207201321	84	1207202049	90	1207202463	77	1207203173	103
41	1206527553	96	1207201320	85	1207202049	93	1207202456	79	1207203171	107
42	1206527552	97	1207201319	86	1207202046	95	1207202450	83	1207203169	111
43	1206527548	101	1207201316	88	1207202044	97	1207202452	85	1207203167	113
44	1206527547	102	1207201318	91	1207202042	98	1207202450	85	1207203163	113
45	1206527547	102	1207201313	94	1207202041	98	1207202449	86	1207203164	120
46	1206527540	109	1207201310	95	1207202041	102	1207202444	91	1207203138	120
47	1206527539	110	1207201309	96	1207202037	102	1207202444	93	1207203155	123
48	1206527534	115	1207201308	97	1207202034	105	1207202442	94	1207203153	126
49	1206527533	116	1207201307	98	1207202034	103	1207202441	96	1207203132	128
50	1206527531	118	1207201300	102	1207202031	112	1207202437	98	1207203130	129
51	1206527528	121	1207201302	102	1207202027	113	1207202437	98	1207203144	134
52	1206527527	121	1207201301	105	1207202026	113	1207202437	103	1207203144	135
53	1206527523	126	1207201299	103	1207202026	115	1207202432	103	1207203143	137
54	1206527523	126	1207201296	108	1207202024	117	1207202428	107	1207203141	141
							1207202427			141
55	1206527521	128	1207201294	110	1207202016	123		110	1207203135	143
56	1206527518	131	1207201294	110	1207202016	123	1207202425	110	1207203134	
57	1206527517	132	1207201289	115	1207202015	124	1207202424	111	1207203132	146
58	1206527514	135	1207201289	115	1207202012	127	1207202414	121	1207203129	149
59	1206527513	136	1207201285	119	1207202011	128	1207202414	121	1207203129	149

60	1206527512	137	1207201284	120	1207202011	128	1207202413	122	1207203122	156
61	1206527506	143	1207201280	124	1207202009	130	1207202409	126	1207203120	158
62	1206527505	144	1207201279	125	1207202006	133	1207202409	126	1207203118	160
63	1206527504	145	1207201278	126	1207202003	136	1207202405	130	1207203115	163
64	1206527499	150	1207201277	127	1207202002	137	1207202405	130	1207203113	165
65	1206527497	152	1207201276	128	1207202000	139	1207202402	133	1207203113	165
66	1206527496	153	1207201272	132	1207201999	140	1207202400	135	1207203107	171
67	1206527494	155	1207201271	133	1207201996	143	1207202396	139	1207203105	173
68	1206527493	156	1207201270	134	1207201993	146	1207202396	139	1207203103	175
69	1206527492	157	1207201265	139	1207201992	147	1207202395	140	1207203102	176
70	1206527490	159	1207201264	140	1207201986	153	1207202395	140	1207203101	177
71	1206527488	161	1207201263	141	1207201984	155	1207202395	140	1207203096	182
72	1206527482	167	1207201261	143	1207201980	159	1207202390	145	1207203095	183
73	1206527480	169	1207201257	147	1207201979	160	1207202389	146	1207203092	186
74	1206527479	170	1207201257	147	1207201978	161	1207202388	147	1207203090	188
75	1206527479	170	1207201253	151	1207201977	162	1207202384	151	1207203089	189
76	1206527474	175	1207201253	151	1207201975	164	1207202380	155	1207203088	190
77	1206527473	176	1207201252	152	1207201974	165	1207202379	156	1207203088	190
78	1206527473	176	1207201250	154	1207201971	168	1207202370	165	1207203082	196
79	1206527467	182	1207201249	155	1207201970	169	1207202370	165	1207203076	202
80	1206527464	185	1207201246	158	1207201967	172	1207202369	166	1207203074	204
81	1206527460	189	1207201244	160	1207201963	176	1207202367	168	1207203073	205
82	1206527458	191	1207201240	164	1207201958	181	1207202365	170	1207203072	206
83	1206527458	191	1207201236	168	1207201957	182	1207202362	173	1207203071	207
84	1206527454	195	1207201235	169	1207201953	186	1207202361	174	1207203067	211
85	1206527453	196	1207201234	170	1207201952	187	1207202360	175	1207203066	212
86	1206527452	197	1207201234	170	1207201949	190	1207202354	181	1207203061	217
87	1206527449	200	1207201230	174	1207201947	192	1207202354	181	1207203060	218
88	1206527448	201	1207201225	179	1207201942	197	1207202352	183	1207203060	218
89	1206527444	205	1207201222	182	1207201939	200	1207202343	192	1207203057	221
90	1206527439	210	1207201220	184	1207201937	202	1207202342	193	1207203055	223
91	1206527436	213	1207201220	184	1207201935	204	1207202340	195	1207203048	230
92	1206527433	216	1207201220	184	1207201932	207	1207202339	196	1207203044	234
93	1206527432	217	1207201212	192	1207201931	208	1207202339	196	1207203042	236
94	1206527431	218	1207201209	195	1207201928	211	1207202332	203	1207203038	240
95	1206527426	223	1207201208	196	1207201927	212	1207202331	204	1207203038	240
96	1206527423	226	1207201206	198	1207201926	213	1207202326	209	1207203028	250
97	1206527422	227	1207201203	201	1207201920	219	1207202326	209	1207203023	255
98	1206527414	235	1207201196	208	1207201915	224	1207202318	217	1207203022	256
99	1206527405	244	1207201188	216	1207201911	228	1207202317	218	1207203020	258
100	1206527398	251	1207201184	220	1207201907	232	1207202316	219	1207203018	260

Mandelbrot

Condor

Condo	Condor Mandelbrot 1 Machine												
Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A			
1	1202179294	0	1202179833	0	1202180457	0	1202181184	0	1202266406	0			
2	1202179297	3	1202179834	1	1202180469	12	1202181186	2	1202266410	4			
3	1202179302	8	1202179838	5	1202180470	13	1202181189	5	1202266414	8			
4	1202179306	12	1202179842	9	1202180475	18	1202181192	8	1202266418	12			
5	1202179308	14	1202179845	12	1202180478	21	1202181194	10	1202266422	16			
6	1202179309	15	1202179848	15	1202180480	23	1202181197	13	1202266426	20			
7	1202179313	19	1202179849	16	1202180482	25	1202181199	15	1202266430	24			
8	1202179317	23	1202179853	20	1202180485	28	1202181202	18	1202266434	28			
9	1202179321	27	1202179857	24	1202180489	32	1202181204	20	1202266438	32			
10	1202179321	27	1202179860	27	1202180492	35	1202181207	23	1202266441	35			

11	1202179324	30	1202179861	28	1202180493	36	1202181208	24	1202266445	39
12	1202179324	34	1202179861	32	1202180493	41	1202181208	29	1202266445	43
13	1202179328	37	1202179869	36	1202180498	44	1202181213	29	1202266449	46
14	1202179334	40	1202179872	39	1202180505	48	1202181218	34	1202266456	50
15	1202179336				1202180505	48	1202181220	36	1202266460	
16	1202179337	43	1202179877	44	1202180511	54	1202181222	38	1202266463	57
17	1202179341	47	1202179881	48	1202180515	58	1202181225	41	1202266467	61
18	1202179345	51	1202179881	48	1202180515	58	1202181227	43	1202266471	65
19	1202179348	54	1202179885	52	1202180519	62	1202181230	46	1202266475	69
20	1202179349	55	1202179890	57	1202180523	66	1202181234	50	1202266479	73
21	1202179388	94	1202179893	60	1202180526	69	1202181236	52	1202266482	76
22	1202179392	98	1202179895	62	1202180527	70	1202181240	56	1202266486	80
23	1202179393	99	1202179896	63	1202180530	73	1202181242	58	1202266489	83
24	1202179399	105	1202179900	67	1202180534	77	1202181246	62	1202266493	87
25	1202179403	109	1202179904	71	1202180537	80	1202181248	64	1202266497	91
26	1202179404	110	1202179907	74	1202180540	83	1202181253	69	1202266501	95
27	1202179409	115	1202179908	75	1202180541	84	1202181255	71	1202266504	98
28	1202179413	119	1202179913	80	1202180546	89	1202181260	76	1202266508	102
29	1202179414	120	1202179916	83	1202180549	92	1202181262	78	1202266512	106
30	1202179417	123	1202179919	86	1202180550	93	1202181267	83	1202266516	110
31	1202179420	126	1202179920	87	1202180552	95	1202181268	84	1202266519	113
32	1202179424	130	1202179925	92	1202180557	100	1202181274	90	1202266523	117
33	1202179425	131	1202179928	95	1202180561	104	1202181275	91	1202266526	120
34	1202179428	134	1202179931	98	1202180561	104	1202181281	97	1202266530	124
35	1202179433	139	1202179932	99	1202180565	108	1202181282	98	1202266534	128
36	1202179437	143	1202179936	103	1202180570	113	1202181289	105	1202266538	132
37	1202179440	146	1202179940	107	1202180574	117	1202181291	107	1202266542	136
38	1202179440	146	1202179943	110	1202180576	119	1202181296	112	1202266545	139
39	1202179446	152	1202179944	111	1202180577	120	1202181298	114	1202266549	143
40	1202179450	156	1202179949	116	1202180582	125	1202181303	119	1202266553	147
41	1202179453	159	1202179952	119	1202180586	129	1202181305	121	1202266556	150
42	1202179453	159	1202179955	122	1202180587	130	1202181311	127	1202266560	154
43	1202179459	165	1202179956	123	1202180590	133	1202181312	128	1202266563	157
44	1202179463	169	1202179960	127	1202180595	138	1202181319	135	1202266567	161
45	1202179464	170	1202179964	131	1202180599	142	1202181321	137	1202266571	165
46	1202179467	173	1202179967	134	1202180601	144	1202181327	143	1202266575	169
47	1202179470	176	1202179968	135	1202180603	146	1202181329	145	1202266579	173
48	1202179474	180	1202179970	137	1202180606	149	1202181329	152	1202266583	177
49	1202179475	181	1202179975	142	1202180610	153	1202181337	153	1202266587	181
50	1202179473	184	1202179979	146	1202180613	156	1202181337	160	1202266591	185
									1202266595	189
51	1202179482	188	1202179981	148	1202180613	156	1202181346	162		
52	1202179485	191	1202179982	149	1202180618	161	1202181351	167	1202266599	193
53	1202179488	194	1202179986	153	1202180622	165	1202181353	169	1202266603	197
54	1202179489	195	1202179990	157	1202180625	168	1202181358	174	1202266608	202
55	1202179494	200	1202179992	159	1202180626	169	1202181360	176	1202266612	206
56	1202179497	203	1202179994	161	1202180628	171	1202181365	181	1202266617	211
57	1202179499	205	1202179997	164	1202180632	175	1202181367	183	1202266622	216
58	1202179501	207	1202180001	168	1202180636	179	1202181372	188	1202266627	221
59	1202179505	211	1202180004	171	1202180639	182	1202181374	190	1202266632	226
60	1202179509	215	1202180005	172	1202180640	183	1202181379	195	1202266637	231
61	1202179510	216	1202180075	242	1202180642	185	1202181381	197	1202266642	236
62	1202179512	218	1202180087	254	1202180647	190	1202181386	202	1202266647	241
63	1202179517	223	1202180089	256	1202180651	194	1202181387	203	1202266653	247
64	1202179521	227	1202180093	260	1202180654	197	1202181393	209	1202266658	252
65	1202179524	230	1202180097	264	1202180655	198	1202181394	210	1202266663	257
66	1202179526	232	1202180098	265	1202180660	203	1202181400	216	1202266668	262
67	1202179527	233	1202180101	268	1202180663	206	1202181402	218	1202266673	267
68	1202179531	237	1202180104	271	1202180664	207	1202181406	222	1202266678	272
		+							+	

70	1202179538	244	1202180111	278	1202180670	213	1202181411	227	1202266688	282
71	1202179538	244	1202180111	278	1202180670	213	1202181411	228	1202266688	282
										292
72	1202179541	247	1202180117	284	1202180675	218	1202181415	231	1202266698	
73	1202179545	251	1202180121	288	1202180678	221	1202181418	234	1202266703	297
74	1202179549	255	1202180124	291	1202180681	224	1202181419	235	1202266708	302
75	1202179550	256	1202180124	291	1202180685	228	1202181423	239	1202266712	306
76	1202179553	259	1202180130	297	1202180686	229	1202181424	240	1202266716	310
77	1202179557	263	1202180134	301	1202180688	231	1202181426	242	1202266720	314
78	1202179560	266	1202180134	301	1202180692	235	1202181429	245	1202266725	319
79	1202179563	269	1202180138	305	1202180695	238	1202181430	246	1202266730	324
80	1202179564	270	1202180142	309	1202180699	242	1202181433	249	1202266736	330
81	1202179568	274	1202180146	313	1202180700	243	1202181434	250	1202266741	335
82	1202179572	278	1202180146	313	1202180703	246	1202181437	253	1202266747	341
83	1202179575	281	1202180150	317	1202180706	249	1202181438	254	1202266753	347
84	1202179576	282	1202180153	320	1202180710	253	1202181441	257	1202266759	353
85	1202179581	287	1202180157	324	1202180713	256	1202181443	259	1202266765	359
86	1202179584	290	1202180160	327	1202180714	257	1202181444	260	1202266771	365
87	1202179587	293	1202180161	328	1202180714	261	1202181447	263	1202266778	372
88	1202179588	294	1202180166	333	1202180722	265	1202181449	265	1202266785	379
89	1202179593	299	1202180170	337	1202180726	269	1202181450	266	1202266792	386
90	1202179596	302	1202180170	337	1202180726	269	1202181453	269	1202266798	392
91	1202179599	305	1202180173	340	1202180729	272	1202181454	270	1202266805	399
92	1202179600	306	1202180178	345	1202180734	277	1202181457	273	1202266812	406
93	1202179604	310	1202180182	349	1202180737	280	1202181459	275	1202266819	413
94	1202179608	314	1202180182	349	1202180741	284	1202181460	276	1202266825	419
95	1202179612	318	1202180186	353	1202180741	284	1202181463	279	1202266832	426
96	1202179612	318	1202180189	356	1202180747	290	1202181464	280	1202266839	433
97	1202179615	321	1202180193	360	1202180751	294	1202181467	283	1202266846	440
98	1202179620	326	1202180196	363	1202180753	296	1202181470	286	1202266853	447
99	1202179623	329	1202180197	364	1202180754	297	1202181471	287	1202266860	454
100	1202179627	333	1202180202	369	1202180762	305	1202181474	290	1202266867	461
101	1202179627	333	1202180205	372	1202181057	600	1202181533	349	1202266874	468
102	1202179632	338	1202180207	374	1202181058	601	1202181535	351	1202266882	476
103	1202179636	342	1202180208	375	1202181061	604	1202181537	353	1202266889	483
		345		379		606			1202266897	491
104	1202179639		1202180212		1202181063		1202181539	355		
105	1202179640	346	1202180216	383	1202181065	608	1202181541	357	1202266906	500
106	1202179645	351	1202180219	386	1202181067	610	1202181543	359	1202266914	508
107	1202179648	354	1202180220	387	1202181069	612	1202181545	361	1202266922	516
108	1202179651	357	1202180222	389	1202181071	614	1202181547	363	1202266930	524
109	1202179651	357	1202180227	394	1202181073	616	1202181549	365	1202266938	532
110	1202179658	364	1202180231	398	1202181075	618	1202181551	367	1202266946	540
111	1202179662	368	1202180231	398	1202181077	620	1202181553	369	1202266953	547
112	1202179662	368	1202180235	402	1202181079	622	1202181555	371	1202266960	554
113	1202179665	371	1202180240	407	1202181081	624	1202181557	373	1202266967	561
114	1202179669	375	1202180244	411	1202181083	626	1202181559	375	1202266974	568
115	1202179673	379	1202180244	411	1202181085	628	1202181561	377	1202266982	576
116	1202179676	382	1202180248	415	1202181087	630	1202181563	379	1202266988	582
117	1202179676	382	1202180252	419	1202181089	632	1202181565	381	1202266994	588
118	1202179681	387	1202180255	422	1202181091	634	1202181567	383	1202267001	595
119	1202179685	391	1202180258	425	1202181092	635	1202181569	385	1202267008	602
120	1202179688	394	1202180259	426	1202181095	638	1202181509	388	1202267008	609
						639		389		
121	1202179710	416	1202180264	431	1202181096		1202181573		1202267021	615
122	1202179722	428	1202180267	434	1202181099	642	1202181576	392	1202267028	622
123	1202179724	430	1202180269	436	1202181101	644	1202181578	394	1202267034	628
124	1202179728	434	1202180271	438	1202181102	645	1202181580	396	1202267041	635
125	1202179731	437	1202180275	442	1202181105	648	1202181581	397	1202267048	642
126	1202179733	439	1202180279	446	1202181107	650	1202181584	400	1202267055	649
127	1202179735	441	1202180281	448	1202181108	651	1202181585	401	1202267061	655

	129	1202179742	448	1202180286	453	1202181112	655	1202181590	406	1202267072	666
131 1202179746 452 120218029 460 120218116 659 120218159 410 1202267081 675 132 1202179750 456 1202180294 461 1202181120 663 1202181595 411 1202267085 679											
1221 1202179750 456 1202180294 461 1202181120 663 1202181595 411 1202267085 679 133											
133 1202179754 460 1202180298 465 1202181121 664 120218198 414 1202267082 682 134											
134 1202179757 463 1202180302 469 1202181123 666 1202181601 417 1202267092 686 135 1202187758 464 1202180305 472 1202181126 669 1202181602 418 1202267096 690 136 1202187761 467 1202180305 472 1202181127 670 1202181605 421 1202267099 693 137 1202179764 470 120218011 478 1202181130 673 1202181606 422 1202267103 697 138 1202179776 473 1202180315 482 1202181131 674 1202181609 425 1202267110 704 1302189707 477 1202180319 486 1202181137 680 1202181614 429 1202267114 708 140 1202179774 480 1202180319 486 1202181137 680 1202181614 430 1202267114 708 141 1202179774 480 1202180322 489 1202181138 681 1202181614 430 1202267114 711 142 120217978 484 1202180327 489 1202181148 684 1202181617 433 1202267112 715 143 120217978 488 1202180327 494 1202181148 688 1202181614 430 1202267123 719 144 1202179786 492 1202180333 500 1202181148 688 1202181624 437 1202267123 726 146 120217978 492 1202180333 500 1202181148 691 1202181624 437 1202267123 726 146 120217979 496 1202180333 500 1202181150 693 1202181624 448 1202267133 733 148 120217979 503 1202180333 506 1202181150 693 1202181629 445 1202267139 733 148 120217979 503 1202180334 508 1202181150 699 1202181629 445 1202267134 737 149 120217979 503 120218035 517 1202181150 699 1202181623 448 1202267134 737 149 1202179780 514 1202180350 517 1202181162 705 1202181624 450 1202267144 748 151 1202179815 519 1202181162 705 1202181624 450 1202267145 748 151 120217981 517 1202180350 517 1202181162 705 1202181624 450 1202267145 748 151 120217981 517 120218036 527 1202181164 707 1202181644 460 1202267157 751 151 151 120217981 523 120218036 530 1202181164											
135 1202179758 464 1202180305 472 1202181126 669 1202181602 418 1202267096 690 136											
136 1202179761 467 1202180305 472 1202181127 670 1202181605 421 1202267099 693 137 1202179764 470 1202180311 478 120218130 673 1202181606 422 1202267103 697 138 1202179767 473 1202180315 482 120218131 674 1202181609 425 1202267106 700 139 1202179770 476 1202180316 483 120218134 677 1202181610 426 1202267110 704 140 1202179771 477 1202180319 486 120218133 680 1202181613 429 1202267114 708 141 1202179774 480 1202180322 489 120218138 681 1202181614 430 1202267117 711 142 1202179778 484 1202180326 493 1202181140 684 1202181617 433 120226712 715 143 1202179786 492 1202180329 496 1202181142 685 1202181617 435 1202267125 719 144 1202179786 492 1202180339 506 1202181148 688 120218162 488 1202267128 722 145 1202179786 492 1202180339 506 1202181148 691 120218162 441 1202267136 730 147 1202179798 500 1202180339 506 1202181150 693 120218162 441 1202267136 730 148 1202179798 504 1202180348 516 120218156 699 120218162 448 1202267147 741 150 1202179805 511 1202180348 515 1202181150 699 120218163 448 1202267147 741 151 1202179805 511 1202180348 515 1202181150 699 120218163 448 1202267147 741 151 1202179805 511 1202180348 515 1202181160 703 120218163 449 1202267147 741 151 1202179805 514 1202180356 527 1202181160 703 120218164 460 1202267160 754 153 1202179817 523 120218036 530 1202181160 701 120218164 460 1202267160 754 156 120217982 528 120218036 530 1202181170 714 120218165 469 1202267170 761 156 120217982 528 120218036 530 1202181170 714 120218165 469 1202267170 761 156 120217982 528 120218036 530 1202181170 714 120218165 469 1202267170 761 157 1202											
137 1202179764 470 1202180311 478 1202181130 673 1202181606 422 1202267103 697 138											
138 1202179767 473 1202180315 482 1202181131 674 1202181609 425 1202267106 700 139											
199 1202179770 476 1202180316 483 1202181137 680 1202181610 426 1202267110 704 140	137	1202179764	470	1202180311		1202181130	673	1202181606	422	1202267103	697
140 1202179771 477 1202180319 486 1202181137 680 1202181613 429 1202267114 708 141 1202179774 480 1202180322 489 1202181138 681 1202181614 430 1202267117 711 142 1202179778 484 1202180326 493 1202181141 684 1202181617 433 1202267125 719 143 1202179786 492 1202180329 496 1202181145 685 1202181621 437 1202267128 722 144 1202179786 492 1202180333 500 1202181148 691 1202181622 438 1202267132 726 146 1202179790 496 120218033 504 1202181150 693 120218622 438 1202267132 726 147 1202179794 500 1202180333 506 1202181150 693 1202181622 441 1202267132 73 148 1202179794	138	1202179767	473	1202180315	482	1202181131	674	1202181609	425	1202267106	700
141 1202179774 480 1202180322 489 1202181138 681 1202181614 430 1202267117 711 142	139	1202179770	476	1202180316	483	1202181134	677	1202181610	426	1202267110	704
142 1202179778 484 1202180326 493 1202181141 684 1202181617 433 1202267121 715 143 1202179782 488 1202180327 494 1202181142 685 1202181619 435 1202267125 719 144 1202179786 492 1202180329 496 1202181148 681 1202181621 437 1202267128 722 145 1202179786 492 1202180333 500 1202181148 691 1202181622 438 1202267123 726 146 1202179790 496 1202180337 504 1202181150 693 1202181625 441 1202267132 726 147 1202179794 500 1202180337 506 1202181150 693 1202181626 442 1202267130 730 147 1202179794 500 1202180341 508 1202181154 697 1202181626 442 1202267143 737 149 1202179	140	1202179771	477	1202180319	486	1202181137	680	1202181613	429	1202267114	708
143 1202179782 488 1202180327 494 1202181142 685 1202181619 435 1202267125 719 144 1202179786 492 1202180329 496 1202181145 688 1202181621 437 1202267128 722 145 1202179786 492 1202180333 500 1202181169 693 1202181622 438 1202267128 722 146 1202179790 496 1202180337 504 1202181150 693 1202181625 441 1202267132 726 147 1202179794 500 1202180339 506 1202181151 694 1202181626 442 1202267139 733 148 1202179797 503 1202180341 508 1202181154 697 1202181629 445 1202267143 737 149 1202179798 504 1202180344 511 1202181158 699 1202181629 448 1202267147 741 150 1202179	141	1202179774	480	1202180322	489	1202181138	681	1202181614	430	1202267117	711
144 1202179786 492 1202180329 496 1202181145 688 1202181621 437 1202267128 722 145 1202179786 492 1202180333 500 1202181148 691 1202181622 438 1202267132 726 146 1202179790 496 1202180337 504 1202181150 693 1202181625 441 1202267136 730 147 1202179794 500 1202180331 506 1202181151 694 1202181626 442 1202267139 733 148 1202179797 503 1202180341 508 1202181154 697 1202181629 445 1202267143 737 149 1202179801 507 1202180344 511 1202181156 699 1202181632 448 1202267147 741 150 1202179801 507 1202180348 515 1202181158 701 1202181634 450 1202267150 744 151 1202179	142	1202179778	484	1202180326	493	1202181141	684	1202181617	433	1202267121	715
145 1202179786 492 1202180333 500 1202181148 691 1202181622 438 1202267132 726 146 1202179790 496 1202180337 504 1202181150 693 1202181625 441 1202267136 730 147 1202179794 500 1202180341 508 1202181151 694 1202181626 442 1202267139 733 148 1202179797 503 1202180341 508 1202181154 697 1202181629 445 1202267143 737 149 1202179798 504 1202180344 511 1202181156 699 1202181632 448 1202267147 741 150 1202179801 507 1202180348 515 1202181158 701 1202181634 450 1202267150 744 151 1202179805 511 1202180350 517 1202181160 703 1202181634 451 1202267154 748 152 1202179	143	1202179782	488	1202180327	494	1202181142	685	1202181619	435	1202267125	719
146 1202179790 496 1202180337 504 1202181150 693 1202181625 441 1202267136 730 147 1202179794 500 1202180339 506 1202181151 694 1202181626 442 1202267139 733 148 1202179797 503 1202180341 508 1202181154 697 1202181629 445 1202267143 737 149 1202179798 504 1202180344 511 1202181156 699 1202181632 448 1202267147 741 150 1202179801 507 1202180348 515 1202181158 701 1202181634 450 1202267150 744 151 1202179805 511 1202180350 517 1202181160 703 1202181635 451 1202267157 751 152 1202179808 514 1202180355 522 1202181162 705 1202181637 453 1202267157 751 153 1202179	144	1202179786	492	1202180329	496	1202181145	688	1202181621	437	1202267128	722
147 1202179794 500 1202180339 506 1202181151 694 1202181626 442 1202267139 733 148 1202179797 503 1202180341 508 1202181154 697 1202181629 445 1202267143 737 149 1202179798 504 1202180344 511 1202181156 699 1202181632 448 1202267147 741 150 1202179801 507 1202180348 515 1202181158 701 1202181632 448 1202267150 744 151 1202179805 511 1202180350 517 1202181160 703 1202181635 451 1202267154 748 152 1202179808 514 1202180352 519 1202181162 705 1202181637 453 1202267157 751 153 1202179811 517 1202180355 522 1202181164 707 1202181640 456 1202267160 754 154 1202179	145	1202179786	492	1202180333	500	1202181148	691	1202181622	438	1202267132	726
148 1202179797 503 1202180341 508 1202181154 697 1202181629 445 1202267143 737 149 1202179798 504 1202180344 511 1202181156 699 1202181632 448 1202267147 741 150 1202179801 507 1202180348 515 1202181158 701 1202181634 450 1202267150 744 151 1202179805 511 1202180350 517 1202181160 703 1202181635 451 1202267154 748 152 1202179808 514 1202180352 519 1202181162 705 1202181637 453 1202267157 751 153 1202179811 517 1202180355 522 1202181164 707 1202181640 456 1202267160 754 154 1202179812 518 1202180359 526 1202181166 709 1202181643 459 1202267167 761 155 1202179	146	1202179790	496	1202180337	504	1202181150	693	1202181625	441	1202267136	730
149 1202179798 504 1202180344 511 1202181156 699 1202181632 448 1202267147 741 150 1202179801 507 1202180348 515 1202181158 701 1202181634 450 1202267150 744 151 1202179805 511 1202180350 517 1202181160 703 1202181635 451 1202267154 748 152 1202179808 514 1202180352 519 1202181162 705 1202181637 453 1202267157 751 153 1202179811 517 1202180355 522 1202181164 707 1202181640 456 1202267160 754 154 1202179812 518 1202180359 526 1202181166 709 1202181643 459 1202267164 758 155 1202179817 523 1202180363 530 1202181169 712 1202181646 462 1202267167 761 156 1202179	147	1202179794	500	1202180339	506	1202181151	694	1202181626	442	1202267139	733
150 1202179801 507 1202180348 515 1202181158 701 1202181634 450 1202267150 744 151 1202179805 511 1202180350 517 1202181160 703 1202181635 451 1202267154 748 152 1202179808 514 1202180352 519 1202181162 705 1202181637 453 1202267157 751 153 1202179811 517 1202180355 522 1202181164 707 1202181640 456 1202267160 754 154 1202179812 518 1202180359 526 1202181166 709 1202181643 459 1202267164 758 155 1202179817 523 1202180360 527 1202181169 712 1202181646 462 1202267167 761 156 1202179820 526 1202180363 530 1202181171 714 1202181648 464 1202267171 765 157 1202179	148	1202179797	503	1202180341	508	1202181154	697	1202181629	445	1202267143	737
151 1202179805 511 1202180350 517 1202181160 703 1202181635 451 1202267154 748 152 1202179808 514 1202180352 519 1202181162 705 1202181637 453 1202267157 751 153 1202179811 517 1202180355 522 1202181164 707 1202181640 456 1202267160 754 154 1202179812 518 1202180359 526 1202181166 709 1202181643 459 1202267164 758 155 1202179817 523 1202180360 527 1202181169 712 1202181646 462 1202267167 761 156 1202179820 526 1202180363 530 1202181171 714 1202181648 464 1202267171 765 157 1202179822 528 1202180367 534 1202181174 717 1202181651 467 1202267175 769 158 1202179	149	1202179798	504	1202180344	511	1202181156	699	1202181632	448	1202267147	741
152 1202179808 514 1202180352 519 1202181162 705 1202181637 453 1202267157 751 153 1202179811 517 1202180355 522 1202181164 707 1202181640 456 1202267160 754 154 1202179812 518 1202180359 526 1202181166 709 1202181643 459 1202267164 758 155 1202179817 523 1202180360 527 1202181169 712 1202181646 462 1202267167 761 156 1202179820 526 1202180363 530 1202181171 714 1202181648 464 1202267171 765 157 1202179822 528 1202180367 534 1202181174 717 1202181651 467 1202267175 769 158 1202179824 530 1202180373 540 1202181179 722 1202181650 472 1202267182 776	150	1202179801	507	1202180348	515	1202181158	701	1202181634	450	1202267150	744
153 1202179811 517 1202180355 522 1202181164 707 1202181640 456 1202267160 754 154 1202179812 518 1202180359 526 1202181166 709 1202181643 459 1202267164 758 155 1202179817 523 1202180360 527 1202181169 712 1202181646 462 1202267167 761 156 1202179820 526 1202180363 530 1202181171 714 1202181648 464 1202267171 765 157 1202179822 528 1202180367 534 1202181174 717 1202181651 467 1202267175 769 158 1202179824 530 1202180370 537 1202181176 719 1202181653 469 1202267178 772 159 1202179828 534 1202180373 540 1202181179 722 1202181656 472 1202267182 776	151	1202179805	511	1202180350	517	1202181160	703	1202181635	451	1202267154	748
154 1202179812 518 1202180359 526 1202181166 709 1202181643 459 1202267164 758 155 1202179817 523 1202180360 527 1202181169 712 1202181646 462 1202267167 761 156 1202179820 526 1202180363 530 1202181171 714 1202181648 464 1202267171 765 157 1202179822 528 1202180367 534 1202181174 717 1202181651 467 1202267175 769 158 1202179824 530 1202180370 537 1202181176 719 1202181653 469 1202267178 772 159 1202179828 534 1202180373 540 1202181179 722 1202181656 472 1202267182 776	152	1202179808	514	1202180352	519	1202181162	705	1202181637	453	1202267157	751
155 1202179817 523 1202180360 527 1202181169 712 1202181646 462 1202267167 761 156 1202179820 526 1202180363 530 1202181171 714 1202181648 464 1202267171 765 157 1202179822 528 1202180367 534 1202181174 717 1202181651 467 1202267175 769 158 1202179824 530 1202180370 537 1202181176 719 1202181653 469 1202267178 772 159 1202179828 534 1202180373 540 1202181179 722 1202181656 472 1202267182 776	153	1202179811	517	1202180355	522	1202181164	707	1202181640	456	1202267160	754
156 1202179820 526 1202180363 530 1202181171 714 1202181648 464 1202267171 765 157 1202179822 528 1202180367 534 1202181174 717 1202181651 467 1202267175 769 158 1202179824 530 1202180370 537 1202181176 719 1202181653 469 1202267178 772 159 1202179828 534 1202180373 540 1202181179 722 1202181656 472 1202267182 776	154	1202179812	518	1202180359	526	1202181166	709	1202181643	459	1202267164	758
157 1202179822 528 1202180367 534 1202181174 717 1202181651 467 1202267175 769 158 1202179824 530 1202180370 537 1202181176 719 1202181653 469 1202267178 772 159 1202179828 534 1202180373 540 1202181179 722 1202181656 472 1202267182 776	155	1202179817	523	1202180360	527	1202181169	712	1202181646	462	1202267167	761
158 1202179824 530 1202180370 537 1202181176 719 1202181653 469 1202267178 772 159 1202179828 534 1202180373 540 1202181179 722 1202181656 472 1202267182 776	156	1202179820	526	1202180363	530	1202181171	714	1202181648	464	1202267171	765
159 1202179828 534 1202180373 540 1202181179 722 1202181656 472 1202267182 776	157	1202179822	528	1202180367	534	1202181174	717	1202181651	467	1202267175	769
	158	1202179824	530	1202180370	537	1202181176	719	1202181653	469	1202267178	772
160 1202179831 537 1202180374 541 1202181181 724 1202181657 473 1202267185 779	159	1202179828	534	1202180373	540	1202181179	722	1202181656	472	1202267182	776
	160	1202179831	537	1202180374	541	1202181181	724	1202181657	473	1202267185	779

Condor	Mandelbro	t 2 Machi	nes							
Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A
1	1202207136	0	1202207592	0	1202208066	0	1202208530	0	1202208966	0
2	1202207141	5	1202207596	4	1202208070	4	1202208533	3	1202208968	2
3	1202207145	9	1202207600	8	1202208074	8	1202208535	5	1202208970	4
4	1202207149	13	1202207604	12	1202208078	12	1202208536	6	1202208972	6
5	1202207153	17	1202207608	16	1202208082	16	1202208539	9	1202208974	8
6	1202207157	21	1202207612	20	1202208086	20	1202208540	10	1202208976	10
7	1202207159	23	1202207614	22	1202208088	22	1202208543	13	1202208978	12
8	1202207161	25	1202207616	24	1202208090	24	1202208544	14	1202208980	14
9	1202207163	27	1202207619	27	1202208092	26	1202208547	17	1202208982	16
10	1202207165	29	1202207620	28	1202208094	28	1202208548	18	1202208985	19
11	1202207167	31	1202207623	31	1202208096	30	1202208550	20	1202208986	20
12	1202207169	33	1202207624	32	1202208098	32	1202208552	22	1202208989	23
13	1202207171	35	1202207627	35	1202208099	33	1202208555	25	1202208991	25
14	1202207172	36	1202207628	36	1202208102	36	1202208556	26	1202208993	27
15	1202207174	38	1202207631	39	1202208104	38	1202208559	29	1202208995	29
16	1202207177	41	1202207632	40	1202208107	41	1202208560	30	1202208997	31
17	1202207178	42	1202207636	44	1202208108	42	1202208563	33	1202208999	33
18	1202207181	45	1202207637	45	1202208111	45	1202208564	34	1202209002	36
19	1202207183	47	1202207640	48	1202208113	47	1202208567	37	1202209003	37
20	1202207184	48	1202207641	49	1202208114	48	1202208569	39	1202209006	40
21	1202207187	51	1202207643	51	1202208117	51	1202208570	40	1202209008	42
22	1202207188	52	1202207646	54	1202208118	52	1202208573	43	1202209009	43
23	1202207191	55	1202207648	56	1202208120	54	1202208574	44	1202209012	46
24	1202207193	57	1202207649	57	1202208123	57	1202208577	47	1202209013	47

25	1202207194	58	1202207652	60	1202208125	59	1202208579	49	1202209016	50
26	1202207197	61	1202207653	61	1202208126	60	1202208580	50	1202209017	51
27	1202207198	62	1202207656	64	1202208129	63	1202208583	53	1202209020	54
28	1202207201	65	1202207658	66	1202208130	64	1202208585	55	1202209021	55
29	1202207202	66	1202207660	68	1202208134	68	1202208586	56	1202209024	58
30	1202207205	69	1202207661	69	1202208134	68	1202208588	58	1202209026	60
31	1202207206	70	1202207664	72	1202208138	72	1202208590	60	1202209027	61
32	1202207209	73	1202207667	75	1202208140	74	1202208593	63	1202209030	64
									1202209030	
33	1202207210	74	1202207669	77	1202208141	75	1202208595	65		65
34	1202207213	77	1202207670	78	1202208144	78	1202208597	67	1202209034	68
35	1202207214	78	1202207673	81	1202208146	80	1202208600	70	1202209035	69
36	1202207217	81	1202207674	82	1202208147	81	1202208601	71	1202209038	72
37	1202207219	83	1202207677	85	1202208150	84	1202208604	74	1202209040	74
38	1202207220	84	1202207678	86	1202208151	85	1202208605	75	1202209042	76
39	1202207223	87	1202207681	89	1202208154	88	1202208608	78	1202209043	77
40	1202207224	88	1202207682	90	1202208155	89	1202208609	79	1202209045	79
41	1202207227	91	1202207685	93	1202208158	92	1202208612	82	1202209048	82
42	1202207228	92	1202207687	95	1202208160	94	1202208613	83	1202209049	83
43	1202207231	95	1202207688	96	1202208162	96	1202208616	86	1202209052	86
44	1202207232	96	1202207691	99	1202208163	97	1202208619	89	1202209053	87
45	1202207235	99	1202207694	102	1202208166	100	1202208620	90	1202209056	90
46	1202207236	100	1202207696	104	1202208168	102	1202208622	92	1202209057	91
47	1202207239	103	1202207697	105	1202208169	103	1202208624	94	1202209061	95
48	1202207241	105	1202207701	109	1202208172	106	1202208627	97	1202209061	95
49	1202207243	107	1202207702	110	1202208176	110	1202208628	98	1202209065	99
50	1202207245	109	1202207704	112	1202208178	112	1202208632	102	1202209068	102
51	1202207247	111	1202207706	114	1202208180	114	1202208636	106	1202209070	104
52	1202207249	113	1202207708	116	1202208182	116	1202208637	107	1202209071	105
53	1202207251	115	1202207712	120	1202208184	118	1202208640	110	1202209074	108
54	1202207253	117	1202207713	121	1202208187	121	1202208642	112	1202209076	110
55	1202207256	120	1202207717	125	1202208189	123	1202208645	115	1202209079	113
56	1202207258	122	1202207719	127	1202208191	125	1202208648	118	1202209081	115
57	1202207261	125	1202207722	130	1202208194	128	1202208650	120	1202209084	118
58	1202207263	127	1202207724	132	1202208196	130	1202208653	123	1202209086	120
59	1202207266	130	1202207727	135	1202208199	133	1202208655	125	1202209089	123
60	1202207268	132	1202207728	136	1202208200	134	1202208658	128	1202209092	126
61	1202207271	135	1202207732	140	1202208204	138	1202208660	130	1202209094	128
62	1202207273	137	1202207734	142	1202208207	141	1202208663	133	1202209097	131
63	1202207276	140	1202207737	145	1202208210	144	1202208665	135	1202209099	133
64	1202207279	143	1202207740	148	1202208210	146	1202208669	139	1202209102	136
65	1202207273	145	1202207740	150	1202208212	149	1202208669	139	1202209102	139
	1202207281	143		153				144		141
66			1202207745	155	1202208216	150	1202208674		1202209107	141
67	1202207286	150	1202207747		1202208219	153	1202208676	146	1202209110	
68	1202207289	153	1202207750	158	1202208222	156	1202208679	149	1202209112	146
69	1202207291	155	1202207752	160	1202208225	159	1202208681	151	1202209115	149
70	1202207293	157	1202207754	162	1202208227	161	1202208684	154	1202209117	151
71	1202207296	160	1202207757	165	1202208230	164	1202208686	156	1202209120	154
72	1202207299	163	1202207759	167	1202208232	166	1202208689	159	1202209123	157
73	1202207301	165	1202207761	169	1202208235	169	1202208691	161	1202209125	159
74	1202207303	167	1202207763	171	1202208237	171	1202208693	163	1202209127	161
75	1202207306	170	1202207766	174	1202208240	174	1202208695	165	1202209129	163
76	1202207308	172	1202207768	176	1202208242	176	1202208698	168	1202209131	165
77	1202207310	174	1202207770	178	1202208244	178	1202208701	171	1202209134	168
78	1202207313	177	1202207773	181	1202208247	181	1202208703	173	1202209135	169
79	1202207315	179	1202207775	183	1202208249	183	1202208706	176	1202209139	173
80	1202207319	183	1202207779	187	1202208253	187	1202208709	179	1202209141	175
81	1202207321	185	1202207782	190	1202208254	188	1202208712	182	1202209145	179
82	1202207324	188	1202207785	193	1202208258	192	1202208716	186	1202209148	182
83	1202207327	191	1202207789	197	1202208261	195	1202208718	188	1202209151	185
					•				•	

84	1202207331	195	1202207791	199	1202208265	199	1202208722	192	1202209154	188
84										
85	1202207333	197	1202207795	203	1202208267	201	1202208724	194	1202209157	191
86	1202207337	201	1202207797	205	1202208272	206	1202208728	198	1202209161	195
87	1202207340	204	1202207802	210	1202208274	208	1202208731	201	1202209164	198
88	1202207344	208	1202207804	212	1202208279	213	1202208735	205	1202209169	203
89	1202207347	211	1202207809	217	1202208281	215	1202208737	207	1202209171	205
90	1202207350	214	1202207811	219	1202208285	219	1202208742	212	1202209176	210
91	1202207354	218	1202207816	224	1202208287	221	1202208744	214	1202209177	211
92	1202207357	221	1202207818	226	1202208292	226	1202208749	219	1202209182	216
93	1202207361	225	1202207823	231	1202208294	228	1202208751	221	1202209184	218
94	1202207364	228	1202207825	233	1202208299	233	1202208756	226	1202209190	224
95	1202207369	233	1202207830	238	1202208302	236	1202208758	228	1202209192	226
96	1202207371	235	1202207832	240	1202208307	241	1202208763	233	1202209197	231
97	1202207376	240	1202207838	246	1202208309	243	1202208765	235	1202209199	233
98	1202207378	242	1202207841	249	1202208314	248	1202208771	241	1202209204	238
99	1202207383	247	1202207845	253	1202208316	250	1202208772	242	1202209206	240
100	1202207385	249	1202207848	256	1202208321	255	1202208779	249	1202209211	245
								-		
101	1202207391	255	1202207852	260	1202208323	257	1202208779	249	1202209213	247
102	1202207393	257	1202207856	264	1202208329	263	1202208786	256	1202209219	253
103	1202207399	263	1202207860	268	1202208331	265	1202208789	259	1202209221	255
104	1202207401	265	1202207864	272	1202208337	271	1202208795	265	1202209227	261
105	1202207407	271	1202207868	276	1202208339	273	1202208797	267	1202209229	263
106	1202207409	273	1202207872	280	1202208346	280	1202208803	273	1202209236	270
107	1202207416	280	1202207877	285	1202208348	282	1202208805	275	1202209237	271
108	1202207417	281	1202207880	288	1202208354	288	1202208812	282	1202209244	278
109	1202207424	288	1202207886	294	1202208356	290	1202208813	283	1202209246	280
110	1202207425	289	1202207888	296	1202208362	296	1202208820	290	1202209252	286
111	1202207432	296	1202207893	301	1202208363	297	1202208822	292	1202209253	287
112	1202207433	297	1202207895	303	1202208370	304	1202208827	297	1202209260	294
113	1202207439	303	1202207900	308	1202208371	305	1202208829	299	1202209261	295
114	1202207441	305	1202207902	310	1202208378	312	1202208834	304	1202209267	301
115	1202207446	310	1202207907	315	1202208379	313	1202208836	306	1202209269	303
116	1202207448	312	1202207909	317	1202208385	319	1202208842	312	1202209274	308
117	1202207453	317	1202207914	322	1202208387	321	1202208843	313	1202209276	310
	1202207455	319		324		326		320	1202209270	315
118			1202207916		1202208392		1202208850			
119	1202207460	324	1202207921	329	1202208394	328	1202208851	321	1202209283	317
120	1202207462	326	1202207923	331	1202208399	333	1202208857	327	1202209288	322
121	1202207467	331	1202207928	336	1202208401	335	1202208860	330	1202209290	324
122	1202207469	333	1202207929	337	1202208406	340	1202208864	334	1202209295	329
123	1202207473	337	1202207934	342	1202208407	341	1202208866	336	1202209296	330
124	1202207475	339	1202207936	344	1202208413	347	1202208871	341	1202209302	336
125	1202207480	344	1202207940	348	1202208415	349	1202208873	343	1202209304	338
126	1202207481	345	1202207942	350	1202208419	353	1202208878	348	1202209309	343
127	1202207487	351	1202207947	355	1202208421	355	1202208879	349	1202209310	344
128	1202207488	352	1202207948	356	1202208425	359	1202208884	354	1202209315	349
129	1202207493	357	1202207952	360	1202208426	360	1202208885	355	1202209316	350
130	1202207493	357	1202207954	362	1202208430	364	1202208889	359	1202209320	354
131	1202207498	362	1202207956	364	1202208431	365	1202208891	361	1202209321	355
132	1202207498	362	1202207958	366	1202208433	367	1202208892	362	1202209324	358
133	1202207502	366	1202207959	367	1202208436	370	1202208895	365	1202209326	360
134	1202207503	367	1202207962	370	1202208437	371	1202208896	366	1202209328	362
	_	370				374	1202208890	369		363
135	1202207506		1202207965	373	1202208440				1202209329	
136	1202207507	371	1202207966	374	1202208441	375	1202208900	370	1202209332	366
137	1202207510	374	1202207969	377	1202208444	378	1202208903	373	1202209335	369
138	1202207511	375	1202207970	378	1202208445	379	1202208904	374	1202209338	372
139	1202207514	378	1202207973	381	1202208448	382	1202208907	377	1202209339	373
140	1202207515	379	1202207974	382	1202208449	383	1202208910	380	1202209342	376
141	1202207518	382	1202207977	385	1202208452	386	1202208911	381	1202209343	377
	1202207519	383	1202207979	387	1202208453	387	1202208914	384	1202209346	380

143	1202207522	386	1202207980	388	1202208456	390	1202208915	385	1202209348	382
144	1202207523	387	1202207983	391	1202208457	391	1202208918	388	1202209349	383
145	1202207526	390	1202207984	392	1202208460	394	1202208919	389	1202209352	386
146	1202207527	391	1202207987	395	1202208462	396	1202208922	392	1202209354	388
147	1202207530	394	1202207990	398	1202208463	397	1202208924	394	1202209355	389
148	1202207532	396	1202207991	399	1202208466	400	1202208925	395	1202209358	392
149	1202207533	397	1202207994	402	1202208467	401	1202208928	398	1202209359	393
150	1202207536	400	1202207995	403	1202208470	404	1202208929	399	1202209362	396
151	1202207538	402	1202207998	406	1202208472	406	1202208932	402	1202209364	398
152	1202207539	403	1202207999	407	1202208473	407	1202208933	403	1202209365	399
153	1202207542	406	1202208002	410	1202208476	410	1202208936	406	1202209368	402
154	1202207543	407	1202208003	411	1202208478	412	1202208937	407	1202209369	403
155	1202207546	410	1202208005	413	1202208479	413	1202208940	410	1202209372	406
156	1202207547	411	1202208008	416	1202208482	416	1202208941	411	1202209373	407
157	1202207550	414	1202208009	417	1202208483	417	1202208944	414	1202209376	410
158	1202207551	415	1202208012	420	1202208486	420	1202208945	415	1202209378	412
159	1202207554	418	1202208013	421	1202208487	421	1202208948	418	1202209379	413
160	1202207555	419	1202208016	424	1202208490	424	1202208949	419	1202209382	416

Condor	Mandelbro	t 4 Machin	nes							
Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A
1	1202194603	0	1202194955	0	1202195336	0	1202195696	0	1202196243	0
2	1202194607	4	1202194959	4	1202195338	2	1202195698	2	1202196247	4
3	1202194611	8	1202194964	9	1202195340	4	1202195700	4	1202196251	8
4	1202194615	12	1202194968	13	1202195342	6	1202195702	6	1202196255	12
5	1202194619	16	1202194972	17	1202195344	8	1202195704	8	1202196259	16
6	1202194624	21	1202194976	21	1202195346	10	1202195707	11	1202196263	20
7	1202194626	23	1202194978	23	1202195348	12	1202195708	12	1202196265	22
8	1202194627	24	1202194980	25	1202195350	14	1202195710	14	1202196267	24
9	1202194629	26	1202194982	27	1202195352	16	1202195712	16	1202196269	26
10	1202194631	28	1202194984	29	1202195354	18	1202195714	18	1202196271	28
11	1202194633	30	1202194986	31	1202195356	20	1202195717	21	1202196273	30
12	1202194636	33	1202194988	33	1202195358	22	1202195719	23	1202196275	32
13	1202194637	34	1202194990	35	1202195360	24	1202195720	24	1202196277	34
14	1202194639	36	1202194992	37	1202195361	25	1202195722	26	1202196279	36
15	1202194641	38	1202194993	38	1202195364	28	1202195724	28	1202196281	38
16	1202194642	39	1202194996	41	1202195365	29	1202195726	30	1202196282	39
17	1202194646	43	1202194997	42	1202195368	32	1202195728	32	1202196285	42
18	1202194646	43	1202194999	44	1202195369	33	1202195729	33	1202196287	44
19	1202194649	46	1202195002	47	1202195372	36	1202195732	36	1202196289	46
20	1202194650	47	1202195004	49	1202195373	37	1202195734	38	1202196291	48
21	1202194653	50	1202195005	50	1202195376	40	1202195735	39	1202196292	49
22	1202194654	51	1202195008	53	1202195377	41	1202195738	42	1202196295	52
23	1202194657	54	1202195009	54	1202195380	44	1202195741	45	1202196296	53
24	1202194658	55	1202195011	56	1202195381	45	1202195743	47	1202196299	56
25	1202194661	58	1202195014	59	1202195384	48	1202195744	48	1202196300	57
26	1202194663	60	1202195015	60	1202195385	49	1202195747	51	1202196302	59
27	1202194664	61	1202195018	63	1202195388	52	1202195748	52	1202196305	62
28	1202194668	65	1202195020	65	1202195390	54	1202195751	55	1202196307	64
29	1202194669	66	1202195021	66	1202195392	56	1202195752	56	1202196308	65
30	1202194670	67	1202195024	69	1202195393	57	1202195754	58	1202196311	68
31	1202194673	70	1202195027	72	1202195396	60	1202195757	61	1202196312	69
32	1202194674	71	1202195029	74	1202195399	63	1202195758	62	1202196315	72
33	1202194677	74	1202195030	75	1202195400	64	1202195761	65	1202196318	75
34	1202194678	75	1202195032	77	1202195403	67	1202195762	66	1202196319	76
35	1202194681	78	1202195035	80	1202195404	68	1202195765	69	1202196322	79
36	1202194683	80	1202195037	82	1202195407	71	1202195766	70	1202196324	81
37	1202194684	81	1202195038	83	1202195409	73	1202195769	73	1202196325	82

1	l	l ₋ .	l	1	l	l	l	l	l	l
38	1202194687	84	1202195040	85	1202195410	74	1202195770	74	1202196328	85
39	1202194689	86	1202195043	88	1202195413	77	1202195773	77	1202196329	86
40	1202194690	87	1202195045	90	1202195414	78	1202195774	78	1202196332	89
41	1202194692	89	1202195046	91	1202195417	81	1202195777	81	1202196334	91
42	1202194695	92	1202195049	94	1202195418	82	1202195778	82	1202196335	92
43	1202194696	93	1202195050	95	1202195421	85	1202195781	85	1202196339	96
44	1202194700	97	1202195052	97	1202195423	87	1202195782	86	1202196339	96
45	1202194701	98	1202195055	100	1202195425	89	1202195785	89	1202196342	99
46	1202194702	99	1202195057	102	1202195426	90	1202195787	91	1202196344	101
47	1202194705	102	1202195059	104	1202195428	92	1202195789	93	1202196345	102
48	1202194708	105	1202195061	106	1202195431	95	1202195791	95	1202196349	106
49	1202194710	107	1202195063	108	1202195433	97	1202195793	97	1202196351	108
50	1202194712	109	1202195065	110	1202195435	99	1202195795	99	1202196353	110
51	1202194714	111	1202195067	112	1202195437	101	1202195797	101	1202196355	112
52	1202194716	113	1202195069	114	1202195439	103	1202195799	103	1202196357	114
53	1202194718	115	1202195071	116	1202195441	105	1202195801	105	1202196359	116
54	1202194720	117	1202195074	119	1202195444	108	1202195803	107	1202196361	118
55	1202194723	120	1202195075	120	1202195446	110	1202195806	110	1202196364	121
56	1202194725	122	1202195078	123	1202195448	112	1202195808	112	1202196366	123
57	1202194728	125	1202195080	125	1202195450	114	1202195810	114	1202196368	125
58	1202194730	127	1202195083	128	1202195452	116	1202195813	117	1202196371	128
59	1202194732	129	1202195084	129	1202195455	119	1202195815	119	1202196372	129
60	1202194733	130	1202195086	131	1202195455	119	1202195816	120	1202196374	131
61	1202194736	133	1202195088	133	1202195457	121	1202195817	121	1202196376	133
62	1202194738	135	1202195090	135	1202195460	124	1202195820	124	1202196379	136
63	1202194740	137	1202195093	138	1202195462	126	1202195822	126	1202196380	137
64	1202194743	140	1202195094	139	1202195465	129	1202195824	128	1202196382	139
65	1202194744	141	1202195096	141	1202195466	130	1202195826	130	1202196384	141
66	1202194747	144	1202195098	143	1202195468	132	1202195828	132	1202196386	143
67	1202194750	147	1202195100	145	1202195470	134	1202195830	134	1202196389	146
68	1202194751	148	1202195103	148	1202195472	136	1202195832	136	1202196390	147
69	1202194752	149	1202195103	148	1202195474	138	1202195834	138	1202196392	149
70	1202194755	152	1202195106	151	1202195476	140	1202195837	141	1202196394	151
71	1202194757	154	1202195108	153	1202195478	142	1202195838	142	1202196396	153
72	1202194759	156	1202195110	155	1202195480	144	1202195840	144	1202196398	155
73	1202194761	158	1202195112	157	1202195482	146	1202195842	146	1202196400	157
74	1202194763	160	1202195114	159	1202195483	147	1202195844	148	1202196402	159
75	1202194764	161	1202195115	160	1202195486	150	1202195845	149	1202196403	160
76	1202194766	163	1202195117	162	1202195488	152	1202195848	152	1202196405	162
77	1202194768	165	1202195119	164	1202195491	155	1202195850	154	1202196408	165
78	1202194771	168	1202195122	167	1202195493	157	1202195852	156	1202196410	167
79	1202194773	170	1202195124	169	1202195496	160	1202195854	158	1202196412	169
80	1202194776	173	1202195127	172	1202195498	162	1202195857	161	1202196415	172
81	1202194777	174	1202195129	174	1202195500	164	1202195859	163	1202196417	174
82	1202194780	177	1202195132	177	1202195502	166	1202195861	165	1202196420	177
83	1202194782	179	1202195135	180	1202195504	168	1202195863	167	1202196422	179
84	1202194784	181	1202195137	182	1202195506	170	1202195865	169	1202196424	181
85	1202194786	183	1202195139	184	1202195508	172	1202195867	171	1202196427	184
86	1202194789	186	1202195141	186	1202195510	174	1202195870	174	1202196428	185
87	1202194791	188	1202195144	189	1202195513	177	1202195872	176	1202196431	188
88	1202194794	191	1202195146	191	1202195515	179	1202195875	179	1202196433	190
89	1202194795	192	1202195150	195	1202195517	181	1202195875	179	1202196435	192
90	1202194797	194	1202195150	195	1202195520	184	1202195878	182	1202196438	195
91	1202194799	196	1202195153	198	1202195522	186	1202195880	184	1202196440	197
92	1202194801	198	1202195155	200	1202195524	188	1202195882	186	1202196442	199
93	1202194804	201	1202195157	202	1202195526	190	1202195884	188	1202196444	201
94	1202194806	203	1202195160	205	1202195529	193	1202195887	191	1202196447	204
95	1202194808	205	1202195162	207	1202195531	195	1202195888	192	1202196448	205
96	1202194810	207	1202195163	208	1202195534	198	1202195890	194	1202196450	207
<u> </u>	1	I.	1	<u>I</u>	1	<u>i</u>	1	i	<u> </u>	I

97 1202194812 209 98 1202194815 212 99 1202194817 214 100 1202194819 216 101 1202194821 218 102 1202194824 221 103 1202194827 224 104 1202194828 225 105 1202194830 227 106 1202194832 229 107 1202194836 233 108 1202194838 235 109 1202194840 237 110 1202194841 238 111 1202194844 241 112 1202194844 241 113 1202194844 244 114 1202194847 244 115 1202194852 249 116 1202194853 250 117 1202194853 250 117 1202194861 258 120 1202194863 260	1202195165 1202195167 1202195170 1202195174 1202195174 1202195177 1202195180 1202195186 1202195186 1202195189 1202195191 1202195195 1202195196 1202195196 1202195200 1202195200 1202195201 1202195206 1202195208 1202195208 1202195210 1202195210 1202195216 1202195216 1202195218	210 212 215 217 219 222 225 227 229 231 234 236 237 240 241 244 245 248 249 251 253 255 258 259 261	1202195535 1202195537 1202195539 1202195541 1202195543 1202195545 1202195549 1202195551 1202195555 1202195555 1202195556 1202195560 1202195566 1202195568 1202195568 1202195569 1202195571 1202195571 1202195577 1202195579 1202195579	199 201 203 205 207 209 213 215 217 219 222 224 225 227 230 232 233 235 237 240 241 243	1202195892 1202195896 1202195896 1202195898 1202195900 1202195906 1202195908 1202195912 1202195915 1202195915 1202195915 1202195918 1202195921 1202195922 1202195925 1202195925 1202195926 1202195930 1202195932	196 199 200 202 204 207 210 212 214 216 219 221 222 225 226 229 230 233 234 236	1202196452 1202196455 1202196457 1202196460 1202196463 1202196463 1202196467 1202196467 1202196471 1202196475 1202196475 1202196479 1202196482 1202196483 1202196487 1202196487 1202196490 1202196490	209 212 214 215 217 220 222 224 226 228 231 232 234 236 239 240 243 244 247
99 1202194817 214 100 1202194819 216 101 1202194821 218 102 1202194824 221 103 1202194827 224 104 1202194828 225 105 1202194830 227 106 1202194832 229 107 1202194836 233 108 1202194836 235 109 1202194840 237 110 1202194841 238 111 1202194844 241 112 1202194845 242 113 1202194845 242 113 1202194849 246 115 1202194852 249 116 1202194853 250 117 1202194854 253 118 1202194858 255 119 1202194863 260 121 1202194863 260 121 1202194867 264 <td>1202195170 1202195170 1202195174 1202195177 1202195180 1202195184 1202195186 1202195189 1202195191 1202195195 1202195196 1202195196 1202195200 1202195200 1202195200 1202195201 1202195201 1202195201 1202195201 1202195201 1202195201 1202195201 1202195201 1202195201 1202195201 1202195201 1202195201 1202195201</td> <td>215 217 219 222 225 227 229 231 234 236 237 240 241 244 245 248 249 251 253 255 258 259</td> <td>1202195539 1202195541 1202195543 1202195545 1202195551 1202195553 1202195555 1202195556 1202195560 1202195566 1202195568 1202195568 1202195568 1202195569 1202195571 1202195573 1202195576 1202195577</td> <td>203 205 207 209 213 215 217 219 222 224 225 227 230 232 233 235 237 240 241</td> <td>1202195896 1202195898 1202195900 1202195906 1202195908 1202195910 1202195912 1202195915 1202195915 1202195917 1202195918 1202195921 1202195922 1202195925 1202195926 1202195929 1202195930 1202195932</td> <td>200 202 204 207 210 212 214 216 219 221 222 225 226 229 230 233 234 236</td> <td>1202196457 1202196458 1202196460 1202196463 1202196465 1202196467 1202196469 1202196471 1202196475 1202196479 1202196479 1202196482 1202196483 1202196486 1202196487</td> <td>214 215 217 220 222 224 226 228 231 232 234 236 239 240 243 244 247</td>	1202195170 1202195170 1202195174 1202195177 1202195180 1202195184 1202195186 1202195189 1202195191 1202195195 1202195196 1202195196 1202195200 1202195200 1202195200 1202195201 1202195201 1202195201 1202195201 1202195201 1202195201 1202195201 1202195201 1202195201 1202195201 1202195201 1202195201 1202195201	215 217 219 222 225 227 229 231 234 236 237 240 241 244 245 248 249 251 253 255 258 259	1202195539 1202195541 1202195543 1202195545 1202195551 1202195553 1202195555 1202195556 1202195560 1202195566 1202195568 1202195568 1202195568 1202195569 1202195571 1202195573 1202195576 1202195577	203 205 207 209 213 215 217 219 222 224 225 227 230 232 233 235 237 240 241	1202195896 1202195898 1202195900 1202195906 1202195908 1202195910 1202195912 1202195915 1202195915 1202195917 1202195918 1202195921 1202195922 1202195925 1202195926 1202195929 1202195930 1202195932	200 202 204 207 210 212 214 216 219 221 222 225 226 229 230 233 234 236	1202196457 1202196458 1202196460 1202196463 1202196465 1202196467 1202196469 1202196471 1202196475 1202196479 1202196479 1202196482 1202196483 1202196486 1202196487	214 215 217 220 222 224 226 228 231 232 234 236 239 240 243 244 247
100 1202194819 216 101 1202194821 218 102 1202194824 221 103 1202194827 224 104 1202194828 225 105 1202194830 227 106 1202194832 229 107 1202194836 233 108 1202194838 235 109 1202194840 237 110 1202194841 238 111 1202194843 242 113 1202194845 242 113 1202194847 244 114 1202194852 249 116 1202194853 250 117 1202194853 250 117 1202194858 255 119 1202194863 260 121 1202194863 260 121 1202194865 262 122 1202194867 264 123 1202194871 268 <td>1202195172 1202195174 1202195177 1202195180 1202195184 1202195186 1202195189 1202195191 1202195195 1202195196 1202195196 1202195200 1202195203 1202195204 1202195208 1202195208 1202195210 1202195211 1202195214 1202195216</td> <td>217 219 222 225 227 229 231 234 236 237 240 241 244 245 248 249 251 253 255 258 259</td> <td>1202195541 1202195543 1202195545 1202195549 1202195551 1202195555 1202195555 1202195556 1202195560 1202195568 1202195568 1202195568 1202195569 1202195571 1202195573 1202195576 1202195577</td> <td>205 207 209 213 215 217 219 222 224 225 227 230 232 233 235 237 240</td> <td>1202195898 1202195900 1202195903 1202195906 1202195908 1202195910 1202195912 1202195915 1202195917 1202195918 1202195921 1202195922 1202195925 1202195926 1202195930 1202195932</td> <td>202 204 207 210 212 214 216 219 221 222 225 226 229 230 233 234 236</td> <td>1202196458 1202196460 1202196463 1202196465 1202196469 1202196471 1202196474 1202196475 1202196479 1202196482 1202196483 1202196486 1202196487</td> <td>215 217 220 222 224 226 228 231 232 234 236 239 240 243 244 247</td>	1202195172 1202195174 1202195177 1202195180 1202195184 1202195186 1202195189 1202195191 1202195195 1202195196 1202195196 1202195200 1202195203 1202195204 1202195208 1202195208 1202195210 1202195211 1202195214 1202195216	217 219 222 225 227 229 231 234 236 237 240 241 244 245 248 249 251 253 255 258 259	1202195541 1202195543 1202195545 1202195549 1202195551 1202195555 1202195555 1202195556 1202195560 1202195568 1202195568 1202195568 1202195569 1202195571 1202195573 1202195576 1202195577	205 207 209 213 215 217 219 222 224 225 227 230 232 233 235 237 240	1202195898 1202195900 1202195903 1202195906 1202195908 1202195910 1202195912 1202195915 1202195917 1202195918 1202195921 1202195922 1202195925 1202195926 1202195930 1202195932	202 204 207 210 212 214 216 219 221 222 225 226 229 230 233 234 236	1202196458 1202196460 1202196463 1202196465 1202196469 1202196471 1202196474 1202196475 1202196479 1202196482 1202196483 1202196486 1202196487	215 217 220 222 224 226 228 231 232 234 236 239 240 243 244 247
101 1202194821 218 102 1202194824 221 103 1202194827 224 104 1202194828 225 105 1202194830 227 106 1202194832 229 107 1202194836 233 108 1202194838 235 109 1202194840 237 110 1202194841 238 111 1202194844 241 112 1202194845 242 113 1202194847 244 114 1202194849 246 115 1202194852 249 116 1202194853 250 117 1202194853 250 118 1202194858 255 119 1202194863 260 121 1202194863 260 121 1202194865 262 122 1202194867 264 123 1202194871 268 <td>1202195174 1202195177 1202195180 1202195184 1202195186 1202195186 1202195189 1202195191 1202195192 1202195196 1202195190 1202195200 1202195203 1202195204 1202195208 1202195208 1202195210 1202195211 1202195216 1202195216</td> <td>219 222 225 227 229 231 234 236 237 240 241 244 245 248 249 251 253 255 258 259</td> <td>1202195543 1202195545 1202195549 1202195551 1202195553 1202195558 1202195558 1202195560 1202195561 1202195566 1202195568 1202195569 1202195571 1202195573 1202195576 1202195577</td> <td>207 209 213 215 217 219 222 224 225 227 230 232 233 235 237 240</td> <td>1202195900 1202195903 1202195906 1202195908 1202195910 1202195912 1202195915 1202195917 1202195918 1202195921 1202195922 1202195925 1202195926 1202195929 1202195930 1202195932</td> <td>204 207 210 212 214 216 219 221 222 225 226 229 230 233 234 236</td> <td>1202196460 1202196463 1202196465 1202196467 1202196471 1202196471 1202196475 1202196475 1202196479 1202196482 1202196483 1202196486 1202196487</td> <td>217 220 222 224 226 228 231 232 234 236 239 240 243 244 247</td>	1202195174 1202195177 1202195180 1202195184 1202195186 1202195186 1202195189 1202195191 1202195192 1202195196 1202195190 1202195200 1202195203 1202195204 1202195208 1202195208 1202195210 1202195211 1202195216 1202195216	219 222 225 227 229 231 234 236 237 240 241 244 245 248 249 251 253 255 258 259	1202195543 1202195545 1202195549 1202195551 1202195553 1202195558 1202195558 1202195560 1202195561 1202195566 1202195568 1202195569 1202195571 1202195573 1202195576 1202195577	207 209 213 215 217 219 222 224 225 227 230 232 233 235 237 240	1202195900 1202195903 1202195906 1202195908 1202195910 1202195912 1202195915 1202195917 1202195918 1202195921 1202195922 1202195925 1202195926 1202195929 1202195930 1202195932	204 207 210 212 214 216 219 221 222 225 226 229 230 233 234 236	1202196460 1202196463 1202196465 1202196467 1202196471 1202196471 1202196475 1202196475 1202196479 1202196482 1202196483 1202196486 1202196487	217 220 222 224 226 228 231 232 234 236 239 240 243 244 247
102 1202194824 221 103 1202194827 224 104 1202194828 225 105 1202194830 227 106 1202194836 233 108 1202194836 233 109 1202194840 237 110 1202194841 238 111 1202194844 241 112 1202194845 242 113 1202194847 244 114 1202194849 246 115 1202194852 249 116 1202194853 250 117 1202194856 253 118 1202194858 255 119 1202194861 258 120 1202194863 260 121 1202194865 262 122 1202194867 264 123 1202194871 268 125 1202194873 270 126 1202194877 274 <td>1202195177 1202195180 1202195180 1202195184 1202195186 1202195189 1202195191 1202195195 1202195196 1202195200 1202195203 1202195204 1202195208 1202195208 1202195210 1202195211 1202195216 1202195216</td> <td>222 225 227 229 231 234 236 237 240 241 244 245 248 249 251 253 255 258 259</td> <td>1202195545 1202195549 1202195551 1202195553 1202195558 1202195558 1202195560 1202195563 1202195566 1202195568 1202195569 1202195571 1202195573 1202195577 1202195579</td> <td>209 213 215 217 219 222 224 225 227 230 232 233 235 237 240 241</td> <td>1202195903 1202195906 1202195908 1202195910 1202195915 1202195915 1202195917 1202195918 1202195921 1202195922 1202195925 1202195926 1202195929 1202195930 1202195932</td> <td>207 210 212 214 216 219 221 222 225 226 229 230 233 234 236</td> <td>1202196463 1202196465 1202196467 1202196469 1202196471 1202196474 1202196477 1202196477 1202196482 1202196483 1202196486 1202196487 1202196490</td> <td>220 222 224 226 228 231 232 234 236 239 240 243 244 247</td>	1202195177 1202195180 1202195180 1202195184 1202195186 1202195189 1202195191 1202195195 1202195196 1202195200 1202195203 1202195204 1202195208 1202195208 1202195210 1202195211 1202195216 1202195216	222 225 227 229 231 234 236 237 240 241 244 245 248 249 251 253 255 258 259	1202195545 1202195549 1202195551 1202195553 1202195558 1202195558 1202195560 1202195563 1202195566 1202195568 1202195569 1202195571 1202195573 1202195577 1202195579	209 213 215 217 219 222 224 225 227 230 232 233 235 237 240 241	1202195903 1202195906 1202195908 1202195910 1202195915 1202195915 1202195917 1202195918 1202195921 1202195922 1202195925 1202195926 1202195929 1202195930 1202195932	207 210 212 214 216 219 221 222 225 226 229 230 233 234 236	1202196463 1202196465 1202196467 1202196469 1202196471 1202196474 1202196477 1202196477 1202196482 1202196483 1202196486 1202196487 1202196490	220 222 224 226 228 231 232 234 236 239 240 243 244 247
103 1202194827 224 104 1202194828 225 105 1202194830 227 106 1202194832 229 107 1202194836 233 108 1202194838 235 109 1202194840 237 110 1202194841 238 111 1202194844 241 112 1202194845 242 113 1202194847 244 114 1202194849 246 115 1202194852 249 116 1202194853 250 117 1202194856 253 118 1202194856 253 119 1202194861 258 120 1202194863 260 121 1202194863 260 121 1202194867 264 123 1202194871 268 125 1202194873 270 126 1202194878 275 <td>1202195180 1202195184 1202195186 1202195186 1202195191 1202195192 1202195195 1202195196 1202195200 1202195200 1202195200 1202195206 1202195201 1202195201 1202195208 1202195208 1202195210 1202195211 1202195214 1202195216</td> <td>225 227 229 231 234 236 237 240 241 244 245 248 249 251 253 255 258 259</td> <td>1202195549 1202195551 1202195553 1202195555 1202195556 1202195560 1202195563 1202195566 1202195569 1202195569 1202195571 1202195573 1202195576 1202195577</td> <td>213 215 217 219 222 224 225 227 230 232 233 235 237 240 241</td> <td>1202195906 1202195908 1202195910 1202195912 1202195915 1202195917 1202195918 1202195921 1202195922 1202195925 1202195926 1202195929 1202195930 1202195932</td> <td>210 212 214 216 219 221 222 225 226 229 230 233 234 236</td> <td>1202196465 1202196467 1202196469 1202196471 1202196474 1202196475 1202196479 1202196482 1202196483 1202196486 1202196487</td> <td>222 224 226 228 231 232 234 236 239 240 243 244 247</td>	1202195180 1202195184 1202195186 1202195186 1202195191 1202195192 1202195195 1202195196 1202195200 1202195200 1202195200 1202195206 1202195201 1202195201 1202195208 1202195208 1202195210 1202195211 1202195214 1202195216	225 227 229 231 234 236 237 240 241 244 245 248 249 251 253 255 258 259	1202195549 1202195551 1202195553 1202195555 1202195556 1202195560 1202195563 1202195566 1202195569 1202195569 1202195571 1202195573 1202195576 1202195577	213 215 217 219 222 224 225 227 230 232 233 235 237 240 241	1202195906 1202195908 1202195910 1202195912 1202195915 1202195917 1202195918 1202195921 1202195922 1202195925 1202195926 1202195929 1202195930 1202195932	210 212 214 216 219 221 222 225 226 229 230 233 234 236	1202196465 1202196467 1202196469 1202196471 1202196474 1202196475 1202196479 1202196482 1202196483 1202196486 1202196487	222 224 226 228 231 232 234 236 239 240 243 244 247
104 1202194828 225 105 1202194830 227 106 1202194832 229 107 1202194836 233 108 1202194838 235 109 1202194840 237 110 1202194841 238 111 1202194844 241 112 1202194845 242 113 1202194847 244 114 1202194852 249 116 1202194853 250 117 1202194856 253 118 1202194858 255 119 1202194861 258 120 1202194863 260 121 1202194863 260 121 1202194869 266 122 1202194871 268 125 1202194873 270 126 1202194878 275 127 1202194878 275 129 1202194880 277 <td>1202195182 1202195184 1202195186 1202195189 1202195191 1202195195 1202195196 1202195199 1202195200 1202195204 1202195206 1202195208 1202195210 1202195210 1202195211 1202195214 1202195216</td> <td>227 229 231 234 236 237 240 241 244 245 248 249 251 253 255 258 259</td> <td>1202195551 1202195553 1202195555 1202195556 1202195560 1202195561 1202195563 1202195566 1202195569 1202195571 1202195573 1202195576 1202195577 1202195579</td> <td>215 217 219 222 224 225 227 230 232 233 235 237 240 241</td> <td>1202195908 1202195910 1202195912 1202195915 1202195917 1202195918 1202195921 1202195922 1202195925 1202195926 1202195929 1202195930 1202195932</td> <td>212 214 216 219 221 222 225 226 229 230 233 234 236</td> <td>1202196467 1202196469 1202196471 1202196474 1202196475 1202196479 1202196482 1202196483 1202196486 1202196487</td> <td>224 226 228 231 232 234 236 239 240 243 244 247</td>	1202195182 1202195184 1202195186 1202195189 1202195191 1202195195 1202195196 1202195199 1202195200 1202195204 1202195206 1202195208 1202195210 1202195210 1202195211 1202195214 1202195216	227 229 231 234 236 237 240 241 244 245 248 249 251 253 255 258 259	1202195551 1202195553 1202195555 1202195556 1202195560 1202195561 1202195563 1202195566 1202195569 1202195571 1202195573 1202195576 1202195577 1202195579	215 217 219 222 224 225 227 230 232 233 235 237 240 241	1202195908 1202195910 1202195912 1202195915 1202195917 1202195918 1202195921 1202195922 1202195925 1202195926 1202195929 1202195930 1202195932	212 214 216 219 221 222 225 226 229 230 233 234 236	1202196467 1202196469 1202196471 1202196474 1202196475 1202196479 1202196482 1202196483 1202196486 1202196487	224 226 228 231 232 234 236 239 240 243 244 247
105 1202194830 227 106 1202194832 229 107 1202194836 233 108 1202194838 235 109 1202194840 237 110 1202194841 238 111 1202194844 241 112 1202194845 242 113 1202194847 244 114 1202194849 246 115 1202194852 249 116 1202194853 250 117 1202194856 253 118 1202194858 255 119 1202194861 258 120 1202194863 260 121 1202194863 260 121 1202194865 262 122 1202194867 264 123 1202194871 268 125 1202194873 270 126 1202194875 272 127 1202194878 275 <td>1202195184 1202195186 1202195189 1202195191 1202195195 1202195196 1202195199 1202195200 1202195203 1202195204 1202195208 1202195208 1202195210 1202195211 1202195214 1202195216 1202195218</td> <td>229 231 234 236 237 240 241 244 245 248 249 251 253 255 258 259</td> <td>1202195553 1202195555 1202195558 1202195560 1202195561 1202195566 1202195568 1202195569 1202195571 1202195573 1202195576 1202195577</td> <td>217 219 222 224 225 227 230 232 233 235 237 240</td> <td>1202195910 1202195912 1202195915 1202195917 1202195918 1202195921 1202195922 1202195925 1202195926 1202195929 1202195930 1202195932</td> <td>214 216 219 221 222 225 226 229 230 233 234 236</td> <td>1202196469 1202196471 1202196474 1202196475 1202196479 1202196482 1202196483 1202196486 1202196487</td> <td>226 228 231 232 234 236 239 240 243 244 247</td>	1202195184 1202195186 1202195189 1202195191 1202195195 1202195196 1202195199 1202195200 1202195203 1202195204 1202195208 1202195208 1202195210 1202195211 1202195214 1202195216 1202195218	229 231 234 236 237 240 241 244 245 248 249 251 253 255 258 259	1202195553 1202195555 1202195558 1202195560 1202195561 1202195566 1202195568 1202195569 1202195571 1202195573 1202195576 1202195577	217 219 222 224 225 227 230 232 233 235 237 240	1202195910 1202195912 1202195915 1202195917 1202195918 1202195921 1202195922 1202195925 1202195926 1202195929 1202195930 1202195932	214 216 219 221 222 225 226 229 230 233 234 236	1202196469 1202196471 1202196474 1202196475 1202196479 1202196482 1202196483 1202196486 1202196487	226 228 231 232 234 236 239 240 243 244 247
106 1202194832 229 107 1202194836 233 108 1202194838 235 109 1202194840 237 110 1202194841 238 111 1202194844 241 112 1202194845 242 113 1202194847 244 114 1202194849 246 115 1202194852 249 116 1202194853 250 117 1202194853 250 118 1202194858 255 119 1202194863 260 121 1202194863 260 121 1202194863 260 121 1202194865 262 122 1202194867 264 123 1202194871 268 125 1202194873 270 126 1202194875 272 127 1202194878 275 129 1202194880 277 <td>1202195186 1202195189 1202195191 1202195195 1202195196 1202195196 1202195200 1202195203 1202195204 1202195206 1202195208 1202195210 1202195211 1202195214 1202195216 1202195216</td> <td>231 234 236 237 240 241 244 245 248 249 251 253 255 258 259</td> <td>1202195555 1202195558 1202195560 1202195561 1202195563 1202195568 1202195568 1202195569 1202195571 1202195573 1202195576 1202195577</td> <td>219 222 224 225 227 230 232 233 235 237 240 241</td> <td>1202195912 1202195915 1202195917 1202195918 1202195921 1202195922 1202195925 1202195926 1202195929 1202195930 1202195932</td> <td>216 219 221 222 225 226 229 230 233 234 236</td> <td>1202196471 1202196474 1202196475 1202196477 1202196479 1202196482 1202196483 1202196486 1202196487</td> <td>228 231 232 234 236 239 240 243 244 247</td>	1202195186 1202195189 1202195191 1202195195 1202195196 1202195196 1202195200 1202195203 1202195204 1202195206 1202195208 1202195210 1202195211 1202195214 1202195216 1202195216	231 234 236 237 240 241 244 245 248 249 251 253 255 258 259	1202195555 1202195558 1202195560 1202195561 1202195563 1202195568 1202195568 1202195569 1202195571 1202195573 1202195576 1202195577	219 222 224 225 227 230 232 233 235 237 240 241	1202195912 1202195915 1202195917 1202195918 1202195921 1202195922 1202195925 1202195926 1202195929 1202195930 1202195932	216 219 221 222 225 226 229 230 233 234 236	1202196471 1202196474 1202196475 1202196477 1202196479 1202196482 1202196483 1202196486 1202196487	228 231 232 234 236 239 240 243 244 247
107 1202194836 233 108 1202194838 235 109 1202194840 237 110 1202194841 238 111 1202194844 241 112 1202194845 242 113 1202194847 244 114 1202194849 246 115 1202194852 249 116 1202194853 250 117 1202194856 253 118 1202194858 255 119 1202194861 258 120 1202194863 260 121 1202194863 260 121 1202194865 262 122 1202194869 266 124 1202194871 268 125 1202194873 270 126 1202194875 272 127 1202194878 275 129 1202194880 277	1202195189 1202195191 1202195192 1202195195 1202195199 1202195200 1202195203 1202195204 1202195208 1202195208 1202195210 1202195211 1202195214 1202195216 1202195216	234 236 237 240 241 244 245 248 249 251 253 255 258 259	1202195558 1202195560 1202195561 1202195563 1202195566 1202195569 1202195571 1202195573 1202195576 1202195577 1202195577	222 224 225 227 230 232 233 235 237 240	1202195915 1202195917 1202195918 1202195921 1202195922 1202195925 1202195926 1202195929 1202195930 1202195932	219 221 222 225 226 229 230 233 234 236	1202196474 1202196475 1202196477 1202196479 1202196482 1202196483 1202196486 1202196487 1202196490	231 232 234 236 239 240 243 244 247
108 1202194838 235 109 1202194840 237 110 1202194841 238 111 1202194844 241 112 1202194845 242 113 1202194847 244 114 1202194849 246 115 1202194852 249 116 1202194853 250 117 1202194856 253 118 1202194856 253 119 1202194861 258 120 1202194863 260 121 1202194863 260 121 1202194865 262 122 1202194867 264 123 1202194871 268 124 1202194873 270 126 1202194875 272 127 1202194878 275 129 1202194880 277	1202195191 1202195192 1202195195 1202195196 1202195200 1202195203 1202195204 1202195208 1202195208 1202195210 1202195211 1202195214 1202195216 1202195218	236 237 240 241 244 245 248 249 251 253 255 258 259	1202195560 1202195561 1202195563 1202195566 1202195569 1202195571 1202195573 1202195576 1202195577 1202195579	224 225 227 230 232 233 235 237 240	1202195917 1202195918 1202195921 1202195922 1202195925 1202195926 1202195929 1202195930 1202195932	221 222 225 226 229 230 233 234 236	1202196475 1202196477 1202196479 1202196482 1202196483 1202196486 1202196490	232 234 236 239 240 243 244 247
109 1202194840 237 110 1202194841 238 111 1202194844 241 112 1202194845 242 113 1202194847 244 114 1202194849 246 115 1202194852 249 116 1202194853 250 117 1202194856 253 118 1202194858 255 119 1202194861 258 120 1202194863 260 121 1202194865 262 122 1202194867 264 123 1202194867 264 123 1202194871 268 125 1202194873 270 126 1202194875 272 127 1202194878 275 129 1202194880 277	1202195192 1202195195 1202195196 1202195199 1202195200 1202195203 1202195206 1202195206 1202195208 1202195210 1202195213 1202195214 1202195216 1202195216	237 240 241 244 245 248 249 251 253 255 258 259	1202195561 1202195563 1202195566 1202195569 1202195569 1202195571 1202195573 1202195576 1202195577	225 227 230 232 233 235 237 240 241	1202195918 1202195921 1202195922 1202195925 1202195926 1202195929 1202195930 1202195932	222 225 226 229 230 233 234 236	1202196477 1202196479 1202196482 1202196483 1202196486 1202196487 1202196490	234 236 239 240 243 244 247
110 1202194841 238 111 1202194844 241 112 1202194845 242 113 1202194847 244 114 1202194849 246 115 1202194852 249 116 1202194853 250 117 1202194856 253 118 1202194858 255 119 1202194861 258 120 1202194863 260 121 1202194865 262 122 1202194867 264 123 1202194869 266 124 1202194871 268 125 1202194873 270 126 1202194875 272 127 1202194878 275 129 1202194880 277	1202195195 1202195196 1202195199 1202195200 1202195204 1202195206 1202195208 1202195210 1202195211 1202195214 1202195216 1202195218	240 241 244 245 248 249 251 253 255 258 259	1202195563 1202195566 1202195568 1202195569 1202195571 1202195573 1202195576 1202195577	227 230 232 233 235 237 240 241	1202195921 1202195922 1202195925 1202195926 1202195929 1202195930 1202195932	225 226 229 230 233 234 236	1202196479 1202196482 1202196483 1202196486 1202196487 1202196490	236 239 240 243 244 247
111 1202194844 241 112 1202194845 242 113 1202194847 244 114 1202194849 246 115 1202194852 249 116 1202194853 250 117 1202194856 253 118 1202194858 255 119 1202194861 258 120 1202194863 260 121 1202194865 262 122 1202194867 264 123 1202194869 266 124 1202194871 268 125 1202194873 270 126 1202194875 272 127 1202194877 274 128 1202194878 275 129 1202194880 277	1202195196 1202195199 1202195200 1202195203 1202195204 1202195206 1202195208 1202195210 1202195213 1202195214 1202195216 1202195216	241 244 245 248 249 251 253 255 258 259	1202195566 1202195568 1202195569 1202195571 1202195573 1202195576 1202195577	230 232 233 235 237 240 241	1202195922 1202195925 1202195926 1202195929 1202195930 1202195932	226 229 230 233 234 236	1202196482 1202196483 1202196486 1202196487 1202196490	239 240 243 244 247
112 1202194845 242 113 1202194847 244 114 1202194849 246 115 1202194852 249 116 1202194853 250 117 1202194856 253 118 1202194858 255 119 1202194861 258 120 1202194863 260 121 1202194865 262 122 1202194867 264 123 1202194869 266 124 1202194871 268 125 1202194873 270 126 1202194875 272 127 1202194877 274 128 1202194878 275 129 1202194880 277	1202195199 1202195200 1202195203 1202195204 1202195206 1202195208 1202195210 1202195213 1202195214 1202195216 1202195218	244 245 248 249 251 253 255 258 259	1202195568 1202195569 1202195571 1202195573 1202195576 1202195577 1202195579	232 233 235 237 240 241	1202195925 1202195926 1202195929 1202195930 1202195932	229 230 233 234 236	1202196483 1202196486 1202196487 1202196490	240 243 244 247
113 1202194847 244 114 1202194849 246 115 1202194852 249 116 1202194853 250 117 1202194856 253 118 1202194858 255 119 1202194861 258 120 1202194863 260 121 1202194865 262 122 1202194867 264 123 1202194869 266 124 1202194871 268 125 1202194873 270 126 1202194875 272 127 1202194877 274 128 1202194878 275 129 1202194880 277	1202195200 1202195203 1202195204 1202195206 1202195208 1202195210 1202195213 1202195214 1202195216 1202195218	245 248 249 251 253 255 258 259	1202195569 1202195571 1202195573 1202195576 1202195577 1202195579	233 235 237 240 241	1202195926 1202195929 1202195930 1202195932	230 233 234 236	1202196486 1202196487 1202196490	243 244 247
114 1202194849 246 115 1202194852 249 116 1202194853 250 117 1202194856 253 118 1202194858 255 119 1202194861 258 120 1202194863 260 121 1202194865 262 122 1202194867 264 123 1202194869 266 124 1202194871 268 125 1202194873 270 126 1202194875 272 127 1202194877 274 128 1202194878 275 129 1202194880 277	1202195203 1202195204 1202195206 1202195208 1202195210 1202195213 1202195214 1202195216 1202195218	248 249 251 253 255 258 259	1202195571 1202195573 1202195576 1202195577 1202195579	235 237 240 241	1202195929 1202195930 1202195932	233 234 236	1202196487 1202196490	244
115 1202194852 249 116 1202194853 250 117 1202194856 253 118 1202194858 255 119 1202194861 258 120 1202194863 260 121 1202194865 262 122 1202194867 264 123 1202194869 266 124 1202194871 268 125 1202194873 270 126 1202194875 272 127 1202194877 274 128 1202194878 275 129 1202194880 277	1202195204 1202195206 1202195208 1202195210 1202195213 1202195214 1202195216 1202195218	249 251 253 255 258 259	1202195573 1202195576 1202195577 1202195579	237 240 241	1202195930 1202195932	234	1202196490	247
116 1202194853 250 117 1202194856 253 118 1202194858 255 119 1202194861 258 120 1202194863 260 121 1202194865 262 122 1202194867 264 123 1202194869 266 124 1202194871 268 125 1202194873 270 126 1202194875 272 127 1202194877 274 128 1202194878 275 129 1202194880 277	1202195206 1202195208 1202195210 1202195213 1202195214 1202195216 1202195218	251 253 255 258 259	1202195576 1202195577 1202195579	240 241	1202195932	236		
117 1202194856 253 118 1202194858 255 119 1202194861 258 120 1202194863 260 121 1202194865 262 122 1202194867 264 123 1202194869 266 124 1202194871 268 125 1202194873 270 126 1202194875 272 127 1202194877 274 128 1202194878 275 129 1202194880 277	1202195208 1202195210 1202195213 1202195214 1202195216 1202195218	253 255 258 259	1202195577 1202195579	241			1202196491	240
118 1202194858 255 119 1202194861 258 120 1202194863 260 121 1202194865 262 122 1202194867 264 123 1202194869 266 124 1202194871 268 125 1202194873 270 126 1202194875 272 127 1202194877 274 128 1202194878 275 129 1202194880 277	1202195210 1202195213 1202195214 1202195216 1202195218	255 258 259	1202195579		1202195934	220		248
119 1202194861 258 120 1202194863 260 121 1202194865 262 122 1202194867 264 123 1202194869 266 124 1202194871 268 125 1202194873 270 126 1202194875 272 127 1202194877 274 128 1202194878 275 129 1202194880 277	1202195213 1202195214 1202195216 1202195218	258 259		243		238	1202196493	250
120 1202194863 260 121 1202194865 262 122 1202194867 264 123 1202194869 266 124 1202194871 268 125 1202194873 270 126 1202194875 272 127 1202194877 274 128 1202194878 275 129 1202194880 277	1202195214 1202195216 1202195218	259	1202195581	t	1202195937	241	1202196495	252
121 1202194865 262 122 1202194867 264 123 1202194869 266 124 1202194871 268 125 1202194873 270 126 1202194875 272 127 1202194877 274 128 1202194878 275 129 1202194880 277	1202195216 1202195218			245	1202195939	243	1202196497	254
122 1202194867 264 123 1202194869 266 124 1202194871 268 125 1202194873 270 126 1202194875 272 127 1202194877 274 128 1202194878 275 129 1202194880 277	1202195218	261	1202195583	247	1202195941	245	1202196499	256
123 1202194869 266 124 1202194871 268 125 1202194873 270 126 1202194875 272 127 1202194877 274 128 1202194878 275 129 1202194880 277			1202195585	249	1202195943	247	1202196501	258
124 1202194871 268 125 1202194873 270 126 1202194875 272 127 1202194877 274 128 1202194878 275 129 1202194880 277	1202195220	263	1202195587	251	1202195946	250	1202196503	260
125 1202194873 270 126 1202194875 272 127 1202194877 274 128 1202194878 275 129 1202194880 277		265	1202195589	253	1202195948	252	1202196506	263
126 1202194875 272 127 1202194877 274 128 1202194878 275 129 1202194880 277	1202195222	267	1202195591	255	1202195950	254	1202196508	265
127 1202194877 274 128 1202194878 275 129 1202194880 277	1202195224	269	1202195593	257	1202195952	256	1202196510	267
128 1202194878 275 129 1202194880 277	1202195226	271	1202195595	259	1202195954	258	1202196512	269
129 1202194880 277	1202195228	273	1202195597	261	1202195956	260	1202196514	271
	1202195229	274	1202195599	263	1202195958	262	1202196515	272
130 1202194882 279	1202195230	275	1202195601	265	1202195958	262	1202196517	274
	1202195232	277	1202195602	266	1202195960	264	1202196519	276
131 1202194884 281	1202195234	279	1202195603	267	1202195962	266	1202196521	278
132 1202194886 283	1202195235	280	1202195605	269	1202195963	267	1202196523	280
133 1202194887 284	1202195236	281	1202195607	271	1202195965	269	1202196524	281
134 1202194890 287	1202195238	283	1202195608	272	1202195966	270	1202196526	283
135 1202194892 289	1202195242	287	1202195611	275	1202195969	273	1202196529	286
136 1202194894 291	1202195244	289	1202195614	278	1202195970	274	1202196530	287
137 1202194895 292	1202195245	290	1202195616	280	1202195973	277	1202196533	290
138 1202194898 295	1202195248	293	1202195617	281	1202195974	278	1202196535	292
139 1202194901 298	1202195250	295	1202195620	284	1202195977	281	1202196536	293
140 1202194902 299	1202195252	297	1202195622	286	1202195978	282	1202196539	296
140 1202194902 299 141 1202194905 302	1202193232	299	1202193622	287	1202195981	285	1202196541	298
142 1202194906 303	1202195255	300	1202195626	290	1202195983	287	1202196544	301
143 1202194909 306	1202195258	303	1202195627	291	1202195984	288	1202196545	302
144 1202194911 308	1202195260	305	1202195630	294	1202195987	291	1202196548	305
145 1202194912 309	1202195261	306	1202195633	297	1202195988	292	1202196550	307
146 1202194915 312	1202195265	310	1202195634	298	1202195991	295	1202196551	308
147 1202194916 313	1202195265	310	1202195637	301	1202195994	298	1202196554	311
148 1202194919 316	1202195267	312	1202195639	303	1202195995	299	1202196556	313
149 1202194920 317	1202195270	315	1202195640	304	1202195998	302	1202196557	314
150 1202194923 320	1202195272	317	1202195643	307	1202195999	303	1202196560	317
151 1202194924 321	1202195273	318	1202195644	308	1202196002	306	1202196561	318
152 1202194926 323	1202195276	321	1202195647	311	1202196003	307	1202196564	321
153 1202194929 326	1202195279	324	1202195648	312	1202196006	310	1202196565	322
154 1202194931 328		326	1202195651	315	1202196007	311	1202196568	325
155 1202194932 329	1202195281	327	1202195654	318	1202196010	314	1202196570	327

156	1202194935	332	1202195285	330	1202195655	319	1202196011	315	1202196572	329
157	1202194936	333	1202195286	331	1202195658	322	1202196014	318	1202196573	330
158	1202194939	336	1202195289	334	1202195659	323	1202196017	321	1202196576	333
159	1202194940	337	1202195290	335	1202195662	326	1202196018	322	1202196578	335
160	1202194943	340	1202195293	338	1202195665	329	1202196021	325	1202196579	336

Condo	r Mandelbro	t 8 Machi	nes							
Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A
1	1202198948	0	1202199348	0	1202199738	0	1202200075	0	1202200416	0
2	1202198952	4	1202199349	1	1202199740	2	1202200077	2	1202200420	4
3	1202198957	9	1202199351	3	1202199742	4	1202200079	4	1202200424	8
4	1202198962	14	1202199354	6	1202199744	6	1202200081	6	1202200428	12
5	1202198966	18	1202199355	7	1202199746	8	1202200083	8	1202200432	16
6	1202198968	20	1202199357	9	1202199748	10	1202200085	10	1202200436	20
7	1202198971	23	1202199360	12	1202199750	12	1202200087	12	1202200438	22
8	1202198972	24	1202199363	15	1202199752	14	1202200089	14	1202200440	24
9	1202198975	27	1202199364	16	1202199754	16	1202200092	17	1202200442	26
10	1202198976	28	1202199367	19	1202199756	18	1202200094	19	1202200445	29
11	1202198978	30	1202199368	20	1202199758	20	1202200096	21	1202200445	29
12	1202198980	32	1202199371	23	1202199760	22	1202200098	23	1202200448	32
13	1202198982	34	1202199372	24	1202199762	24	1202200100	25	1202200450	34
14	1202198984	36	1202199375	27	1202199764	26	1202200102	27	1202200452	36
15	1202198986	38	1202199376	28	1202199766	28	1202200104	29	1202200453	37
16	1202198990	42	1202199379	31	1202199768	30	1202200106	31	1202200455	39
17	1202198991	43	1202199382	34	1202199770	32	1202200108	33	1202200458	42
18	1202198993	45	1202199384	36	1202199772	34	1202200110	35	1202200460	44
19	1202198993	45	1202199385	37	1202199775	37	1202200112	37	1202200462	46
20	1202198996	48	1202199387	39	1202199776	38	1202200114	39	1202200463	47
21	1202199000	52	1202199390	42	1202199777	39	1202200115	40	1202200465	49
22	1202199001	53	1202199391	43	1202199780	42	1202200118	43	1202200468	52
23	1202199003	55	1202199394	46	1202199782	44	1202200119	44	1202200470	54
24	1202199004	56	1202199395	47	1202199783	45	1202200122	47	1202200471	55
25	1202199007	59	1202199398	50	1202199786	48	1202200122	48	1202200471	58
26	1202199009	61	1202199400	52	1202199787	49	1202200125	50	1202200474	59
27	1202199010	62	1202199402	54	1202199790	52	1202200123	53	1202200473	62
28	1202199010	65	1202199402	55	1202199790	53	1202200128	54	1202200478	64
29	1202199016	68	1202199406	58	1202199791	56	1202200129	56	1202200480	65
30	1202199010	69	1202199406	59	1202199794	59	1202200131	59	1202200481	68
31	1202199017		1202199407	62	1202199797		1202200134	60	1202200484	70
	1202199019	71				61				
32		72	1202199411	63	1202199800	62	1202200138	63	1202200487	71
33	1202199023	75	1202199414	66	1202199802	64	1202200141	66	1202200490	74
34	1202199025	77	1202199415	67	1202199805	67	1202200142	67	1202200491	75
35	1202199026	78	1202199418	70	1202199806	68	1202200145	70	1202200494	78
36	1202199029	81	1202199419	71	1202199809	71	1202200147	72	1202200496	80
37	1202199030	82	1202199422	74	1202199810	72	1202200148	73	1202200497	81
38	1202199033	85	1202199424	76	1202199812	74	1202200151	76	1202200500	84
39	1202199035	87	1202199425	77	1202199815	77	1202200152	77	1202200503	87
40	1202199036	88	1202199429	81	1202199817	79	1202200155	80	1202200504	88
41	1202199039	91	1202199429	81	1202199818	80	1202200157	82	1202200507	91
42	1202199040	92	1202199432	84	1202199821	83	1202200158	83	1202200508	92
43	1202199043	95	1202199434	86	1202199823	85	1202200161	86	1202200511	95
44	1202199044	96	1202199435	87	1202199827	89	1202200164	89	1202200513	97
45	1202199047	99	1202199438	90	1202199828	90	1202200166	91	1202200514	98
46	1202199048	100	1202199439	91	1202199830	92	1202200167	92	1202200517	101
47	1202199051	103	1202199442	94	1202199832	94	1202200170	95	1202200519	103
48	1202199054	106	1202199444	96	1202199834	96	1202200172	97	1202200521	105
49	1202199056	108	1202199446	98	1202199836	98	1202200174	99	1202200523	107
50	1202199058	110	1202199448	100	1202199838	100	1202200176	101	1202200525	109
51	1202199060	112	1202199450	102	1202199840	102	1202200178	103	1202200527	111

52	1202199062	114	1202199452	104	1202199842	104	1202200180	105	1202200529	113
	1202199062		1202199452	104	1202199842	104	1202200180	105	1202200529	113
53		116								
54	1202199066	118	1202199457	109	1202199847	109	1202200185	110	1202200534	118
55	1202199069	121	1202199458	110	1202199849	111	1202200187	112	1202200536	120
56	1202199071	123	1202199461	113	1202199850	112	1202200188	113	1202200538	122
57	1202199073	125	1202199463	115	1202199854	116	1202200192	117	1202200540	124
58	1202199076	128	1202199466	118	1202199855	117	1202200193	118	1202200542	126
59	1202199078	130	1202199467	119	1202199857	119	1202200195	120	1202200545	129
60	1202199079	131	1202199469	121	1202199859	121	1202200197	122	1202200547	131
61	1202199082	134	1202199472	124	1202199861	123	1202200199	124	1202200549	133
62	1202199084	136	1202199473	125	1202199864	126	1202200201	126	1202200551	135
63	1202199086	138	1202199475	127	1202199865	127	1202200203	128	1202200554	138
64	1202199087	139	1202199477	129	1202199867	129	1202200205	130	1202200555	139
65	1202199089	141	1202199480	132	1202199869	131	1202200208	133	1202200557	141
66	1202199091	143	1202199482	134	1202199872	134	1202200209	134	1202200559	143
67	1202199094	146	1202199484	136	1202199873	135	1202200211	136	1202200562	146
68	1202199096	148	1202199486	138	1202199875	137	1202200213	138	1202200564	148
69	1202199098	150	1202199488	140	1202199877	139	1202200215	140	1202200566	150
70	1202199101	153	1202199490	142	1202199879	141	1202200218	143	1202200568	152
71		155	1202199490	144	1202199879	143		144	-	155
	1202199103						1202200219		1202200571	
72	1202199103	155	1202199494	146	1202199884	146	1202200221	146	1202200572	156
73	1202199106	158	1202199495	147	1202199885	147	1202200223	148	1202200573	157
74	1202199108	160	1202199498	150	1202199888	150	1202200224	149	1202200576	160
75	1202199110	162	1202199499	151	1202199890	152	1202200227	152	1202200578	162
76	1202199111	163	1202199501	153	1202199892	154	1202200228	153	1202200579	163
77	1202199113	165	1202199504	156	1202199894	156	1202200230	155	1202200581	165
78	1202199116	168	1202199507	159	1202199897	159	1202200233	158	1202200584	168
79	1202199119	171	1202199509	161	1202199900	162	1202200236	161	1202200586	170
80	1202199121	173	1202199512	164	1202199902	164	1202200238	163	1202200589	173
81	1202199122	174	1202199514	166	1202199903	165	1202200240	165	1202200590	174
82	1202199125	177	1202199516	168	1202199906	168	1202200242	167	1202200593	177
83	1202199127	179	1202199518	170	1202199908	170	1202200244	169	1202200596	180
84	1202199129	181	1202199520	172	1202199910	172	1202200246	171	1202200597	181
85	1202199131	183	1202199522	174	1202199912	174	1202200248	173	1202200599	183
86	1202199134	186	1202199524	176	1202199914	176	1202200251	176	1202200601	185
87	1202199137	189	1202199527	179	1202199917	179	1202200253	178	1202200604	188
88	1202199139	191	1202199529	181	1202199919	181	1202200255	180	1202200606	190
89	1202199141	193	1202199532	184	1202199921	183	1202200258	183	1202200609	193
	1202199141						1202200258		1202200610	
90		195	1202199534	186	1202199923	185		184		194
91	1202199145	197	1202199534	186	1202199925	187	1202200261	186	1202200612	196
92	1202199147	199	1202199537	189	1202199928	190	1202200263	188	1202200613	197
93	1202199150	202	1202199539	191	1202199930	192	1202200265	190	1202200616	200
94	1202199152	204	1202199541	193	1202199933	195	1202200268	193	1202200618	202
95	1202199154	206	1202199543	195	1202199934	196	1202200270	195	1202200621	205
96	1202199156	208	1202199545	197	1202199936	198	1202200272	197	1202200622	206
97	1202199158	210	1202199548	200	1202199938	200	1202200273	198	1202200625	209
98	1202199161	213	1202199549	201	1202199941	203	1202200275	200	1202200627	211
99	1202199163	215	1202199552	204	1202199943	205	1202200277	202	1202200629	213
100	1202199165	217	1202199553	205	1202199944	206	1202200279	204	1202200632	216
101	1202199167	219	1202199555	207	1202199946	208	1202200281	206	1202200634	218
102	1202199170	222	1202199557	209	1202199949	211	1202200284	209	1202200637	221
103	1202199172	224	1202199560	212	1202199951	213	1202200287	212	1202200639	223
104	1202199175	227	1202199563	215	1202199954	216	1202200289	214	1202200642	226
105	1202199177	229	1202199565	217	1202199955	217	1202200292	217	1202200643	227
106	1202199180	232	1202199566	218	1202199957	219	1202200293	218	1202200645	229
107	1202199183	235	1202199569	221	1202199960	222	1202200296	221	1202200648	232
108	1202199185	237	1202199571	223	1202199962	224	1202200298	223	1202200650	234
109	1202199186	238	1202199572	224	1202199963	225	1202200299	224	1202200652	236
110	1202199188	240	1202199575	227	1202199965	227	1202200302	227	1202200653	237

159	1202199287	339	1202199669	321	1202200062	324	1202200394	319	1202200749	333
158	1202199285	337	1202199668	320	1202200060	322	1202200393	318	1202200746	330
157	1202199282	334	1202199665	317	1202200057	319	1202200390	315	1202200745	329
156	1202199281	333	1202199664	316	1202200056	318	1202200389	314	1202200744	328
155	1202199279	331	1202199661	313	1202200053	315	1202200386	311	1202200741	325
154	1202199276	328	1202199658	310	1202200052	314	1202200385	310	1202200738	322
153	1202199275	327	1202199657	309	1202200050	312	1202200382	307	1202200737	321
152	1202199273	325	1202199655	307	1202200048	310	1202200381	306	1202200734	318
151	1202199270	322	1202199652	304	1202200045	307	1202200379	304	1202200733	317
150	1202199269	321	1202199651	303	1202200044	306	1202200376	301	1202200731	315
149	1202199267	319	1202199648	300	1202200041	303	1202200375	300	1202200728	312
148	1202199264	316	1202199647	299	1202200040	302	1202200373	298	1202200727	311
147	1202199263	315	1202199644	296	1202200038	300	1202200370	295	1202200725	309
146	1202199260	312	1202199643	295	1202200035	297	1202200369	294	1202200722	306
145	1202199259	311	1202199640	292	1202200032	294	1202200366	291	1202200720	304
144	1202199256	308	1202199639	291	1202200031	293	1202200365	290	1202200719	303
143	1202199253	305	1202199636	288	1202200028	290	1202200363	288	1202200716	300
142	1202199252	304	1202199635	287	1202200027	289	1202200360	285	1202200715	299
141	1202199249	301	1202199633	285	1202200024	286	1202200359	284	1202200712	296
140	1202199246	298	1202199630	282	1202200023	285	1202200356	281	1202200711	295
139	1202199243	295	1202199629	281	1202200020	282	1202200355	280	1202200708	292
138	1202199242	294	1202199627	279	1202200019	281	1202200352	277	1202200705	289
137	1202199240	292	1202199624	276	1202200016	278	1202200351	276	1202200704	288
136	1202199237	289	1202199623	275	1202200015	277	1202200348	273	1202200701	285
135	1202199236	288	1202199621	273	1202200013	275	1202200346	271	1202200700	284
134	1202199234	286	1202199618	270	1202200010	272	1202200345	270	1202200698	282
133	1202199231	283	1202199615	267	1202200009	271	1202200343	268	1202200696	280
132	1202199228	280	1202199614	266	1202200007	269	1202200341	266	1202200694	278
131	1202199227	279	1202199612	264	1202200005	267	1202200339	264	1202200692	276
130	1202199226	278	1202199612	264	1202200004	266	1202200338	263	1202200691	275
129	1202199225	277	1202199610	262	1202200003	265	1202200337	262	1202200690	274
128	1202199223	275	1202199609	261	1202200002	264	1202200335	260	1202200688	272
127	1202199222	274	1202199607	259	1202200000	262	1202200334	259	1202200687	271
126	1202199220	272	1202199605	257	1202199998	260	1202200332	257	1202200685	269
125	1202199218	270	1202199602	254	1202199995	257	1202200330	255	1202200683	267
124	1202199216	268	1202199601	253	1202199994	256	1202200328	253	1202200680	264
123	1202199213	265	1202199599	251	1202199992	254	1202200326	251	1202200679	263
122	1202199212	264	1202199597	249	1202199991	253	1202200323	248	1202200676	260
121	1202199210	262	1202199595	247	1202199988	250	1202200321	246	1202200675	259
120	1202199207	259	1202199593	245	1202199986	248	1202200320	245	1202200673	257
119	1202199206	258	1202199591	243	1202199984	246	1202200319	244	1202200672	256
118	1202199204	256	1202199590	242	1202199982	244	1202200316	241	1202200669	253
117	1202199201	253	1202199587	239	1202199979	241	1202200314	239	1202200667	251
116	1202199200	252	1202199585	237	1202199977	239	1202200312	237	1202200665	249
115	1202199197	249	1202199583	235	1202199975	237	1202200310	235	1202200662	246
114	1202199195	247	1202199581	233	1202199972	234	1202200309	234	1202200660	244
113	1202199193	245	1202199580	232	1202199970	232	1202200307	232	1202200658	242
112	1202199191	243	1202199577	229	1202199968	230	1202200305	230	1202200656	240
111	1202199190	242	1202199576	228	1202199967	229	1202200302	227	1202200655	239

Condor	Condor Mandelbrot 16 Machines													
Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A				
1	1202204733	0	1202205085	0	1202205496	0	1202205867	0	1202206213	0				
2	1202204737	4	1202205086	1	1202205498	2	1202205869	2	1202206215	2				
3	1202204741	8	1202205089	4	1202205499	3	1202205871	4	1202206217	4				
4	1202204745	12	1202205091	6	1202205501	5	1202205874	7	1202206219	6				
5	1202204749	16	1202205094	9	1202205503	7	1202205875	8	1202206221	8				
6	1202204753	20	1202205096	11	1202205506	10	1202205879	12	1202206223	10				

17	1202204755	1 22	1202205000	12	1202205509	12	1202205001	L ₁₄	1202206225	12
7	1202204755	22	1202205098	13	1202205508	12	1202205881	14	1202206225	12
8	1202204757	24	1202205100	15	1202205509	13	1202205884	17	1202206228	15
9	1202204759	26	1202205102	17	1202205512	16	1202205885	18	1202206229	16
10	1202204761	28	1202205104	19	1202205513	17	1202205887	20	1202206232	19
11	1202204763	30	1202205106	21	1202205515	19	1202205889	22	1202206234	21
12	1202204765	32	1202205108	23	1202205518	22	1202205891	24	1202206236	23
13	1202204767	34	1202205110	25	1202205519	23	1202205893	26	1202206238	25
14	1202204769	36	1202205112	27	1202205521	25	1202205895	28	1202206239	26
15	1202204771	38	1202205114	29	1202205524	28	1202205897	30	1202206242	29
16	1202204772	39	1202205116	31	1202205525	29	1202205899	32	1202206244	31
17	1202204774	41	1202205118	33	1202205528	32	1202205901	34	1202206246	33
18	1202204777	44	1202205120	35	1202205529	33	1202205903	36	1202206248	35
19	1202204779	46	1202205122	37	1202205532	36	1202205905	38	1202206250	37
20	1202204780	47	1202205124	39	1202205533	37	1202205907	40	1202206251	38
21	1202204782	49	1202205125	40	1202205535	39	1202205909	42	1202206254	41
22	1202204784	51	1202205128	43	1202205539	43	1202205910	43	1202206254	42
	1202204787	54		44						
23			1202205129		1202205540	44	1202205913	46	1202206259	46
24	1202204789	56	1202205132	47	1202205541	45	1202205915	48	1202206259	46
25	1202204790	57	1202205133	48	1202205544	48	1202205916	49	1202206262	49
26	1202204793	60	1202205135	50	1202205545	49	1202205919	52	1202206263	50
27	1202204795	62	1202205138	53	1202205548	52	1202205920	53	1202206266	53
28	1202204796	63	1202205140	55	1202205551	55	1202205923	56	1202206267	54
29	1202204800	67	1202205142	57	1202205553	57	1202205924	57	1202206270	57
30	1202204801	68	1202205143	58	1202205554	58	1202205927	60	1202206273	60
31	1202204804	71	1202205146	61	1202205556	60	1202205930	63	1202206275	62
32	1202204805	72	1202205148	63	1202205560	64	1202205931	64	1202206276	63
33	1202204808	75	1202205149	64	1202205560	64	1202205933	66	1202206278	65
34	1202204810	77	1202205151	66	1202205563	67	1202205936	69	1202206281	68
35	1202204811	78	1202205154	69	1202205565	69	1202205938	71	1202206282	69
36	1202204813	80	1202205157	72	1202205567	71	1202205940	73	1202206285	72
37	1202204816	83	1202205158	73	1202205569	73	1202205942	75	1202206286	73
38	1202204817	84	1202205160	75	1202205570	74	1202205943	76	1202206289	76
39	1202204820	87	1202205162	77	1202205573	77	1202205947	80	1202206291	78
40	1202204823	90	1202205165	80	1202205576	80	1202205948	81	1202206293	80
41	1202204824	91	1202205167	82	1202205578	82	1202205952	85	1202206294	81
42		94		83		83		86		84
	1202204827		1202205168		1202205579		1202205953		1202206297	
43	1202204828	95	1202205170	85	1202205582	86	1202205955	88	1202206298	85
44	1202204831	98	1202205173	88	1202205584	88	1202205956	89	1202206301	88
45	1202204832	99	1202205176	91	1202205586	90	1202205958	91	1202206303	90
46	1202204835	102	1202205177	92	1202205587	91	1202205961	94	1202206304	91
47	1202204837	104	1202205180	95	1202205589	93	1202205963	96	1202206307	94
48	1202204839	106	1202205182	97	1202205592	96	1202205965	98	1202206311	98
49	1202204841	108	1202205184	99	1202205594	98	1202205967	100	1202206313	100
50	1202204843	110	1202205186	101	1202205596	100	1202205969	102	1202206315	102
51	1202204845	112	1202205188	103	1202205598	102	1202205971	104	1202206317	104
52	1202204847	114	1202205190	105	1202205600	104	1202205973	106	1202206319	106
53	1202204849	116	1202205192	107	1202205602	106	1202205975	108	1202206321	108
54	1202204852	119	1202205194	109	1202205604	108	1202205977	110	1202206324	111
55	1202204854	121	1202205197	112	1202205607	111	1202205980	113	1202206325	112
56	1202204856	123	1202205199	114	1202205609	113	1202205982	115	1202206328	115
57	1202204859	126	1202205202	117	1202205612	116	1202205985	118	1202206320	118
58	1202204860	127	1202205203	118	1202205613	117	1202205987	120	1202206333	120
59	1202204860	127	1202205205	120	1202205615	117	1202205987	120	1202206335	120
60	1202204864	131	1202205207	122	1202205617	121	1202205991	124	1202206337	124
61	1202204866	133	1202205209	124	1202205620	124	1202205994	127	1202206339	126
62	1202204869	136	1202205212	127	1202205621	125	1202205996	129	1202206341	128
63	1202204871	138	1202205214	129	1202205623	127	1202205997	130	1202206344	131
64	1202204872	139	1202205215	130	1202205626	130	1202205999	132	1202206346	133
65	1202204875	142	1202205217	132	1202205628	132	1202206001	134	1202206347	134

66	1202204876	143	1202205219	134	1202205630	134	1202206004	137	1202206349	136
	1202204876	-					1202206004			
67		145	1202205222	137	1202205632	136		139	1202206351	138
68	1202204880	147	1202205223	138	1202205634	138	1202206007	140	1202206353	140
69	1202204883	150	1202205225	140	1202205636	140	1202206009	142	1202206355	142
70	1202204884	151	1202205227	142	1202205638	142	1202206011	144	1202206358	145
71	1202204886	153	1202205229	144	1202205640	144	1202206013	146	1202206360	147
72	1202204888	155	1202205230	145	1202205642	146	1202206015	148	1202206362	149
73	1202204890	157	1202205233	148	1202205644	148	1202206017	150	1202206364	151
74	1202204892	159	1202205235	150	1202205646	150	1202206019	152	1202206366	153
75	1202204894	161	1202205237	152	1202205649	153	1202206021	154	1202206368	155
76	1202204896	163	1202205239	154	1202205650	154	1202206022	155	1202206371	158
77	1202204899	166	1202205242	157	1202205652	156	1202206025	158	1202206372	159
78	1202204902	169	1202205245	160	1202205655	159	1202206028	161	1202206375	162
79	1202204904	171	1202205246	161	1202205658	162	1202206030	163	1202206378	165
80	1202204907	174	1202205249	164	1202205659	163	1202206033	166	1202206380	167
81	1202204909	176	1202205251	166	1202205662	166	1202206035	168	1202206383	170
82	1202204911	178	1202205253	168	1202205664	168	1202206036	169	1202206386	173
83	1202204913	180	1202205255	170	1202205666	170	1202206039	172	1202206388	175
84	1202204915	182	1202205257	172	1202205669	173	1202206041	174	1202206390	177
85	1202204918	185	1202205259	174	1202205670	174	1202206043	176	1202206392	179
86	1202204919	186	1202205262	177	1202205672	176	1202206045	178	1202206395	182
87	1202204922	189	1202205263	178	1202205675	179	1202206048	181	1202206397	184
88	1202204925	192	1202205266	181	1202205678	182	1202206050	183	1202206399	186
89	1202204927	194	1202205268	183	1202205679	183	1202206053	186	1202206401	188
90	1202204928	195	1202205270	185	1202205681	185	1202206054	187	1202206404	191
91	1202204930	197	1202205270	187	1202205683	187	1202206054	189	1202206406	193
92	1202204930	199		190		189	1202206059	192		195
			1202205275		1202205685				1202206408	
93	1202204934	201	1202205276	191	1202205687	191	1202206061	194	1202206411	198
94	1202204937	204	1202205278	193	1202205688	192	1202206063	196	1202206413	200
95	1202204938	205	1202205280	195	1202205692	196	1202206066	199	1202206415	202
96	1202204941	208	1202205283	198	1202205693	197	1202206068	201	1202206418	205
97	1202204942	209	1202205284	199	1202205695	199	1202206070	203	1202206420	207
98	1202204944	211	1202205286	201	1202205697	201	1202206073	206	1202206423	210
99	1202204946	213	1202205288	203	1202205699	203	1202206075	208	1202206425	212
100	1202204949	216	1202205291	206	1202205702	206	1202206078	211	1202206428	215
101	1202204951	218	1202205292	207	1202205703	207	1202206080	213	1202206429	216
102	1202204953	220	1202205295	210	1202205707	211	1202206081	214	1202206432	219
103	1202204956	223	1202205297	212	1202205709	213	1202206084	217	1202206435	222
104	1202204959	226	1202205299	214	1202205712	216	1202206086	219	1202206437	224
105	1202204960	227	1202205301	216	1202205713	217	1202206089	222	1202206440	227
106	1202204963	230	1202205304	219	1202205716	220	1202206091	224	1202206441	228
107	1202204965	232	1202205306	221	1202205718	222	1202206093	226	1202206444	231
108	1202204967	234	1202205308	223	1202205719	223	1202206095	228	1202206446	233
109	1202204969	236	1202205309	224	1202205721	225	1202206097	230	1202206447	234
110	1202204970	237	1202205311	226	1202205723	227	1202206099	232	1202206450	237
111	1202204972	239	1202205313	228	1202205725	229	1202206101	234	1202206451	238
112	1202204973	240	1202205314	229	1202205727	231	1202206102	235	1202206454	241
113	1202204975	242	1202205316	231	1202205728	232	1202206105	238	1202206456	243
114	1202204977	244	1202205318	233	1202205730	234	1202206106	239	1202206459	246
115	1202204979	246	1202205320	235	1202205732	236	1202206108	241	1202206460	247
116	1202204982	249	1202205323	238	1202205734	238	1202206110	243	1202206463	250
117	1202204984	251	1202205325	240	1202205737	241	1202206112	245	1202206465	252
118	1202204985	252	1202205327	242	1202205739	243	1202206115	248	1202206467	254
119	1202204987	254	1202205327	244	1202205742	246	1202206116	249	1202206470	257
						246				257
120	1202204989	256	1202205331	246	1202205742		1202206118	251	1202206472	
121	1202204001	259		144	1202205745	249	1202206120	253	1202206474	261
122	1202204991	258	1202205334		1202205747	251	1202206122	255	1202206477	264
122	1202204993	260	1202205336	251	1202205747	251	1202206122	255	1202206477	264
122 123 124					1202205747 1202205748 1202205750	251 252 254	1202206122 1202206124 1202206127	255 257 260	1202206477 1202206479 1202206480	264 266 267

125	1202204999	266	1202205342	257	1202205753	257	1202206130	263	1202206483	270
126	1202205000	267	1202205344	259	1202205755	259	1202206130	263	1202206485	272
127	1202205002	269	1202205345	260	1202205757	261	1202206133	266	1202206487	274
128	1202205005	272	1202205348	263	1202205759	263	1202206134	267	1202206488	275
129	1202205005	272	1202205350	265	1202205761	265	1202206135	268	1202206489	276
130	1202205007	274	1202205352	267	1202205761	265	1202206137	270	1202206491	278
131	1202205009	276	1202205353	268	1202205763	267	1202206139	272	1202206492	279
132	1202205011	278	1202205355	270	1202205765	269	1202206140	273	1202206494	281
133	1202205012	279	1202205357	272	1202205767	271	1202206142	275	1202206495	282
134	1202205015	282	1202205358	273	1202205768	272	1202206143	276	1202206497	284
135	1202205017	284	1202205361	276	1202205771	275	1202206146	279	1202206500	287
136	1202205018	285	1202205363	278	1202205772	276	1202206147	280	1202206502	289
137	1202205021	288	1202205364	279	1202205775	279	1202206150	283	1202206503	290
138	1202205022	289	1202205367	282	1202205778	282	1202206152	285	1202206506	293
139	1202205025	292	1202205370	285	1202205780	284	1202206154	287	1202206508	295
140	1202205027	294	1202205373	288	1202205781	285	1202206155	288	1202206510	297
141	1202205028	295	1202205375	290	1202205784	288	1202206158	291	1202206512	299
142	1202205031	298	1202205376	291	1202205786	290	1202206159	292	1202206513	300
143	1202205032	299	1202205379	294	1202205787	291	1202206162	295	1202206517	304
144	1202205035	302	1202205380	295	1202205790	294	1202206164	297	1202206517	304
145	1202205038	305	1202205383	298	1202205792	296	1202206166	299	1202206520	307
146	1202205041	308	1202205384	299	1202205793	297	1202206167	300	1202206522	309
147	1202205043	310	1202205387	302	1202205796	300	1202206170	303	1202206523	310
148	1202205044	311	1202205389	304	1202205797	301	1202206173	306	1202206526	313
149	1202205047	314	1202205390	305	1202205800	304	1202206174	307	1202206528	315
150	1202205048	315	1202205393	308	1202205802	306	1202206177	310	1202206530	317
151	1202205051	318	1202205394	309	1202205803	307	1202206178	311	1202206531	318
152	1202205053	320	1202205397	312	1202205806	310	1202206181	314	1202206534	321
153	1202205054	321	1202205399	314	1202205807	311	1202206182	315	1202206537	324
154	1202205058	325	1202205400	315	1202205810	314	1202206185	318	1202206538	325
155	1202205059	326	1202205403	318	1202205811	315	1202206186	319	1202206541	328
156	1202205062	329	1202205405	320	1202205814	318	1202206189	322	1202206544	331
157	1202205063	330	1202205406	321	1202205816	320	1202206191	324	1202206545	332
158	1202205066	333	1202205409	324	1202205818	322	1202206192	325	1202206548	335
159	1202205067	334	1202205412	327	1202205819	323	1202206195	328	1202206550	337
160	1202205070	337	1202205414	329	1202205822	326	1202206196	329	1202206553	340

BOINC

BOING	C Mandelbro	ot 1 Mach	ine							
Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A
1	1202432713	0	1202435713	0	1202438480	0	1202441775	0	1202444597	0
2	1202432728	15	1202435729	16	1202438500	20	1202441790	15	1202444618	21
3	1202432743	30	1202435744	31	1202438515	35	1202441805	30	1202444634	37
4	1202432759	46	1202435763	50	1202438531	51	1202441821	46	1202444654	57
5	1202432775	62	1202435780	67	1202438551	71	1202441836	61	1202444669	72
6	1202432789	76	1202435800	87	1202438567	87	1202441851	76	1202444689	92
7	1202432805	92	1202435815	102	1202438582	102	1202441867	92	1202444705	108
8	1202432820	107	1202435831	118	1202438597	117	1202441881	106	1202444720	123
9	1202432835	122	1202435846	133	1202438613	133	1202441897	122	1202444735	138
10	1202432851	138	1202435862	149	1202438629	149	1202441913	138	1202444751	154
11	1202432866	153	1202435881	168	1202438643	163	1202441928	153	1202444766	169
12	1202432881	168	1202435897	184	1202438659	179	1202441948	173	1202444781	184
13	1202432901	188	1202435913	200	1202438674	194	1202441963	188	1202444802	205
14	1202432921	208	1202435932	219	1202438689	209	1202441983	208	1202444817	220
15	1202432942	229	1202435948	235	1202438705	225	1202441999	224	1202444832	235
16	1202432957	244	1202435968	255	1202438724	244	1202442014	239	1202444848	251
17	1202432978	265	1202435984	271	1202438740	260	1202442030	255	1202444863	266

10	120242222	200	120242522	200	1202420=55	200	12024422	1200	1202444575	201
18	1202432993	280	1202435999	286	1202438760 1202438781	280	1202442044	269	1202444878	281
20	1202433008	295 310	1202436015	302	1202438781	301	1202442060 1202442076	301	1202444894	312
21	1202433023	325	1202436029	333	1202438790	331	1202442076	316	1202444909	327
22	1202433054	341	1202436060	347	1202438811	347	1202442091	331	1202444924	347
23	1202433070	357	1202436076	363	1202438842	362	1202442106	351	1202444964	367
24	1202433076	372	1202436076	383	1202438857	377	1202442126	371	1202444980	383
25	1202433100	387	1202436070	399	1202438877	393	1202442140	388	1202444995	398
26	1202433116	403	1202436127	414	1202438888	408	1202442103	402	1202445016	419
27	1202433116	422	1202436127	434	1202438888	423	1202442177	419	1202445031	434
28	1202433156	443	1202436147	455	1202438919	439	1202442208	433	1202445050	453
29	1202433172	459	1202436183	470	1202438934	454	1202442223	448	1202445067	470
30	1202433187	474	1202436198	485	1202438949	469	1202442239	464	1202445082	485
31	1202433202	489	1202436218	505	1202438965	485	1202442253	478	1202445097	500
32	1202433222	509	1202436239	526	1202438980	500	1202442269	494	1202445118	521
33	1202433238	525	1202436254	541	1202438994	514	1202442285	510	1202445133	536
34	1202433254	541	1202436269	556	1202439016	536	1202442300	525	1202445148	551
35	1202433269	556	1202436285	572	1202439031	551	1202442315	540	1202445168	571
36	1202433289	576	1202436300	587	1202439046	566	1202442336	561	1202445184	587
37	1202433304	591	1202436319	606	1202439061	581	1202442351	576	1202445199	602
38	1202433319	606	1202436336	623	1202439078	598	1202442366	591	1202445214	617
39	1202433334	621	1202436351	638	1202439092	612	1202442382	607	1202445230	633
40	1202433350	637	1202436366	653	1202439112	632	1202442398	623	1202445250	653
41	1202433365	652	1202436382	669	1202439129	649	1202442412	637	1202445266	669
42	1202433381	668	1202436398	685	1202439144	664	1202442428	653	1202445280	683
43	1202433396	683	1202436412	699	1202439158	678	1202442443	668	1202445296	699
44	1202433411	698	1202436432	719	1202439173	693	1202442463	688	1202445312	715
45	1202433427	714	1202436449	736	1202439189	709	1202442483	708	1202445332	735
46	1202433442	729	1202436462	749	1202439204	724	1202442499	724	1202445352	755
47	1202433457	744	1202436479	766	1202439223	743	1202442519	744	1202445372	775
48	1202433473	760	1202436494	781	1202439240	760	1202442534	759	1202445392	795
49	1202433488	775	1202436509	796	1202439255	775	1202442555	780	1202445408	811
50	1202433508	795	1202436526	813	1202439270	790	1202442570	795	1202445423	826
51	1202433529	816	1202436539	826	1202439285	805	1202442585	810	1202445443	846
52	1202433549	836	1202436556	843	1202439301	821	1202442600	825	1202445458	861
53	1202433564	851	1202436571	858	1202439320	840	1202442616	841	1202445479	882
54	1202433580	867	1202436586	873	1202439337	857	1202442631	856	1202445494	897
55	1202433595	882	1202436601	888	1202439352	872	1202442646	871	1202445514	917
56	1202433615	902	1202436617	904	1202439367	887	1202442667	892	1202445529	932
57	1202433631	918	1202436637	924	1202439384	904	1202442682	907	1202445550	953
58	1202433651	938	1202436652	939	1202439397	917	1202442697	922	1202445569	972
59	1202433666	953	1202436673	960	1202439413	933	1202442713	938	1202445585	988
60	1202433681	968	1202436688	975	1202439428	948	1202442734	959	1202445601	1004
61	1202433701	988	1202436703	990	1202439444	964	1202442748	973	1202445616	1019
62	1202433721	1008	1202436724	1011	1202439459	979	1202442769	994	1202445632	1035
63	1202433737	1024	1202436739	1026	1202439474	994	1202442783	1008	1202445646	1049
64	1202433752	1039	1202436754	1041	1202439490	1010	1202442799	1024	1202445662	1065
65	1202433772	1059	1202436770	1057	1202439505	1025	1202442815	1040	1202445677	1080
66	1202433788	1075	1202436784	1071	1202439520	1040	1202442830	1055	1202445691	1094
67	1202433807	1094	1202436800	1087	1202439536	1056	1202442845	1070	1202445708	1111
68	1202433824	1111	1202436816	1103	1202439551	1071	1202442861	1086	1202445723	1126
69	1202433839	1126	1202436831	1118	1202439566	1086	1202442876	1101	1202445738	1141
70	1202433854	1141	1202436846	1133	1202439586	1106	1202442891	1116	1202445754	1157
71	1202433869	1156	1202436862	1149	1202439602	1122	1202442906	1131	1202445769	1172
72	1202433885	1172	1202436881	1168	1202439617	1137	1202442922	1147	1202445784	1187
73	1202433900	1187	1202436901	1188	1202439633	1153	1202442937	1162	1202445804	1207
74	1202433920	1207	1202436917	1204	1202439648	1168	1202442958	1183	1202445820	1223
75	1202433936	1223	1202436933	1220	1202439663	1183	1202442977	1202	1202445835	1238
75	1202 133730		1202100700	1220	1202.57005	1103	1	1202	1202	1-00

l	l	1.050	1	1	1	1	1,,,,,,,,,,,	1.2.0	1	
77	1202433966	1253	1202436968	1255	1202439694	1214	1202442993	1218	1202445871	1274
78	1202433982	1269	1202436984	1271	1202439709	1229	1202443024	1249	1202445886	1289
79	1202434002	1289	1202436999	1286	1202439730	1250	1202443039	1264	1202445902	1305
80	1202434022	1309	1202437019	1306	1202439745	1265	1202443054	1279	1202445921	1324
81	1202434038	1325	1202437038	1325	1202439760	1280	1202443075	1300	1202445937	1340
82	1202434053	1340	1202437055	1342	1202439776	1296	1202443095	1320	1202445958	1361
83	1202434068	1355	1202437070	1357	1202439791	1311	1202443110	1335	1202445973	1376
84	1202434089	1376	1202437090	1377	1202439811	1331	1202443131	1356	1202445989	1392
85	1202434104	1391	1202437106	1393	1202439827	1347	1202443146	1371	1202446005	1408
86	1202434119	1406	1202437121	1408	1202439842	1362	1202443166	1391	1202446035	1438
87	1202434135	1422	1202437136	1423	1202439862	1382	1202443182	1407	1202446051	1454
88	1202434150	1437	1202437152	1439	1202439879	1399	1202443196	1421	1202446065	1468
90	1202434165	1452 1467	1202437167	1454	1202439893	1413	1202443212	1437	1202446081	1484
91	1202434180	1483	1202437182	1490	1202439913	1449	1202443228	1468	1202446096	1515
92	1202434196	1498	1202437203	1505	1202439929	1468	1202443243	1482	1202446112	1530
93				1519		1489		1504		1551
94	1202434227	1514	1202437232		1202439969	1505	1202443279		1202446148	
95	1202434246	1533 1549	1202437249	1536 1556	1202439985	1520	1202443294	1519	1202446164	1567
96				1571		1540	1202443309		1202446177	1580
96	1202434277	1564 1580	1202437284	15/1	1202440020	1540	1202443325	1550 1565	1202446194 1202446209	1612
98	1202434293	1595	1202437300	1602	1202440041	1580	1202443340	1580	1202446209	1627
99	1202434308	1611	1202437313	1616	1202440060	1596	1202443333	1596	1202446224	1648
100	1202434339	1626	1202437329	1633	1202440092	1612	1202443386	1611	1202446260	1663
101	1202434354	1641	1202437340	1648	1202440107	1627	1202443401	1626	1202446276	1679
102	1202434370	1657	1202437376	1663	1202440122	1642	1202443422	1647	1202446291	1694
103	1202434390	1677	1202437392	1679	1202440137	1657	1202443437	1662	1202446306	1709
104	1202434405	1692	1202437411	1698	1202440153	1673	1202443452	1677	1202446381	1784
105	1202434420	1707	1202437427	1714	1202440169	1689	1202443468	1693	1202446381	1784
106	1202434436	1723	1202437447	1734	1202440184	1704	1202443483	1708	1202446381	1784
107	1202434451	1738	1202437463	1750	1202440199	1719	1202443498	1723	1202446416	1819
108	1202434467	1754	1202437478	1765	1202440214	1734	1202443513	1738	1202446432	1835
109	1202434481	1768	1202437494	1781	1202440230	1750	1202443530	1755	1202446447	1850
110	1202434497	1784	1202437509	1796	1202440245	1765	1202443545	1770	1202446462	1865
111	1202434497	1784	1202437524	1811	1202440260	1780	1202443560	1785	1202446478	1881
112	1202434533	1820	1202437540	1827	1202440280	1800	1202443574	1799	1202446493	1896
113	1202434548	1835	1202437555	1842	1202440300	1820	1202443590	1815	1202446508	1911
114	1202434563	1850	1202437570	1857	1202440316	1836	1202443605	1830	1202446524	1927
115	1202434583	1870	1202437586	1873	1202440331	1851	1202443621	1846	1202446539	1942
116	1202434598	1885	1202437606	1893	1202440347	1867	1202443636	1861	1202446555	1958
117	1202434614	1901	1202437621	1908	1202440366	1886	1202443652	1877	1202446570	1973
118	1202434630	1917	1202437637	1924	1202440382	1902	1202443667	1892	1202446590	1993
119	1202434645	1932	1202437652	1939	1202440402	1922	1202443682	1907	1202446609	2012
120	1202434665	1952	1202437667	1954	1202440418	1938	1202443701	1926	1202446626	2029
121	1202434686	1973	1202437683	1970	1202440437	1957	1202443718	1943	1202446641	2044
122	1202434701	1988	1202437703	1990	1202440453	1973	1202443733	1958	1202446661	2064
123	1202434716	2003	1202437723	2010	1202440469	1989	1202443749	1974	1202446680	2083
124	1202434737	2024	1202437743	2030	1202440484	2004	1202443764	1989	1202446697	2100
125	1202434752	2039	1202437759	2046	1202440499	2019	1202443780	2005	1202446712	2115
126	1202434767	2054	1202437774	2061	1202440515	2035	1202443794	2019	1202446727	2130
127	1202434784	2071	1202437789	2076	1202440535	2055	1202443810	2035	1202446743	2146
128	1202434798	2085	1202437806	2093	1202440550	2070	1202443830	2055	1202446758	2161
129	1202434813	2100	1202437820	2107	1202440571	2091	1202443845	2070	1202446773	2176
130	1202434828	2115	1202437835	2122	1202440586	2106	1202443861	2086	1202446789	2192
131	1202434844	2131	1202437851	2138	1202440601	2121	1202443876	2101	1202446804	2207
132	1202434859	2146	1202437866	2153	1202440621	2141	1202443890	2115	1202446819	2222
133	1202434875	2162	1202437881	2168	1202440637	2157	1202443906	2131	1202446835	2238
134	1202434890	2177	1202437897	2184	1202440652	2172	1202443923	2148	1202446855	2258
135	1202434904	2191	1202437912	2199	1202440672	2192	1202443937	2162	1202446875	2278

136	1202434921	2208	1202437927	2214	1202440692	2212	1202443952	2177	1202446891	2294
137	1202434937	2224	1202437943	2230	1202440708	2228	1202443968	2193	1202446906	2309
138	1202434951	2238	1202437958	2245	1202440722	2242	1202443983	2208	1202446926	2329
139	1202434967	2254	1202437978	2265	1202440739	2259	1202443998	2223	1202446942	2345
140	1202434982	2269	1202437994	2281	1202440758	2278	1202444017	2242	1202446957	2360
141	1202434997	2284	1202438009	2296	1202440778	2298	1202444039	2264	1202446972	2375
142	1202435013	2300	1202438028	2315	1202440798	2318	1202444054	2279	1202446988	2391
143	1202435029	2316	1202438045	2332	1202440814	2334	1202444070	2295	1202447003	2406
144	1202435043	2330	1202438064	2351	1202440830	2350	1202444090	2315	1202447018	2421
145	1202435058	2345	1202438080	2367	1202440845	2365	1202444105	2330	1202447034	2437
146	1202435074	2361	1202438096	2383	1202440861	2381	1202444121	2346	1202447049	2452
147	1202435089	2376	1202438111	2398	1202440877	2397	1202444136	2361	1202447064	2467
148	1202435104	2391	1202438131	2418	1202440911	2431	1202444151	2376	1202447080	2483
149	1202435120	2407	1202438147	2434	1202440932	2452	1202444167	2392	1202447099	2502
150	1202435135	2422	1202438162	2449	1202440947	2467	1202444183	2408	1202447115	2518
151	1202435150	2437	1202438177	2464	1202440967	2487	1202444197	2422	1202447131	2534
152	1202435171	2458	1202438193	2480	1202440983	2503	1202444212	2437	1202447146	2549
153	1202435186	2473	1202438209	2496	1202441000	2520	1202444228	2453	1202447161	2564
154	1202435201	2488	1202438224	2511	1202441016	2536	1202444243	2468	1202447176	2579
155	1202435222	2509	1202438239	2526	1202441029	2549	1202444258	2483	1202447192	2595
156	1202435237	2524	1202438254	2541	1202441044	2564	1202444274	2499	1202447207	2610
157	1202435252	2539	1202438274	2561	1202441060	2580	1202444294	2519	1202447223	2626
158	1202435273	2560	1202438290	2577	1202441075	2595	1202444313	2538	1202447242	2645
159	1202435288	2575	1202438304	2591	1202441141	2661	1202444330	2555	1202447258	2661
160	1202435353	2640	1202438370	2657	1202441270	2790	1202444395	2620	1202447323	2726

								1		
Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A
1	1202357242	0	1202425099	0	1202427021	0	1202428763	0	1202430517	0
2	1202357244	2	1202425104	5	1202427032	11	1202428764	1	1202430518	1
3	1202357257	15	1202425119	20	1202427038	17	1202428780	17	1202430532	15
4	1202357259	17	1202425119	20	1202427046	25	1202428780	17	1202430538	21
5	1202357274	32	1202425135	36	1202427052	31	1202428795	32	1202430547	30
6	1202357274	32	1202425139	40	1202427062	41	1202428796	33	1202430554	37
7	1202357288	46	1202425149	50	1202427071	50	1202428810	47	1202430563	46
8	1202357289	47	1202425155	56	1202427078	57	1202428815	52	1202430569	52
9	1202357305	63	1202425166	67	1202427088	67	1202428825	62	1202430579	62
10	1202357305	63	1202425170	71	1202427092	71	1202428835	72	1202430590	73
11	1202357319	77	1202425181	82	1202427103	82	1202428841	78	1202430598	81
12	1202357320	78	1202425190	91	1202427108	87	1202428851	88	1202430605	88
13	1202357334	92	1202425197	98	1202427118	97	1202428856	93	1202430615	98
14	1202357340	98	1202425207	108	1202427123	102	1202428866	103	1202430620	103
15	1202357351	109	1202425212	113	1202427135	114	1202428871	108	1202430630	113
16	1202357355	113	1202425225	126	1202427142	121	1202428881	118	1202430635	118
17	1202357366	124	1202425226	127	1202427154	133	1202428887	124	1202430646	129
18	1202357371	129	1202425241	142	1202427159	138	1202428897	134	1202430655	138
19	1202357380	138	1202425243	144	1202427169	148	1202428907	144	1202430675	158
20	1202357386	144	1202425257	158	1202427174	153	1202428912	149	1202430676	159
21	1202357395	153	1202425257	158	1202427184	163	1202428921	158	1202430690	173
22	1202357401	159	1202425272	173	1202427189	168	1202428927	164	1202430691	174
23	1202357411	169	1202425278	179	1202427200	179	1202428937	174	1202430706	189
24	1202357416	174	1202425288	189	1202427205	184	1202428948	185	1202430707	190
25	1202357431	189	1202425292	193	1202427220	199	1202428957	194	1202430721	204
26	1202357432	190	1202425303	204	1202427225	204	1202428963	200	1202430722	205
27	1202357447	205	1202425309	210	1202427236	215	1202428973	210	1202430737	220
28	1202357451	209	1202425318	219	1202427245	224	1202428978	215	1202430741	224
29	1202357462	220	1202425328	229	1202427251	230	1202428992	229	1202430751	234
30	1202357468	226	1202425334	235	1202427260	239	1202428994	231	1202430762	245
	1202357478	236	1202425344	245	1202427271	250	1202429009	246	1202430766	249

32	1202357482	240	1202425349	250	1202427276	255	1202429014	251	1202430778	261
33	1202357482	251	1202425349	260	1202427276	265	1202429014	264	12024307/8	266
34	1202357498	256	1202425364	265	1202427291	270	1202429028	265	1202430793	276
35	1202357511	269	1202425378	279	1202427301	280	1202429044	281	1202430797	280
36	1202357513	271	1202425385	286	1202427305	284	1202429045	282	1202430809	292
37	1202357527	285	1202425394	295	1202427318	297	1202429063	300	1202430813	296
38	1202357533	291	1202425400	301	1202427322	301	1202429065	302	1202430823	306
39	1202357543	301	1202425410	311	1202427333	312	1202429079	316	1202430828	311
40	1202357553	311	1202425415	316	1202427337	316	1202429080	317	1202430839	322
41	1202357558	316	1202425426	327	1202427347	326	1202429094	331	1202430848	331
42	1202357568	326	1202425436	337	1202427357	336	1202429095	332	1202430854	337
43	1202357572	330	1202425441	342	1202427363	342	1202429110	347	1202430869	352
44	1202357584	342	1202425451	352	1202427373	352	1202429115	352	1202430869	352
45	1202357589	347	1202425455	356	1202427379	358	1202429125	362	1202430884	367
46	1202357599	357	1202425466	367	1202427388	367	1202429131	368	1202430885	368
47	1202357605	363	1202425475	376	1202427393	372	1202429140	377	1202430900	383
48	1202357614	372	1202425482	383	1202427403	382	1202429145	382	1202430900	383
49	1202357620	378	1202425491	392	1202427409	388	1202429156	393	1202430915	398
50	1202357634	392	1202425497	398	1202427419	398	1202429162	399	1202430915	398
51	1202357634	392	1202425507	408	1202427424	403	1202429171	408	1202430932	415
52	1202357649	407	1202425517	418	1202427434	413	1202429181	418	1202430935	418
53	1202357650	408	1202425517	422	1202427439	418	1202429186	423	1202430947	430
54	1202357665	423	1202425536	437	1202427453	432	1202429180	434	1202430947	445
		423		437				439		
55	1202357665		1202425538		1202427459	438	1202429202		1202430977	460
56	1202357680	438	1202425553	454	1202427470	449	1202429212	449	1202430993	476
57	1202357685	443	1202425557	458	1202427475	454	1202429216	453	1202431008	491
58	1202357696	454	1202425568	469	1202427485	464	1202429232	469	1202431016	499
59	1202357705	463	1202425573	474	1202427491	470	1202429233	470	1202431016	499
60	1202357711	469	1202425585	486	1202427501	480	1202429248	485	1202431016	499
61	1202357721	479	1202425588	489	1202427505	484	1202429253	490	1202431023	506
62	1202357727	485	1202425599	500	1202427515	494	1202429263	500	1202431039	522
63	1202357736	494	1202425603	504	1202427520	499	1202429272	509	1202431046	529
64	1202357743	501	1202425614	515	1202427531	510	1202429279	516	1202431055	538
65	1202357751	509	1202425618	519	1202427537	516	1202429290	527	1202431060	543
66	1202357756	514	1202425630	531	1202427546	525	1202429293	530	1202431069	552
67	1202357766	524	1202425633	534	1202427552	531	1202429304	541	1202431077	560
68	1202357777	535	1202425645	546	1202427562	541	1202429309	546	1202431085	568
69	1202357781	539	1202425655	556	1202427568	547	1202429319	556	1202431093	576
70	1202357792	550	1202425660	561	1202427577	556	1202429324	561	1202431100	583
71	1202357797	555	1202425669	570	1202427582	561	1202429335	572	1202431108	591
72	1202357807	565	1202425676	577	1202427596	575	1202429340	577	1202431116	599
73	1202357812	570	1202425689	590	1202427598	577	1202429350	587	1202431124	607
74	1202357824	582	1202425691	592	1202427617	596	1202429354	591	1202431124	613
		585		605		596		602		
75	1202357827		1202425704		1202427617		1202429365		1202431138	621
76	1202357839	597	1202425711	612	1202427633	612	1202429370	607	1202431151	634
77	1202357843	601	1202425726	627	1202427635	614	1202429381	618	1202431153	636
78	1202357853	611	1202425726	627	1202427648	627	1202429385	622	1202431166	649
79	1202357858	616	1202425740	641	1202427649	628	1202429395	632	1202431175	658
80	1202357870	628	1202425742	643	1202427665	644	1202429401	638	1202431182	665
81	1202357873	631	1202425755	656	1202427668	647	1202429411	648	1202431190	673
82	1202357884	642	1202425756	657	1202427683	662	1202429419	656	1202431197	680
83	1202357888	646	1202425771	672	1202427685	664	1202429427	664	1202431216	699
84	1202357899	657	1202425772	673	1202427698	677	1202429436	673	1202431223	706
85	1202357903	661	1202425785	686	1202427700	679	1202429443	680	1202431232	715
86	1202357919	677	1202425788	689	1202427714	693	1202429451	688	1202431241	724
87	1202357919	677	1202425801	702	1202427715	694	1202429457	694	1202431248	731
88	1202357935	693	1202425802	703	1202427729	708	1202429467	704	1202431261	744
89	1202357938	696	1202425817	718	1202427730	709	1202429473	710	1202431263	746
		1	1				1	i .	1	1

La		l		l ==.	l	l	l	l ===	l	l =
91	1202357954	712	1202425833	734	1202427745	724	1202429492	729	1202431278	761
92	1202357966	724	1202425834	735	1202427760	739	1202429501	738	1202431291	774
93	1202357970	728	1202425846	747	1202427760	739	1202429508	745	1202431294	777
94	1202357982	740	1202425853	754	1202427776	755	1202429517	754	1202431306	789
95	1202357990	748	1202425867	768	1202427780	759	1202429523	760	1202431308	791
96	1202357997	755	1202425869	770	1202427791	770	1202429532	769	1202431320	803
97	1202358005	763	1202425884	785	1202427796	775	1202429539	776	1202431325	808
98	1202358011	769	1202425889	790	1202427811	790	1202429553	790	1202431336	819
99	1202358021	779	1202425899	800	1202427813	792	1202429554	791	1202431344	827
100	1202358027	785	1202425905	806	1202427826	805	1202429568	805	1202431352	835
101	1202358035	793	1202425914	815	1202427828	807	1202429569	806	1202431360	843
102	1202358047	805	1202425920	821	1202427841	820	1202429583	820	1202431367	850
103	1202358051	809 821	1202425928 1202425936	829 837	1202427857	836 836	1202429590	827 834	1202431377	860
104	1202358063	824	1202425943	844	1202427837	852	1202429397	842	1202431383	878
106	1202358000	835	1202425945	857	1202427876	855	1202429603	851	1202431393	882
107	1202358077	843	1202425959	860	1202427870	866	1202429614	859	1202431399	894
107	1202358083	855	1202425971	872	1202427893	872	1202429622	872	1202431411	897
109			1202425971	875	1202427893	882	1202429633	877		909
110	1202358106 1202358113	864 871	1202425974	887	1202427903	887	1202429640	892	1202431426 1202431430	913
111	1202358113	880	1202425994	895	1202427908	899	1202429656	893	1202431430	925
112	1202358128	886	1202425))4	903	1202427923	902	1202429668	905	1202431444	927
113	1202358128	900	1202426002	916	1202427923	902	1202429668	903	1202431444	940
114	1202358150	908	1202426018	919	1202427933	923	1202429686	923	1202431459	942
115	1202358150	915	1202426030	931	1202427944	927	1202429687	924	1202431439	957
116	1202358163	921	1202426034	935	1202427948	938	1202429700	937	1202431477	960
117	1202358103	930	1202426045	946	1202427965	944	1202429700	939	1202431477	972
118	1202358179	937	1202426048	949	1202427903	953	1202429702	951	1202431493	976
119	1202358177	945	1202426065	966	1202427974	959	1202429717	954	1202431493	989
120	1202358194	952	1202426068	969	1202427900	969	1202429717	967	1202431508	991
121	1202358203	961	1202426084	985	1202427994	973	1202429733	970	1202431500	1003
122	1202358210	968	1202426085	986	1202428004	983	1202429746	983	1202431523	1006
123	1202358219	977	1202426101	1002	1202428011	990	1202429749	986	1202431539	1022
124	1202358230	988	1202426103	1004	1202428020	999	1202429763	1000	1202431555	1038
125	1202358234	992	1202426117	1018	1202428027	1006	1202429767	1004	1202431568	1051
126	1202358245	1003	1202426119	1020	1202428036	1015	1202429781	1018	1202431585	1068
127	1202358248	1006	1202426133	1034	1202428051	1030	1202429783	1020	1202431595	1078
128	1202358265	1023	1202426139	1040	1202428071	1050	1202429797	1034	1202431595	1078
129	1202358268	1026	1202426148	1049	1202428087	1066	1202429799	1036	1202431595	1078
130	1202358282	1040	1202426155	1056	1202428102	1081	1202429812	1049	1202431604	1087
131	1202358284	1042	1202426167	1068	1202428122	1101	1202429814	1051	1202431620	1103
132	1202358296	1054	1202426175	1076	1202428137	1116	1202429827	1064	1202431625	1108
133	1202358300	1058	1202426183	1084	1202428153	1132	1202429828	1065	1202431635	1118
134	1202358311	1069	1202426190	1091	1202428162	1141	1202429844	1081	1202431640	1123
135	1202358313	1071	1202426199	1100	1202428167	1146	1202429845	1082	1202431655	1138
136	1202358327	1085	1202426206	1107	1202428183	1162	1202429857	1094	1202431662	1145
137	1202358330	1088	1202426214	1115	1202428198	1177	1202429860	1097	1202431671	1154
138	1202358342	1100	1202426226	1127	1202428213	1192	1202429874	1111	1202431680	1163
139	1202358345	1103	1202426232	1133	1202428229	1208	1202429875	1112	1202431687	1170
140	1202358357	1115	1202426242	1143	1202428235	1214	1202429888	1125	1202431702	1185
141	1202358360	1118	1202426249	1150	1202428235	1214	1202429891	1128	1202431706	1189
142	1202358372	1130	1202426258	1159	1202428244	1223	1202429904	1141	1202431715	1198
143	1202358375	1133	1202426265	1166	1202428260	1239	1202429906	1143	1202431722	1205
144	1202358389	1147	1202426272	1173	1202428275	1254	1202429922	1159	1202431737	1220
145	1202358391	1149	1202426281	1182	1202428290	1269	1202429924	1161	1202431742	1225
146	1202358403	1161	1202426289	1190	1202428305	1284	1202429937	1174	1202431752	1235
147	1202358411	1169	1202426303	1204	1202428316	1295	1202429939	1176	1202431757	1240
148	1202358418	1176	1202426311	1212	1202428321	1300	1202429952	1189	1202431772	1255
149	1202358426	1184	1202426318	1219	1202428335	1314	1202429959	1196	1202431773	1256
		l .		l .		l .		l		

150	1202358434	1192	1202426326	1227	1202428346	1325	1202429971	1208	1202431788	1271
151	1202358442	1200	1202426333	1234	1202428352	1331	1202429975	1212	1202431793	1276
152	1202358449	1207	1202426339	1240	1202428362	1341	1202429987	1224	1202431808	1291
153	1202358461	1219	1202426349	1250	1202428367	1346	1202429991	1228	1202431808	1291
154	1202358469	1227	1202426357	1258	1202428381	1360	1202430004	1241	1202431824	1307
155	1202358481	1239	1202426364	1265	1202428387	1366	1202430005	1242	1202431825	1308
156	1202358485	1243	1202426371	1272	1202428397	1376	1202430018	1255	1202431837	1320
157	1202358497	1255	1202426380	1281	1202428397	1376	1202430022	1259	1202431839	1322
158	1202358500	1258	1202426436	1337	1202428407	1386	1202430033	1270	1202431854	1337
159	1202358562	1320	1202426445	1346	1202428462	1441	1202430086	1323	1202431904	1387
160	1202358564	1322	1202426740	1641	1202428473	1452	1202430098	1335	1202431919	1402

BOING	C Mandelbro	ot 4 Mach	ines							
Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A
1	1202350032	0	1202353436	0	1202354370	0	1202355358	0	1202356396	0
2	1202350036	4	1202353440	4	1202354373	3	1202355359	1	1202356398	2
3	1202350038	6	1202353443	7	1202354373	3	1202355362	4	1202356401	5
4	1202350050	18	1202353447	11	1202354374	4	1202355365	7	1202356402	6
5	1202350052	20	1202353451	15	1202354388	18	1202355373	15	1202356410	14
6	1202350053	21	1202353453	17	1202354388	18	1202355376	18	1202356414	18
7	1202350067	35	1202353459	23	1202354389	19	1202355377	19	1202356415	19
8	1202350068	36	1202353462	26	1202354390	20	1202355381	23	1202356417	21
9	1202350072	40	1202353466	30	1202354401	31	1202355388	30	1202356428	32
10	1202350083	51	1202353470	34	1202354403	33	1202355391	33	1202356431	35
11	1202350083	51	1202353477	41	1202354404	34	1202355392	34	1202356433	37
12	1202350087	55	1202353486	50	1202354407	37	1202355397	39	1202356434	38
13	1202350087	55	1202353487	51	1202354418	48	1202355377	46	1202356445	49
14	1202350087	67	1202353487	52	1202354419	49	1202355404	48	1202356445	49
15	1202350099	67	1202353488	57	1202354421	51	1202355407	49	1202356447	51
16	1202350099	82	1202353500	64	1202354421	59	1202355411	53	1202356452	56
17	1202350114	84	1202353500	68	1202354429	61	1202335411	61	1202356462	66
18	1202350119	87	1202353504	70	1202354433	63	1202355420	62	1202356462	66
19	1202350119	96		71		64		62		67
20	1202350128	101	1202353507	84	1202354434	75	1202355420	71	1202356463	75
							1202335429			
21	1202350134	102	1202353522	86	1202354449	79		73	1202356476	80
22	1202350144	112	1202353523		1202354449	79	1202355434	76	1202356478	82
23	1202350149	117	1202353524	88	1202354450	80	1202355448	90	1202356480	84
24	1202350151	119	1202353537	101	1202354460	90	1202355449	91	1202356490	94
25	1202350159	127	1202353538	102	1202354462	92	1202355451	93	1202356491	95
26	1202350165	133	1202353539	103	1202354464	94	1202355456	98	1202356493	97
27	1202350168	136	1202353539	103	1202354471	101	1202355465	107	1202356495	99
28	1202350175	143	1202353552	116	1202354479	109	1202355468	110	1202356506	110
29	1202350184	152	1202353558	122	1202354479	109	1202355471	113	1202356508	112
30	1202350188	156	1202353558	122	1202354483	113	1202355471	113	1202356511	115
31	1202350190	158	1202353559	123	1202354487	117	1202355480	122	1202356515	119
32	1202350200	168	1202353574	138	1202354493	123	1202355486	128	1202356521	125
33	1202350203	171	1202353574	138	1202354494	124	1202355488	130	1202356524	128
34	1202350206	174	1202353574	138	1202354499	129	1202355488	130	1202356528	132
35	1202350218	186	1202353587	151	1202354501	131	1202355495	137	1202356535	139
36	1202350219	187	1202353588	152	1202354509	139	1202355502	144	1202356537	141
37	1202350220	188	1202353590	154	1202354510	140	1202355503	145	1202356539	143
38	1202350233	201	1202353591	155	1202354516	146	1202355504	146	1202356542	146
39	1202350236	204	1202353601	165	1202354519	149	1202355511	153	1202356550	154
40	1202350236	204	1202353605	169	1202354524	154	1202355516	158	1202356552	156
41	1202350250	218	1202353606	170	1202354525	155	1202355518	160	1202356554	158
42	1202350250	218	1202353607	171	1202354535	165	1202355525	167	1202356562	166
43	1202350256	224	1202353617	181	1202354537	167	1202355532	174	1202356567	171
44	1202350266	234	1202353620	184	1202354540	170	1202355534	176	1202356569	173
45	1202350268	236	1202353624	188	1202354544	174	1202355534	176	1202356570	174

46	1202350276	244	1202353633	197	1202354550	180	1202355541	183	1202356579	183
47	1202350270	249	1202353635	199	1202354553	183	1202355541	190	1202356582	186
48	1202350281	256	1202353636	200	1202354555	185	1202355549	191	1202356586	190
49	1202350292	260	1202353639	203	1202354565	195	1202355553	195	1202356586	190
50	1202350297	265	1202353653	217	1202354566	196	1202355557	199	1202356594	198
51	1202350304	272	1202353655	219	1202354568	198	1202355563	205	1202356598	202
52	1202350312	280	1202353655	219	1202354571	201	1202355565	207	1202356600	204
53	1202350313	281	1202353657	221	1202354581	211	1202355568	210	1202356602	206
54	1202350319	287	1202353668	232	1202354582	212	1202355571	213	1202356609	213
55	1202350327	295	1202353669	233	1202354585	215	1202355577	219	1202356613	217
56	1202350332	300	1202353671	235	1202354586	216	1202355580	222	1202356615	219
57	1202350335	303	1202353672	236	1202354597	227	1202355586	228	1202356617	221
58	1202350342	310	1202353686	250	1202354598	228	1202355588	230	1202356628	232
59	1202350348	316	1202353689	253	1202354602	232	1202355594	236	1202356628	232
60	1202350350	318	1202353690	254	1202354605	235	1202355595	237	1202356631	235
61	1202350358	326	1202353691	255	1202354611	241	1202355604	246	1202356645	249
62	1202350363	331	1202353701	265	1202354613	243	1202355606	248	1202356647	251
63	1202350365	333	1202353704	268	1202354618	248	1202355609	251	1202356647	251
64	1202350377	345	1202353705	269	1202354621	251	1202355611	253	1202356647	251
65	1202350378	346	1202353706	270	1202354627	257	1202355622	264	1202356660	264
66	1202350380	348	1202353719	283	1202354633	263	1202355624	266	1202356662	266
67	1202350391	359	1202353722	286	1202354636	266	1202355624	266	1202356662	266
68	1202350393	361	1202353722	286	1202354642	272	1202355626	268	1202356664	268
69	1202350394	362	1202353724	288	1202354651	281	1202355638	280	1202356675	279
70	1202350401	369	1202353736	300	1202354651	281	1202355640	282	1202356679	283
71	1202350408	376	1202353737	301	1202354653	283	1202355640	282	1202356682	286
72	1202350409	377	1202353739	303	1202354662	292	1202355642	284	1202356689	293
73	1202350411	379	1202353740	304	1202354667	297	1202355652	294	1202356691	295
74	1202350417	385	1202353740	317	1202354669	299	1202355655	297	1202356694	298
75	1202350417	392	1202353753	317	1202354670	300	1202355655	297	1202356698	302
	1202350424	396	1202353755	319	1202354678	308	1202355657	299	1202356705	309
76	1202350428	400	1202353756		1202354678	314	1202355668	310	1202356703	311
77	1202350432	402		320		_		313		
78			1202353767	331	1202354685	315	1202355671		1202356709	313
79	1202350440	408	1202353768	332	1202354687	317	1202355675	317	1202356714	318
80	1202350443	411	1202353770	334	1202354692	322	1202355687	329	1202356720	324
81	1202350447	415	1202353771	335	1202354699	329	1202355687	329	1202356722	326
82	1202350448	416	1202353784	348	1202354703	333	1202355690	332	1202356725	329
83	1202350455	423	1202353785	349	1202354703	333	1202355700	342	1202356728	332
84	1202350459	427	1202353788	352	1202354708	338	1202355701	343	1202356735	339
85	1202350462	430	1202353801	365	1202354715	345	1202355703	345	1202356738	342
86	1202350462	430	1202353801	365	1202354718	348	1202355705	347	1202356740	344
87	1202350470	438	1202353803	367	1202354721	351	1202355716	358	1202356744	348
88	1202350474	442	1202353816	380	1202354723	353	1202355718	360	1202356754	358
89	1202350477	445	1202353816	380	1202354735	365	1202355723	365	1202356754	358
90	1202350482	450	1202353824	388	1202354735	365	1202355726	368	1202356755	359
91	1202350486	454	1202353831	395	1202354738	368	1202355730	372	1202356759	363
92	1202350489	457	1202353832	396	1202354742	372	1202355734	376	1202356768	372
93	1202350492	460	1202353843	407	1202354750	380	1202355738	380	1202356770	374
94	1202350498	466	1202353846	410	1202354750	380	1202355745	387	1202356774	378
95	1202350501	469	1202353846	410	1202354752	382	1202355749	391	1202356774	378
96	1202350505	473	1202353846	410	1202354758	388	1202355749	391	1202356783	387
97	1202350512	480	1202353847	411	1202354765	395	1202355754	396	1202356789	393
98	1202350513	481	1202353847	411	1202354768	398	1202355763	405	1202356791	395
99	1202350517	485	1202353862	426	1202354778	408	1202355766	408	1202356791	395
100	1202350520	488	1202353865	429	1202354781	411	1202355773	415	1202356798	402
101	1202350528	496	1202353876	440	1202354783	413	1202355776	418	1202356805	409
102	1202350529	497	1202353878	442	1202354794	424	1202355782	424	1202356805	409
103	1202350523	501	1202353879	443	1202354796	426	1202355794	436	1202356811	415
104	1202350535	508	1202353892	456	1202354798	428	1202355794	436	1202356814	418

I	I	l .	I	I	I	l	I	l	1	I
105	1202350543	511	1202353893	457	1202354809	439	1202355796	438	1202356820	424
106	1202350545	513	1202353896	460	1202354811	441	1202355813	455	1202356821	425
107	1202350548	516	1202353909	473	1202354817	447	1202355815	457	1202356821	425
108	1202350555	523	1202353910	474	1202354824	454	1202355829	471	1202356826	430
109	1202350559	527	1202353912	476	1202354826	456	1202355832	474	1202356829	433
110	1202350560	528	1202353927	491	1202354833	463	1202355846	488	1202356841	445
111	1202350562	530	1202353928	492	1202354839	469	1202355852	494	1202356841	445
112	1202350570	538	1202353929	493	1202354843	473	1202355860	502	1202356844	448
113	1202350574	542	1202353929	493	1202354849	479	1202355867	509	1202356851	455
114	1202350575	543	1202353929	493	1202354855	485	1202355877	519	1202356857	461
115	1202350577	545	1202353943	507	1202354857	487	1202355877	519	1202356860	464
116	1202350585	553	1202353948	512	1202354865	495	1202355877	519	1202356861	465
117	1202350590	558	1202353957	521	1202354870	500	1202355880	522	1202356866	470
118	1202350593	561	1202353960	524	1202354872	502	1202355881	523	1202356872	476
119	1202350594	562	1202353969	533	1202354879	509	1202355897	539	1202356874	478
120	1202350601	569	1202353972	536	1202354890	520	1202355900	542	1202356876	480
121	1202350605	573	1202353973	537	1202354892	522	1202355908	550	1202356882	486
122	1202350608	576	1202353984	548	1202354895	525	1202355914	556	1202356887	491
123	1202350610	578	1202353989	553	1202354906	536	1202355916	558	1202356891	495
124	1202350616	584	1202353989	553	1202354908	538	1202355927	569	1202356892	496
125	1202350619	587	1202353989	553	1202354910	540	1202355927	569	1202356898	502
126	1202350623	591	1202353989	553	1202354920	550	1202355931	573	1202356902	506
127	1202350629	597	1202353990	554	1202354923	553	1202355944	586	1202356906	510
128	1202350635	603	1202353999	563	1202354926	556	1202355944	586	1202356911	515
129	1202350636	604	1202354003	567	1202354926	566	1202355947	589	1202356913	517
130	1202350640	608	1202354004	568	1202354939	569	1202355959	601	1202356922	526
131	1202350644	612	1202354004	578	1202354939	571	1202355959	601	1202356922	526
132	1202350651	619	1202354018	582	1202354949	579	1202355961	603	1202356929	533
133	1202350657	625	1202354020	584	1202354954	584	1202355974	616	1202356932	536
134	1202350659	627	1202354021	585	1202354956	586	1202355975	617	1202356937	541
135	1202350660	628	1202354030	594	1202354957	587	1202355977	619	1202356938	542
136	1202350666	634	1202354033	597	1202354972	602	1202355990	632	1202356949	553
137	1202350674	642	1202354035	599	1202354975	605	1202355992	634	1202356953	557
138	1202350677	645	1202354036	600	1202354976	606	1202356004	646	1202356957	561
139	1202350677	645	1202354045	609	1202354987	617	1202356005	647	1202356964	568
140	1202350680	648	1202354048	612	1202354989	619	1202356012	654	1202356968	572
141	1202350691	659	1202354050	614	1202354991	621	1202356021	663	1202356973	577
142	1202350694	662	1202354052	616	1202355002	632	1202356032	674	1202356984	588
143	1202350696	664	1202354065	629	1202355011	641	1202356035	677	1202356986	590
144	1202350697	665	1202354065	629	1202355011	641	1202356047	689	1202356999	603
145	1202350706	674	1202354066	630	1202355017	647	1202356048	690	1202357003	607
146	1202350711	679	1202354067	631	1202355025	655	1202356048	690	1202357011	615
147	1202350712	680	1202354079	643	1202355025	655	1202356048	690	1202357011	615
148	1202350712	680	1202354081	645	1202355025	655	1202356050	692	1202357011	615
149	1202350725	693	1202354081	645	1202355027	657	1202356066	708	1202357014	618
150	1202350726	694	1202354084	648	1202355033	663	1202356068	710	1202357023	627
151	1202350727	695	1202354096	660	1202355040	670	1202356079	721	1202357029	633
152	1202350727	695	1202354096	660	1202355042	672	1202356079	721	1202357040	644
153	1202350740	708	1202354099	663	1202355048	678	1202356081	723	1202357040	644
154	1202350742	710	1202354101	665	1202355055	685	1202356083	725	1202357044	648
155	1202350743	711	1202354112	676	1202355057	687	1202356083	725	1202357044	648
156	1202350747	715	1202354117	681	1202355112	742	1202356148	790	1202357044	648
157	1202350805	773	1202354164	728	1202355113	743	1202356148	790	1202357045	649
158	1202350807	775	1202354166	730	1202355120	750	1202356151	793	1202357104	708
159	1202350807	775	1202354177	741	1202355122	752	1202356158	800	1202357106	710
160	1202350812	780	1202354182	746	1202355179	809	1202356224	866	1202357110	714
		1		I		1	1	<u> </u>	-	I

BOINC N	Mandelbro	t 8 Machi	nes							
Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A

1	1202346359	0	1202347011	0	1202347755	0	1202348646	0	1202349304	0
2	1202346359	1	1202347011	0	1202347753	5	1202348646	7	1202349304	2
3	1202346360	2	1202347011	1	1202347760	5	1202348653	8	1202349306	4
4	1202346362	3	1202347013	2	1202347762	9	1202348654	8	1202349310	6
5	1202346365	6	1202347013	2	1202347764		1202348654	8	1202349311	7
6	1202346366	7	1202347014	3	1202347767	12	1202348656	10	1202349311	7
7	1202346366	7	1202347014	3	1202347767	12	1202348656	10	1202349312	8
8	1202346367	8	1202347015	4	1202347770	15	1202348661	15	1202349315	11
9	1202346376	17	1202347027	16	1202347774	19	1202348667	21	1202349320	16
10	1202346376	17	1202347027	16	1202347776	21	1202348668	22	1202349322	18
11	1202346377	18	1202347028	17	1202347782	27	1202348669	23	1202349324	20
12	1202346379	20	1202347029	18	1202347782	27	1202348670	24	1202349324	20
13	1202346381	22	1202347029	18	1202347783	28	1202348675	29	1202349326	22
14	1202346382	23	1202347029	18	1202347784	29	1202348676	30	1202349326	22
15	1202346386	27	1202347030	19	1202347787	32	1202348678	32	1202349331	27
16	1202346387	28	1202347030	19	1202347790	35	1202348683	37	1202349332	28
17	1202346392	33	1202347041	30	1202347792	37	1202348683	37	1202349336	32
18	1202346393	34	1202347043	32	1202347797	42	1202348684	38	1202349338	34
19	1202346394	35	1202347043	32	1202347797	42	1202348689	43	1202349341	37
20	1202346395	36	1202347045	34	1202347798	43	1202348690	44	1202349341	37
21	1202346396	37	1202347046	35	1202347802	47	1202348691	45	1202349342	38
22	1202346398	39	1202347047	36	1202347805	50	1202348693	47	1202349345	41
23	1202346402	43	1202347048	37	1202347805	50	1202348698	52	1202349347	43
24	1202346402	43	1202347051	40	1202347811	56	1202348699	53	1202349347	43
25	1202346406	47	1202347057	46	1202347812	57	1202348700	54	1202349351	47
26	1202346407	48	1202347059	48	1202347813	58	1202348706	60	1202349353	49
27	1202346410	51	1202347059	48	1202347813	59	1202348707	61	1202349355	51
28	1202346411	52	1202347059	48	1202347817	62	1202348710	64	1202349360	56
29	1202346412	53	1202347060	49	1202347820	65	1202348711	65	1202349362	58
30	1202346413	54	1202347062	51	1202347820	65	1202348713	67	1202349363	59
31	1202346415	56	1202347064	53	1202347830	75	1202348714	68	1202349363	59
32	1202346418	59	1202347066	55	1202347831	76	1202348715	69	1202349366	62
33	1202346422	63	1202347074	63	1202347832	77	1202348723	77	1202349369	65
34	1202346425	66	1202347075	64	1202347832	77	1202348726	80	1202349371	67
35	1202346426	67	1202347077	66	1202347834	79	1202348727	81	1202349373	69
36	1202346427	68	1202347079	68	1202347835	80	1202348729	83	1202349376	72
37	1202346427	68	1202347082	71	1202347840	85	1202348730	84	1202349376	72
38	1202346427	68	1202347082	71	1202347845	90	1202348730	84	1202349378	74
39	1202346432	73	1202347085	74	1202347846	91	1202348731	85	1202349378	74
40	1202346438	79	1202347090	79	1202347848	93	1202348740	94	1202349380	76
41	1202346438	79	1202347090	79	1202347848	93	1202348743	97	1202349386	82
42	1202346441	82	1202347094	83	1202347851	96	1202348744	98	1202349387	83
43	1202346441	82	1202347095	84	1202347852	97	1202348746	100	1202349389	85
44	1202346442	83	1202347096	85	1202347855	100	1202348746	100	1202349392	88
45	1202346442	83	1202347099	88	1202347864	109	1202348746	100	1202349392	88
46	1202346444	85	1202347100	89	1202347866	111	1202348747	101	1202349393	89
47	1202346448	89	1202347102	91	1202347866	111	1202348758	112	1202349397	93
48	1202346453	94	1202347102	93	1202347868	113	1202348759	113	1202349401	97
49	1202346453	94	1202347104	94	1202347808	121	1202348759	113	1202349401	98
50	1202346455	96	1202347110	99	1202347876	121	1202348760	114	1202349404	100
51	1202346457	98	1202347111	100	1202347878	123	1202348761	115	1202349407	103
52	1202346458	99	1202347112	101	1202347881	126	1202348761	115	1202349407	103
53	1202346459	100	1202347115	104	1202347883	128	1202348762	116	1202349408	104
54	1202346462	103	1202347117	106	1202347883	128	1202348774	128	1202349408	104
55	1202346463	104	1202347121	110	1202347884	129	1202348774	128	1202349413	109
56	1202346468	109	1202347124	113	1202347890	135	1202348775	129	1202349415	111
57	1202346469	110	1202347125	114	1202347892	137	1202348776	130	1202349417	113
58	1202346471	112	1202347126	115	1202347894	139	1202348776	130	1202349419	115
	1202346473	114	1202347127	116	1202347897	142	1202348777	131	1202349422	118

60	1202346474	115	1202347133	122	1202347900	145	1202348777	131	1202349423	119
60			1202347133	122	1202347900	145	1202348777	145	1202349423	119
61	1202346476	117								
62	1202346482	123	1202347137	126	1202347903	148	1202348792	146	1202349423	119
63	1202346482	123	1202347139	128	1202347908	153	1202348794	148	1202349428	124
64	1202346483	124	1202347140	129	1202347910	155	1202348795	149	1202349431	127
65	1202346484	125	1202347147	136	1202347913	158	1202348796	150	1202349434	130
66	1202346489	130	1202347152	141	1202347913	158	1202348798	152	1202349434	130
67	1202346490	131	1202347153	142	1202347915	160	1202348798	152	1202349438	134
68	1202346492	133	1202347155	144	1202347916	161	1202348806	160	1202349438	134
69	1202346493	134	1202347155	144	1202347921	166	1202348807	161	1202349439	135
70	1202346498	139	1202347156	145	1202347924	169	1202348809	163	1202349442	138
71	1202346499	140	1202347156	145	1202347925	170	1202348810	164	1202349443	139
72	1202346500	141	1202347162	151	1202347928	173	1202348814	168	1202349449	145
73	1202346502	143	1202347167	156	1202347929	174	1202348814	168	1202349450	146
74	1202346503	144	1202347170	159	1202347930	175	1202348816	170	1202349453	149
75	1202346504	145	1202347171	160	1202347933	178	1202348821	175	1202349454	150
76	1202346507	148	1202347172	161	1202347937	182	1202348821	175	1202349455	151
77	1202346508	149	1202347172	161	1202347939	184	1202348825	179	1202349457	153
78	1202346513	154	1202347176	165	1202347942	187	1202348827	181	1202349462	158
79	1202346514	155	1202347170	168	1202347942	188	1202348828	182	1202349463	159
80	1202346515	156	1202347179	173	1202347943	188	1202348828	182	1202349463	160
			1202347184						1202349467	
81	1202346520	161		175	1202347946	191	1202348831	185		163
82	1202346521	162	1202347187	176	1202347952	197	1202348837	191	1202349470	166
83	1202346522	163	1202347187	176	1202347953	198	1202348838	192	1202349470	166
84	1202346522	163	1202347189	178	1202347955	200	1202348840	194	1202349473	169
85	1202346528	169	1202347192	181	1202347957	202	1202348841	195	1202349479	175
86	1202346528	169	1202347194	183	1202347958	203	1202348843	197	1202349479	175
87	1202346528	169	1202347201	190	1202347963	208	1202348844	198	1202349479	175
88	1202346529	170	1202347202	191	1202347965	210	1202348847	201	1202349483	179
89	1202346535	176	1202347203	192	1202347967	212	1202348852	206	1202349485	181
90	1202346537	178	1202347204	193	1202347967	212	1202348853	207	1202349486	182
91	1202346539	180	1202347207	196	1202347969	214	1202348855	209	1202349488	184
92	1202346544	185	1202347209	198	1202347972	217	1202348856	210	1202349490	186
93	1202346544	185	1202347215	204	1202347974	219	1202348860	214	1202349493	189
94	1202346545	186	1202347217	206	1202347980	225	1202348860	214	1202349494	190
95	1202346548	189	1202347218	207	1202347981	226	1202348862	216	1202349495	191
96	1202346549	190	1202347219	208	1202347983	228	1202348866	220	1202349500	196
97	1202346555	196	1202347222	211	1202347984	229	1202348867	221	1202349500	196
98	1202346557	198	1202347222	211	1202347984	229	1202348871	225	1202349503	199
99	1202346558	199	1202347225	214	1202347989	234	1202348872	226	1202349504	200
100	1202346559	200	1202347232	221	1202347994	239	1202348873	227	1202349505	201
101	1202346560	201	1202347232	221	1202347998	243	1202348875	229	1202349509	205
102	1202346560	201	1202347233	222	1202347999	244	1202348883	237	1202349510	206
103	1202346563	204	1202347234	223	1202348002	247	1202348883	237	1202349510	206
104	1202346568	209	1202347237	226	1202348005	250	1202348886	240	1202349518	214
105	1202346571	212	1202347238	227	1202348005	250	1202348887	241	1202349519	215
106	1202346574	215	1202347239	228	1202348012	257	1202348889	243	1202349520	216
107	1202346574	215	1202347246	235	1202348014	259	1202348889	243	1202349521	217
108	1202346574	215	1202347247	236	1202348014	259	1202348892	246	1202349524	220
109	1202346575	216	1202347248	237	1202348018	263	1202348897	251	1202349525	221
110	1202346577	218	1202347249	238	1202348020	265	1202348898	252	1202349525	221
111	1202346579	220	1202347252	241	1202348024	269	1202348902	256	1202349525	221
112	1202346585	226	1202347253	242	1202348026	271	1202348903	257	1202349534	230
113	1202346586	227	1202347255	244	1202348028	273	1202348904	258	1202349537	233
114	1202346590	231	1202347262	251	1202348029	274	1202348905	259	1202349537	233
115	1202346591	232	1202347262	251	1202348036	281	1202348908	262	1202349538	234
116	1202346591	232	1202347267	256	1202348038	283	1202348912	266	1202349539	235
117	1202346592	233	1202347268	257	1202348039	284	1202348913	267	1202349540	236

119	1202346594	235	1202347269	258	1202348045	290	1202348919	273	1202349540	236
120	1202346599	240	1202347272	261	1202348049	294	1202348920	274	1202349551	247
121	1202346603	244	1202347277	266	1202348050	295	1202348921	275	1202349551	247
122	1202346606	247	1202347282	271	1202348053	298	1202348927	281	1202349554	250
123	1202346607	248	1202347284	273	1202348055	300	1202348928	282	1202349554	250
124	1202346608	249	1202347284	273	1202348056	301	1202348931	285	1202349554	250
125	1202346609	250	1202347286	275	1202348059	304	1202348933	287	1202349555	251
126	1202346609	250	1202347286	275	1202348065	310	1202348935	289	1202349556	252
127	1202346612	253	1202347288	277	1202348070	315	1202348936	290	1202349556	252
128	1202346614	255	1202347297	286	1202348071	316	1202348938	292	1202349566	262
129	1202346617	258	1202347298	287	1202348075	320	1202348943	297	1202349570	266
130	1202346620	261	1202347299	288	1202348075	320	1202348948	302	1202349571	267
131	1202346622	263	1202347300	289	1202348075	320	1202348948	302	1202349571	267
132	1202346624	265	1202347303	292	1202348080	325	1202348950	304	1202349571	267
133	1202346625	266	1202347305	294	1202348085	330	1202348951	305	1202349572	268
134	1202346629	270	1202347309	298	1202348086	331	1202348953	307	1202349575	271
135	1202346629	270	1202347312	301	1202348091	336	1202348954	308	1202349575	271
136	1202346631	272	1202347314	303	1202348091	336	1202348959	313	1202349582	278
137	1202346633	274	1202347315	304	1202348094	339	1202348962	316	1202349586	282
138	1202346636	277	1202347315	304	1202348094	339	1202348964	318	1202349587	283
139	1202346636	277	1202347321	310	1202348100	345	1202348965	319	1202349587	283
140	1202346644	285	1202347322	311	1202348105	350	1202348968	322	1202349590	286
141	1202346645	286	1202347325	314	1202348106	351	1202348970	324	1202349590	286
142	1202346645	286	1202347328	317	1202348108	353	1202348974	328	1202349590	286
143	1202346645	286	1202347331	320	1202348111	356	1202348975	329	1202349591	287
144	1202346645	286	1202347334	323	1202348116	361	1202348978	332	1202349598	294
145	1202346652	293	1202347335	324	1202348119	364	1202348979	333	1202349601	297
146	1202346653	294	1202347336	325	1202348121	366	1202348981	335	1202349601	297
147	1202346656	297	1202347338	327	1202348121	366	1202348983	337	1202349602	298
148	1202346658	299	1202347339	328	1202348122	367	1202348988	342	1202349605	301
149	1202346660	301	1202347396	385	1202348168	413	1202348989	343	1202349606	302
150	1202346661	302	1202347398	387	1202348174	419	1202348993	347	1202349606	302
151	1202346665	306	1202347400	389	1202348175	420	1202348994	348	1202349610	306
152	1202346665	306	1202347403	392	1202348180	425	1202348995	349	1202349662	358
153	1202346717	358	1202347403	392	1202348184	429	1202349046	400	1202349665	361
154	1202346718	359	1202347404	393	1202348185	430	1202349048	402	1202349666	362
155	1202346722	363	1202347405	394	1202348187	432	1202349052	406	1202349667	363
156	1202346723	364	1202347467	456	1202348190	435	1202349058	412	1202349670	366
157	1202346725	366	1202347476	465	1202348238	483	1202349058	412	1202349671	367
158	1202346725	366	1202347476	465	1202348261	506	1202349059	413	1202349671	367
159	1202346730	371	1202347534	523	1202348372	617	1202349059	413	1202349675	371
160	1202346730	371	1202347598	587	1202348461	706	1202349065	419	1202349870	566

BOINC	Mandelbro	t 16 Mach	nines							
Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A
1	1202341941	0	1202343896	0	1202344568	0	1202345300	0	1202345868	0
2	1202341941	0	1202343896	0	1202344568	0	1202345302	2	1202345870	2
3	1202341942	1	1202343897	1	1202344568	0	1202345302	2	1202345870	2
4	1202341942	1	1202343898	2	1202344569	1	1202345303	3	1202345872	4
5	1202341942	1	1202343899	3	1202344571	3	1202345304	4	1202345872	4
6	1202341942	1	1202343900	4	1202344572	4	1202345305	5	1202345874	6
7	1202341945	4	1202343900	4	1202344573	5	1202345306	6	1202345874	6
8	1202341945	4	1202343901	5	1202344573	5	1202345307	7	1202345875	7
9	1202341945	4	1202343901	5	1202344573	5	1202345308	8	1202345875	7
10	1202341945	4	1202343901	5	1202344574	6	1202345308	8	1202345876	8
11	1202341945	4	1202343902	6	1202344574	6	1202345310	10	1202345876	8
12	1202341946	5	1202343902	6	1202344575	7	1202345310	10	1202345877	9
13	1202341946	5	1202343902	6	1202344575	7	1202345312	12	1202345877	9

14	1202341946	5	1202343903	7	1202344576	8	1202345312	12	1202345877	9
15	1202341946	9	1202343903	8	1202344577	9	1202345312	14	1202345877	10
16	1202341955	14	1202343904	8	1202344577	9	1202345314	16	1202345879	11
17	1202341956	15	1202343911	15	1202344577	14	1202345310	21	1202345885	17
18	1202341936	15	1202343911	15	1202344584	16	1202345321	21	1202345887	19
19		16		16	1202344584	16		22		
	1202341957		1202343912			19	1202345322		1202345889	21
20	1202341957	16	1202343912	16	1202344587		1202345324	24	1202345890	
21	1202341960	19	1202343916	20	1202344587	19	1202345324	24	1202345891	23
22	1202341961	20	1202343916	20	1202344588	20	1202345324	24	1202345891	23
23	1202341961	20	1202343917	21	1202344589	21	1202345325	25	1202345891	23
24	1202341961	20	1202343917	21	1202344589	21	1202345325	25	1202345892	24
25	1202341962	21	1202343918	22	1202344589	21	1202345326	26	1202345892	24
26	1202341962	21	1202343919	23	1202344590	22	1202345327	27	1202345893	25
27	1202341962	21	1202343920	24	1202344590	22	1202345328	28	1202345893	25
28	1202341964	23	1202343920	24	1202344591	23	1202345331	31	1202345894	26
29	1202341966	25	1202343921	25	1202344592	24	1202345331	31	1202345894	26
30	1202341966	25	1202343921	25	1202344592	24	1202345331	31	1202345895	27
31	1202341970	29	1202343922	26	1202344593	25	1202345333	33	1202345895	27
32	1202341976	35	1202343922	26	1202344597	29	1202345338	38	1202345898	30
33	1202341976	35	1202343926	30	1202344598	30	1202345338	38	1202345899	31
34	1202341977	36	1202343929	33	1202344599	31	1202345338	38	1202345903	35
35	1202341977	36	1202343930	34	1202344599	31	1202345340	40	1202345905	37
36	1202341977	36	1202343931	35	1202344603	35	1202345341	41	1202345906	38
37	1202341977	36	1202343931	35	1202344603	35	1202345341	41	1202345907	39
38	1202341977	36	1202343932	36	1202344604	36	1202345341	41	1202345908	40
39	1202341977	36	1202343933	37	1202344604	36	1202345342	42	1202345908	40
40	1202341977	36	1202343934	38	1202344607	39	1202345343	43	1202345908	40
41	1202341977	36	1202343935	39	1202344607	39	1202345344	44	1202345908	40
42	1202341978	37	1202343936	40	1202344607	39	1202345346	46	1202345909	41
43	1202341981	40	1202343936	40	1202344608	40	1202345347	47	1202345910	42
44	1202341981	40	1202343936	40	1202344609	41	1202345349	49	1202345910	42
45	1202341982	41	1202343937	41	1202344609	41	1202345349	49	1202345910	42
46	1202341986	45	1202343937	41	1202344610	42	1202345349	49	1202345911	43
47	1202341990	49	1202343940	44	1202344611	43	1202345352	52	1202345912	44
48	1202341991	50	1202343941	45	1202344611	43	1202345353	53	1202345915	47
49	1202341991	50	1202343942	46	1202344617	49	1202345356	56	1202345915	47
50	1202341991	50	1202343944	48	1202344618	50	1202345357	57	1202345920	52
51	1202341992	51	1202343946	50	1202344618	50	1202345357	57	1202345922	54
52	1202341992	51	1202343946 1202343947	50	1202344619	51	1202345357	57	1202345923	55
53	1202341992	51		51		51	1202345358	58	1202345923	55
54	1202341992	51	1202343949	53	1202344622	54	1202345360	60	1202345923	55
55	1202341993	52	1202343950	54	1202344623	55	1202345361	61	1202345923	55
56	1202341994	53	1202343951	55	1202344623	55	1202345361	61	1202345924	56
57	1202341996	55	1202343951	55	1202344623	55	1202345362	62	1202345925	57
58	1202341997	56	1202343952	56	1202344624	56	1202345364	64	1202345925	57
59	1202341997	56	1202343952	56	1202344624	56	1202345364	64	1202345925	57
60	1202341998	57	1202343952	56	1202344625	57	1202345364	64	1202345925	57
61	1202342002	61	1202343953	57	1202344626	58	1202345367	67	1202345925	57
62	1202342006	65	1202343956	60	1202344626	58	1202345368	68	1202345927	59
63	1202342007	66	1202343957	61	1202344627	59	1202345369	69	1202345927	59
64	1202342007	66	1202343959	63	1202344630	62	1202345371	71	1202345931	63
65	1202342007	66	1202343959	63	1202344633	65	1202345372	72	1202345934	66
66	1202342007	66	1202343961	65	1202344634	66	1202345372	72	1202345935	67
67	1202342007	66	1202343961	65	1202344634	66	1202345372	72	1202345937	69
68	1202342008	67	1202343962	66	1202344635	67	1202345373	73	1202345937	69
69	1202342008	67	1202343963	67	1202344636	68	1202345375	75	1202345939	71
70	1202342008	67	1202343963	67	1202344637	69	1202345376	76	1202345939	71
71	1202342011	70	1202343966	70	1202344638	70	1202345378	78	1202345939	71
	1	1	1	1	1					

	I	I	l	1	I	1	I	1	1	I
73	1202342016	75	1202343966	70	1202344640	72	1202345380	80	1202345940	72
74	1202342017	76	1202343967	71	1202344642	74	1202345381	81	1202345940	72
75	1202342017	76	1202343969	73	1202344643	75	1202345382	82	1202345941	73
76	1202342022	81	1202343971	75	1202344643	75	1202345383	83	1202345942	74
77	1202342022	81	1202343972	76	1202344643	75	1202345384	84	1202345943	75
78	1202342022	81	1202343972	76	1202344644	76	1202345384	84	1202345946	78
79	1202342022	81	1202343973	77	1202344646	78	1202345387	87	1202345947	79
80	1202342023	82	1202343975	79	1202344646	78	1202345388	88	1202345949	81
81	1202342023	82	1202343976	80	1202344648	80	1202345389	89	1202345949	81
82	1202342026	85	1202343977	81	1202344648	80	1202345390	90	1202345949	81
83	1202342027	86	1202343978	82	1202344649	81	1202345390	90	1202345953	85
84	1202342027	86	1202343978	82	1202344650	82	1202345392	92	1202345953	85
85	1202342027	86	1202343979	83	1202344651	83	1202345392	92	1202345954	86
86	1202342028	87	1202343981	85	1202344653	85	1202345393	93	1202345956	88
87	1202342028	87	1202343981	85	1202344654	86	1202345394	94	1202345957	89
88	1202342028	91	1202343982	86	1202344655	87	1202345395	95	1202345957	89
89	1202342036	95	1202343982	86	1202344656	88	1202345398	98	1202345958	90
90	1202342036	95	1202343984	88	1202344658	90	1202345399	99	1202345958	90
91	1202342037	96	1202343986	90	1202344658	90	1202345399	99	1202345959	91
92	1202342037	96	1202343986	90	1202344659	91	1202345400	100	1202345959	91
93	1202342037	96	1202343987	91	1202344659	91	1202345401	101	1202345960	92
94	1202342038	97	1202343987	91	1202344661	93	1202345402	102	1202345962	94
95	1202342038	97	1202343988	92	1202344661	93	1202345403	103	1202345962	94
96	1202342038	97	1202343991	95	1202344662	94	1202345405	105	1202345965	97
97	1202342041	100	1202343992	96	1202344663	95	1202345406	106	1202345965	97
98	1202342042	101	1202343992	96	1202344664	96	1202345407	107	1202345966	98
99	1202342043	102	1202343994	98	1202344665	97	1202345407	107	1202345967	99
100	1202342044	103	1202343997	101	1202344666	98	1202345408	108	1202345968	100
101	1202342044	103	1202343997	101	1202344668	100	1202345408	108	1202345971	103
102	1202342047	106	1202343997	101	1202344668	100	1202345409	109	1202345971	103
103	1202342049	108	1202343998	102	1202344669	101	1202345410	110	1202345973	105
104	1202342051	110	1202343998	102	1202344670	102	1202345413	113	1202345973	105
105	1202342052	111	1202343999	103	1202344673	105	1202345415	115	1202345975	107
106	1202342052	111	1202344000	104	1202344674	106	1202345415	115	1202345975	107
107	1202342052	111	1202344002	106	1202344674	106	1202345418	118	1202345975	107
108	1202342052	112	1202344002	106	1202344674	106	1202345418	118	1202345975	107
									1202345977	
109	1202342053	112	1202344003	107	1202344675	107	1202345419	119		109
110	1202342053	112	1202344003	107	1202344676	108	1202345419	119	1202345978	110
111	1202342054	113	1202344005	109	1202344676	108	1202345421	121	1202345980	112
112	1202342058	117	1202344006	110	1202344678	110	1202345422	122	1202345980	112
113	1202342058	117	1202344008	112	1202344679	111	1202345422	122	1202345982	114
114	1202342059	118	1202344010	114	1202344679	111	1202345422	122	1202345983	115
115	1202342059	118	1202344010	114	1202344680	112	1202345423	123	1202345984	116
116	1202342063	122	1202344011	115	1202344681	113	1202345424	124	1202345985	117
117	1202342063	122	1202344012	116	1202344682	114	1202345424	124	1202345986	118
118	1202342067	126	1202344013	117	1202344683	115	1202345425	125	1202345988	120
119	1202342068	127	1202344014	118	1202344685	117	1202345429	129	1202345989	121
120	1202342068	127	1202344014	118	1202344686	118	1202345430	130	1202345990	122
121	1202342068	127	1202344014	118	1202344687	119	1202345431	131	1202345990	122
122	1202342068	127	1202344017	121	1202344688	120	1202345432	132	1202345990	122
123	1202342072	131	1202344017	121	1202344690	122	1202345434	134	1202345990	122
124	1202342072	131	1202344017	121	1202344691	123	1202345434	134	1202345991	123
125	1202342073	132	1202344019	123	1202344691	123	1202345437	137	1202345992	124
126	1202342074	133	1202344019	123	1202344691	123	1202345437	137	1202345996	128
127	1202342074	133	1202344019	126	1202344692	124	1202345437	137	1202345996	128
			1202344022		1202344692					
128	1202342075	134		126		126	1202345437	137	1202345996	128
129	1202342077	136	1202344023	127	1202344695	127	1202345438	138	1202345998	130
130	1202342079	138	1202344024	128	1202344695	127	1202345438	138	1202345998	130
131	1202342082	141	1202344026	130	1202344697	129	1202345440	140	1202345999	131

132	1202342083	142	1202344028	132	1202344697	129	1202345441	141	1202346001	133
133	1202342085	144	1202344028	132	1202344698	130	1202345441	141	1202346002	134
134	1202342087	146	1202344028	132	1202344700	132	1202345444	144	1202346003	135
135	1202342087	146	1202344028	132	1202344702	134	1202345445	145	1202346005	137
136	1202342087	146	1202344029	133	1202344702	134	1202345446	146	1202346006	138
137	1202342088	147	1202344030	134	1202344704	136	1202345449	149	1202346006	138
138	1202342088	147	1202344032	136	1202344704	136	1202345450	150	1202346006	138
139	1202342088	147	1202344033	137	1202344705	137	1202345452	152	1202346006	138
140	1202342088	147	1202344033	137	1202344705	137	1202345453	153	1202346007	139
141	1202342089	148	1202344037	141	1202344705	137	1202345453	153	1202346007	139
142	1202342139	198	1202344037	141	1202344707	139	1202345453	153	1202346010	142
143	1202342142	201	1202344037	141	1202344713	145	1202345453	153	1202346013	145
144	1202342144	203	1202344038	142	1202344759	191	1202345453	153	1202346014	146
145	1202342147	206	1202344088	192	1202344759	191	1202345503	203	1202346063	195
146	1202342148	207	1202344089	193	1202344761	193	1202345505	205	1202346063	195
147	1202342149	208	1202344091	195	1202344762	194	1202345506	206	1202346064	196
148	1202342152	211	1202344092	196	1202344763	195	1202345506	206	1202346066	198
149	1202342152	211	1202344092	196	1202344765	197	1202345509	209	1202346066	198
150	1202342152	211	1202344092	196	1202344768	200	1202345509	209	1202346068	200
151	1202342153	212	1202344093	197	1202344768	200	1202345511	211	1202346069	201
152	1202342153	212	1202344095	199	1202344768	200	1202345514	214	1202346070	202
153	1202342153	212	1202344095	199	1202344769	201	1202345515	215	1202346070	202
154	1202342153	212	1202344097	201	1202344770	202	1202345517	217	1202346070	202
155	1202342153	212	1202344097	201	1202344770	202	1202345517	217	1202346071	203
156	1202342310	369	1202344098	202	1202344770	202	1202345518	218	1202346071	203
157	1202342310	369	1202344102	206	1202344771	203	1202345518	218	1202346074	206
158	1202342431	490	1202344102	206	1202344772	204	1202345518	218	1202346077	209
159	1202342431	490	1202344103	207	1202344779	211	1202345518	218	1202346080	212
160	1202342488	547	1202344103	207	1202345161	593	1202345715	415	1202346080	212

CompTorrent

Count	Corrent Man	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A
1	1206268832	0	1206269386	0	1206269966	0	1206270653	0	1206271155	0
2	1206268831	1	1206269384	1	1206269965	2	1206270652	1	1206271154	1
3	1206268829	3	1206269383	3	1206269964	3	1206270650	3	1206271152	3
4	1206268828	4	1206269381	4	1206269962	5	1206270649	4	1206271151	4
5	1206268826	6	1206269380	6	1206269960	6	1206270647	6	1206271149	6
6	1206268825	7	1206269377	7	1206269959	9	1206270646	7	1206271148	7
7	1206268823	9	1206269376	9	1206269957	10	1206270644	9	1206271146	9
8	1206268822	10	1206269375	10	1206269956	11	1206270643	10	1206271145	10
9	1206268820	12	1206269373	12	1206269954	13	1206270641	12	1206271143	12
10	1206268819	13	1206269372	13	1206269953	14	1206270640	13	1206271142	13
11	1206268817	15	1206269370	15	1206269951	16	1206270638	15	1206271140	15
12	1206268816	16	1206269369	16	1206269950	17	1206270637	16	1206271138	17
13	1206268814	18	1206269367	18	1206269948	19	1206270635	18	1206271137	18
14	1206268813	19	1206269366	19	1206269947	20	1206270634	19	1206271135	20
15	1206268811	21	1206269364	21	1206269945	22	1206270632	21	1206271134	21
16	1206268810	22	1206269362	22	1206269943	24	1206270631	22	1206271132	23
17	1206268808	24	1206269361	24	1206269942	25	1206270629	24	1206271131	24
18	1206268807	25	1206269359	25	1206269940	27	1206270628	25	1206271129	26
19	1206268805	27	1206269358	27	1206269939	28	1206270626	27	1206271128	27
20	1206268804	28	1206269356	28	1206269937	30	1206270440	213	1206271126	29
21	1206268802	30	1206269355	30	1206269936	31	1206270439	214	1206271125	30
22	1206268801	31	1206269354	31	1206269934	32	1206270437	216	1206271123	32
23	1206268799	33	1206269352	33	1206269933	34	1206270436	217	1206271121	34
24	1206268798	34	1206269351	34	1206269931	35	1206270434	219	1206271120	35

25	1206268796	26	1206260240	26	1206260020	27	1206270422	220	1206271119	27
25		36	1206269349	36	1206269930	37	1206270433	220	1206271118	37
26	1206268795	37	1206269347	37	1206269928	39	1206270431	222	1206271117	38
27	1206268793	39	1206269346	39	1206269927	40	1206270430	223	1206271115	40
28	1206268792	40	1206269344	40	1206269925	42	1206270428	225	1206271114	41
29	1206268790	42	1206269343	42	1206269924	43	1206270427	226	1206271112	43
30	1206268788	44	1206269342	44	1206269922	44	1206270425	228	1206271111	44
31	1206268787	45	1206269340	45	1206269920	46	1206270424	229	1206271109	46
32	1206268784	48	1206269337	48	1206269918	49	1206270421	232	1206271106	49
33	1206268781	51	1206269334	51	1206269915	52	1206270418	235	1206271104	51
34	1206268778	54	1206269331	54	1206269911	55	1206270415	238	1206271101	54
35	1206268774	58	1206269327	58	1206269908	59	1206270411	242	1206271097	58
36	1206268770	62	1206269323	62	1206269904	63	1206270407	246	1206271092	63
37	1206268766	66	1206269318	66	1206269900	68	1206270402	251	1206271088	67
38	1206268762	70	1206269314	70	1206269896	72	1206270398	255	1206271084	71
39	1206268758	74	1206269311	74	1206269892	75	1206270394	259	1206271080	75
40	1206268755	77	1206269307	77	1206269888	79	1206270390	263	1206271076	79
41	1206268751	81	1206269303	81	1206269883	83	1206270386	267	1206271072	83
42	1206268746	86	1206269298	86	1206269879	88	1206270381	272	1206271068	87
43	1206268742	90	1206269294	90	1206269875	92	1206270377	276	1206271063	92
44	1206268738	94		94		97	1206270377	281		96
		99	1206269289	99	1206269871			281	1206271059	101
45	1206268733		1206269285		1206269866	101	1206270368		1206271054	
46	1206268729	103	1206269280	103	1206269862	106	1206270364	289	1206271050	105
47	1206268724	108	1206269276	108	1206269857	110	1206270359	294	1206271045	110
48	1206268720	112	1206269271	112	1206269852	115	1206270354	299	1206271041	114
49	1206268716	116	1206269266	116	1206269848	120	1206270350	303	1206271036	119
50	1206268711	121	1206269262	121	1206269843	124	1206270345	308	1206271032	123
51	1206268706	126	1206269257	126	1206269838	129	1206270340	313	1206271027	128
52	1206268700	132	1206269252	132	1206269833	134	1206270335	318	1206271021	134
53	1206268695	137	1206269247	137	1206269828	139	1206270329	324	1206271016	139
54	1206268689	143	1206269241	143	1206269822	145	1206270323	330	1206271010	145
55	1206268683	149	1206269235	149	1206269816	151	1206270318	335	1206271004	151
56	1206268678	154	1206269229	154	1206269810	157	1206270312	341	1206270999	156
57	1206268672	160	1206269223	160	1206269804	163	1206270306	347	1206270993	162
58	1206268667	165	1206269218	165	1206269799	168	1206270301	352	1206270988	167
59	1206268661	171	1206269213	171	1206269794	173	1206270296	357	1206270983	172
60	1206268656	176	1206269208	176	1206269789	178	1206270290	363	1206270978	177
61	1206268652	180	1206269203	180	1206269785	183	1206270286	367	1206270974	181
62	1206268647	185	1206269199	185	1206269780	187	1206270281	372	1206270969	186
63	1206268643	189	1206269194	189	1206269775	192	1206270276	377	1206270965	190
64	1206268638	194	1206269190	194	1206269771	196	1206270270	381	1206270960	195
65	1206268634	198	1206269186	198	1206269767	200	1206270267	386	1206270955	200
66	1206268629	203	1206269181	203	1206269762	205	1206270263	390	1206270950	205
67	1206268624	208	1206269177	208	1206269758	209	1206270258	395	1206270945	210
68	1206268620	212	1206269173	212	1206269754	213	1206270254	399	1206270941	214
69	1206268616	216	1206269169	216	1206269750	217	1206270250	403	1206270937	218
70	1206268611	221	1206269165	221	1206269746	221	1206270246	407	1206270932	223
71	1206268607	225	1206269161	225	1206269742	225	1206270242	411	1206270928	227
72	1206268603	229	1206269157	229	1206269738	229	1206270238	415	1206270924	231
73	1206268598	234	1206269153	234	1206269733	233	1206270233	420	1206270920	235
74	1206268594	238	1206269149	238	1206269729	237	1206270229	424	1206270915	240
75	1206268590	242	1206269145	242	1206269724	241	1206270225	428	1206270911	244
76	1206268587	245	1206269141	245	1206269720	245	1206270221	432	1206270907	248
77	1206268583	249	1206269137	249	1206269717	249	1206270218	435	1206270903	252
78	1206268580	252	1206269134	252	1206269713	252	1206270214	439	1206270900	255
79	1206268577	255	1206269130	255	1206269710	256	1206270211	442	1206270896	259
80	1206268577	259	1206269130	259	1206269716	259	1206270211	442	1206270890	262
81	1206268571	261	1206269124	261	1206269703	262	1206270204	449	1206270890	265
82	1206268568	264	1206269121	264	1206269700	265	1206270201	452	1206270887	268
83	1206268565	267	1206269118	267	1206269697	268	1206270198	455	1206270885	270

84	1206268563	269	1206269115	269	1206269695	271	1206270196	457	1206270882	273
85	1206268563	269	1206269115	269	1206269693	271	1206270196	457	1206270882	275
				271		274				
86	1206268559	273	1206269112		1206269691		1206270192	461	1206270878	277
87	1206268558	274	1206269110	274	1206269689	276	1206270190	463	1206270876	279
88	1206268556	276	1206269108	276	1206269687	278	1206270188	465	1206270875	280
89	1206268554	278	1206269106	278	1206269685	280	1206270186	467	1206270873	282
90	1206268551	281	1206269103	281	1206269682	283	1206270184	469	1206270870	285
91	1206268549	283	1206269101	283	1206269680	285	1206270182	471	1206270868	287
92	1206268547	285	1206269099	285	1206269677	287	1206270179	474	1206270865	290
93	1206268545	287	1206269097	287	1206269675	289	1206270176	477	1206270863	292
94	1206268542	290	1206269095	290	1206269672	291	1206270173	480	1206270860	295
95	1206268540	292	1206269092	292	1206269670	294	1206270171	482	1206270858	297
96	1206268537	295	1206269090	295	1206269667	296	1206270168	485	1206270855	300
97	1206268534	298	1206269087	298	1206269664	299	1206270166	487	1206270853	302
98	1206268531	301	1206269084	301	1206269662	302	1206270163	490	1206270850	305
99	1206268529	303	1206269081	303	1206269659	305	1206270160	493	1206270848	307
100	1206268526	306	1206269079	306	1206269657	307	1206270158	495	1206270846	309
101	1206268524	308	1206269076	308	1206269654	310	1206270155	498	1206270843	312
102	1206268521	311	1206269074	311	1206269652	312	1206270153	501	1206270843	315
				313				503		317
103	1206268519	313	1206269071	316	1206269649	315	1206270150		1206270838	320
104	1206268516	316	1206269069		1206269647	317	1206270147	506	1206270835	
105	1206268514	318	1206269066	318	1206269644	320	1206270144	509	1206270833	322
106	1206268511	321	1206269064	321	1206269642	322	1206270142	511	1206270830	325
107	1206268509	323	1206269062	323	1206269640	324	1206270140	513	1206270828	327
108	1206268507	325	1206269060	325	1206269638	326	1206270138	515	1206270826	329
109	1206268505	327	1206269058	327	1206269637	328	1206270137	516	1206270824	331
110	1206268504	328	1206269056	328	1206269635	330	1206270136	517	1206270823	332
111	1206268502	330	1206269055	330	1206269633	331	1206270134	519	1206270821	334
112	1206268499	333	1206269053	333	1206269632	333	1206270133	520	1206270820	335
113	1206268498	334	1206269052	334	1206269630	334	1206270131	522	1206270818	337
114	1206268496	336	1206269050	336	1206269629	336	1206270129	524	1206270816	339
115	1206268495	337	1206269049	337	1206269627	337	1206270128	525	1206270815	340
116	1206268493	339	1206269047	339	1206269626	339	1206270126	527	1206270813	342
117	1206268492	340	1206269046	340	1206269624	340	1206270125	528	1206270812	343
118	1206268490	342	1206269044	342	1206269623	342	1206270123	530	1206270810	345
119	1206268489	343	1206269043	343	1206269621	343	1206270122	531	1206270809	346
120	1206268487	345	1206269041	345	1206269620	345	1206270122	533	1206270807	348
						347		534		349
121	1206268486	346	1206269039	346	1206269618		1206270119		1206270806	
122	1206268484	348	1206269038	348	1206269617	348	1206270117	536	1206270804	351
123	1206268482	350	1206269036	350	1206269615	350	1206270116	537	1206270803	352
124	1206268481	351	1206269035	351	1206269613	351	1206270114	539	1206270801	354
125	1206268479	353	1206269033	353	1206269612	353	1206270113	540	1206270800	355
126	1206268478	354	1206269032	354	1206269611	354	1206270111	542	1206270798	357
127	1206268476	356	1206269030	356	1206269609	356	1206270110	543	1206270796	359
128	1206268475	357	1206269029	357	1206269607	357	1206270108	545	1206270795	360
129	1206268473	359	1206269027	359	1206269606	359	1206270107	546	1206270793	362
130	1206268472	360	1206269026	360	1206269604	360	1206270105	548	1206270792	363
131	1206268470	362	1206269024	362	1206269603	362	1206270104	549	1206270790	365
132	1206268469	363	1206269023	363	1206269601	363	1206270102	551	1206270789	366
133	1206268467	365	1206269021	365	1206269600	365	1206270100	553	1206270788	367
134	1206268466	366	1206269020	366	1206269598	366	1206270099	554	1206270786	369
135	1206268464	368	1206269018	368	1206269597	368	1206270097	556	1206270785	370
136	1206268463	369	1206269016	369	1206269595	370	1206270097	557	1206270783	372
137	1206268461	371	1206269015	371	1206269594	371	1206270094	559	1206270782	373
138	1206268460	372	1206269013	372	1206269592	373	1206270093	560	1206270780	375
139	1206268458	374	1206269012	374	1206269590	374	1206270091	562	1206270779	376
140	1206268457	375	1206269010	375	1206269589	376	1206270090	563	1206270777	378
141	1206268455	377	1206269009	377	1206269587	377	1206270088	565	1206270776	379
	1206268453	379	1206269007	379	1206269586	379	1206270087	566	1206270774	381

143	1206268447	385	1206269006	385	1206269584	380	1206270085	568	1206270773	382
144	1206268446	386	1206269004	386	1206269583	382	1206270084	569	1206270771	384
145	1206268445	387	1206269003	387	1206269581	383	1206270082	571	1206270770	385
146	1206268443	389	1206269001	389	1206269580	385	1206270081	572	1206270768	387
147	1206268441	391	1206269000	391	1206269578	386	1206270079	574	1206270766	389
148	1206268440	392	1206268998	392	1206269577	388	1206270078	575	1206270765	390
149	1206268438	394	1206268997	394	1206269575	389	1206270076	577	1206270763	392
150	1206268437	395	1206268995	395	1206269574	391	1206270075	578	1206270762	393
151	1206268436	396	1206268994	396	1206269572	392	1206270073	580	1206270761	394
152	1206268434	398	1206268992	398	1206269571	394	1206270072	581	1206270759	396
153	1206268433	399	1206268991	399	1206269570	395	1206270070	583	1206270758	397
154	1206268431	401	1206268990	401	1206269568	396	1206270069	584	1206270756	399
155	1206268430	402	1206268988	402	1206269566	398	1206270067	586	1206270755	400
156	1206268428	404	1206268987	404	1206269565	399	1206270066	587	1206270753	402
157	1206268427	405	1206268985	405	1206269563	401	1206270064	589	1206270751	404
158	1206268425	407	1206268984	407	1206269562	402	1206270063	590	1206270750	405
159	1206268424	408	1206268982	408	1206269561	404	1206270061	592	1206270749	406
160	1206268422	410	1206268981	410	1206269559	405	1206270059	594	1206270747	408

CompT	Torrent Man	delbrot 2	Machines							
Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A
1	1206271472	0	1206271764	0	1206272077	0	1206272460	0	1206272787	0
2	1206271470	2	1206271764	2	1206272075	0	1206272458	2	1206272786	1
3	1206271470	2	1206271762	2	1206272075	2	1206272458	2	1206272786	1
4	1206271468	4	1206271762	4	1206272073	2	1206272456	4	1206272784	3
5	1206271468	4	1206271760	4	1206272073	4	1206272456	4	1206272784	3
6	1206271466	6	1206271760	6	1206272071	4	1206272454	6	1206272782	5
7	1206271466	6	1206271758	6	1206272071	6	1206272454	6	1206272782	5
8	1206271464	8	1206271758	8	1206272069	6	1206272452	8	1206272780	7
9	1206271464	8	1206271756	8	1206272069	8	1206272451	9	1206272779	8
10	1206271462	10	1206271755	10	1206272067	9	1206272450	10	1206272778	9
11	1206271462	10	1206271754	10	1206272067	10	1206272449	11	1206272778	9
12	1206271460	12	1206271754	12	1206272065	10	1206272447	13	1206272775	12
13	1206271460	12	1206271752	12	1206272065	12	1206272447	13	1206272775	12
14	1206271458	14	1206271751	14	1206272063	13	1206272445	15	1206272773	14
15	1206271458	14	1206271750	14	1206272062	14	1206272445	15	1206272773	14
16	1206271456	16	1206271748	16	1206272061	16	1206272443	17	1206272772	15
17	1206271456	16	1206271748	16	1206272060	16	1206272443	17	1206272771	16
18	1206271454	18	1206271747	18	1206272058	17	1206272441	19	1206272769	18
19	1206271454	18	1206271747	18	1206272058	17	1206272441	19	1206272769	18
20	1206271452	20	1206271745	20	1206272056	19	1206272439	21	1206272767	20
21	1206271452	20	1206271745	20	1206272056	19	1206272439	21	1206272767	20
22	1206271450	22	1206271743	22	1206272054	21	1206272437	23	1206272765	22
23	1206271449	23	1206271742	23	1206272054	22	1206272437	23	1206272765	22
24	1206271447	25	1206271740	25	1206272052	24	1206272435	25	1206272763	24
25	1206271447	25	1206271740	25	1206272052	24	1206272435	25	1206272763	24
26	1206271445	27	1206271738	27	1206272050	26	1206272433	27	1206272761	26
27	1206271445	27	1206271738	27	1206272050	26	1206272433	27	1206272761	26
28	1206271443	29	1206271736	29	1206272048	28	1206272430	30	1206272759	28
29	1206271442	30	1206271736	30	1206272048	28	1206272429	31	1206272759	28
30	1206271441	31	1206271734	31	1206272046	30	1206272428	32	1206272757	30
31	1206271440	32	1206271733	32	1206272046	31	1206272427	33	1206272757	30
32	1206271438	34	1206271732	34	1206272044	32	1206272425	35	1206272755	32
33	1206271437	35	1206271732	35	1206272044	32	1206272424	36	1206272755	32
34	1206271434	38	1206271729	38	1206272041	35	1206272422	38	1206272752	35
35	1206271434	38	1206271727	38	1206272039	37	1206272421	39	1206272750	37
36	1206271431	41	1206271725	41	1206272037	39	1206272418	42	1206272748	39
37	1206271429	43	1206271723	43	1206272035	41	1206272416	44	1206272746	41
38	1206271426	46	1206271721	46	1206272032	43	1206272414	46	1206272743	44

20	1206271425	47	1206271710	147	1206272021	16	1206272412	140	1206272742	45
39	1206271425	47	1206271718	47	1206272031	46	1206272412	48	1206272742	45
40	1206271423	49	1206271717	49	1206272028	47	1206272409	51	1206272739	48
41	1206271420	52	1206271714	52	1206272027	50	1206272408	52	1206272736	51
42	1206271418	54	1206271713	54	1206272024	51	1206272405	55	1206272735	52
43	1206271415	57	1206271709	57	1206272022	55	1206272404	56	1206272732	55
44	1206271414	58	1206271709	58	1206272019	55	1206272400	60	1206272731	56
45	1206271410	62	1206271705	62	1206272017	59	1206272399	61	1206272727	60
46	1206271410	62	1206271704	62	1206272013	60	1206272396	64	1206272727	60
47	1206271406	66	1206271700	66	1206272013	64	1206272394	66	1206272723	64
48	1206271405	67	1206271700	67	1206272009	64	1206272391	69	1206272722	65
49	1206271401	71	1206271695	71	1206272008	69	1206272389	71	1206272718	69
50	1206271400	72	1206271695	72	1206272004	69	1206272385	75	1206272718	69
51	1206271395	77	1206271690	77	1206272002	74	1206272384	76 81	1206272713	74
53	1206271395 1206271389	83	1206271690 1206271684	83	1206271998	80	1206272379	81	1206272713	80
54	1206271389	83	1206271684	83	1206271997	80	1206272379	86	1206272707	80
55	1206271383	89	1206271684	89	1206271992	86	1206272374	88	1206272707	86
56	1206271383	90	1206271677	90	1206271996	87	1206272372	92	1206272701	87
57	1206271382	95	1206271677	95	1206271986	91	1206272366	94	1206272700	92
58	1206271377	96	1206271673	96	1206271984	93	1206272360	98	1206272693	93
59	1206271376	100	1206271671	100	1206271980	93	1206272362	100	1206272694	97
60	1206271372	100	1206271667	100	1206271978	99	1206272356	104	1206272688	99
61	1206271370	102	1206271663	102	1206271974	102	1206272355	104	1206272685	102
62	1206271365	107	1206271660	107	1206271973	104	1206272353	108	1206272683	104
63	1206271363	111	1206271656	111	1206271968	104	1206272352	110	1206272683	104
64	1206271361	112	1206271655	112	1206271964	109	1206272347	113	1206272678	109
65	1206271366	116	1206271652	116	1206271963	112	1206272347	115	1206272675	112
66	1206271355	117	1206271650	117	1206271963	114	1206272343	117	1206272673	114
67	1206271353	120	1206271630	120	1206271958	117	1206272343	120	1206272670	117
68	1206271352	122	1206271647	122	1206271955	119	1206272340	122	1206272678	119
69	1206271347	125	1206271643	125	1206271953	122	1206272336	124	1206272665	122
70	1206271347	126	1206271641	126	1206271939	123	1206272333	127	1206272664	123
71	1206271343	129	1206271638	129	1206271949	126	1206272331	129	1206272661	126
72	1206271341	131	1206271636	131	1206271945	128	1206272329	131	1206272659	128
73	1206271338	134	1206271633	134	1206271944	131	1206272327	133	1206272656	131
74	1206271337	135	1206271631	135	1206271941	133	1206272325	135	1206272655	132
75	1206271334	138	1206271629	138	1206271939	135	1206272322	138	1206272652	135
76	1206271332	140	1206271627	140	1206271937	137	1206272320	140	1206272650	137
77	1206271330	142	1206271626	142	1206271935	138	1206272318	142	1206272648	139
78	1206271328	144	1206271623	144	1206271933	141	1206272317	143	1206272646	141
79	1206271326	146	1206271622	146	1206271931	142	1206272314	146	1206272644	143
80	1206271323	149	1206271619	149	1206271929	145	1206272313	147	1206272642	145
81	1206271322	150	1206271619	150	1206271927	145	1206272311	149	1206272641	146
82	1206271320	152	1206271616	152	1206271925	148	1206272310	150	1206272639	148
83	1206271319	153	1206271616	153	1206271925	148	1206272308	152	1206272638	149
84	1206271317	155	1206271613	155	1206271922	151	1206272307	153	1206272636	151
85	1206271317	155	1206271613	155	1206271922	151	1206272306	154	1206272635	152
86	1206271315	157	1206271611	157	1206271920	153	1206272306	154	1206272634	153
87	1206271315	157	1206271611	157	1206271920	153	1206272304	156	1206272633	154
88	1206271313	159	1206271609	159	1206271918	155	1206272303	157	1206272631	156
89	1206271313	159	1206271607	159	1206271916	157	1206272301	159	1206272630	157
90	1206271311	161	1206271606	161	1206271916	158	1206272301	159	1206272629	158
91	1206271311	161	1206271605	161	1206271914	159	1206272299	161	1206272628	159
92	1206271308	164	1206271603	164	1206271913	161	1206272298	162	1206272626	161
93	1206271308	164	1206271602	164	1206271911	162	1206272296	164	1206272625	162
94	1206271305	167	1206271600	167	1206271910	164	1206272294	166	1206272623	164
95	1206271305	167	1206271599	167	1206271908	165	1206272293	167	1206272621	166
96	1206271302	170	1206271597	170	1206271907	167	1206272292	168	1206272620	167
97	1206271301	171	1206271596	171	1206271905	168	1206272290	170	1206272618	169
				I .						

98 1206271300 172 1206271594 172 99 1206271298 174 1206271593 174 100 1206271296 176 1206271592 176 101 1206271294 178 1206271590 178 102 1206271293 179 1206271590 179 103 1206271292 180 1206271587 180 104 1206271290 182 1206271586 182 105 1206271289 183 1206271584 183 106 1206271287 185 1206271584 185 107 1206271287 185 1206271582 185 108 1206271285 187 1206271580 187 109 1206271283 189 1206271580 189 111 1206271283 189 1206271578 189	1206271904 1206271902 1206271899 1206271897 1206271897 1206271895 1206271895 1206271894 1206271892 1206271890 1206271888 1206271888 1206271888 1206271886 1206271886	170 171 172 174 174 177 178 180 180 182 182 184 184 186	1206272289 1206272287 1206272286 1206272284 1206272283 1206272281 1206272279 1206272279 1206272276 1206272275 1206272275 1206272273 1206272271 1206272271	171 173 174 176 177 179 181 181 184 185 185 187	1206272617 1206272615 1206272614 1206272611 1206272610 1206272610 1206272608 1206272607 1206272605 1206272605 1206272603 1206272603	170 172 173 174 176 177 179 180 182 182 184
100 1206271296 176 1206271592 176 101 1206271294 178 1206271590 178 102 1206271293 179 1206271590 179 103 1206271292 180 1206271587 180 104 1206271290 182 1206271586 182 105 1206271289 183 1206271584 183 106 1206271287 185 1206271584 185 107 1206271287 185 1206271582 185 108 1206271285 187 1206271580 187 109 1206271283 189 1206271580 189	1206271899 1206271899 1206271897 1206271897 1206271895 1206271894 1206271892 1206271892 1206271890 1206271888 1206271888 1206271886 1206271886	172 174 177 178 180 180 182 182 184 184	1206272286 1206272284 1206272283 1206272281 1206272279 1206272279 1206272276 1206272275 1206272275 1206272273 1206272271	174 176 177 179 181 181 184 185 185	1206272614 1206272613 1206272611 1206272610 1206272608 1206272607 1206272605 1206272605 1206272603	173 174 176 177 179 180 182 182
101 1206271294 178 1206271590 178 102 1206271293 179 1206271590 179 103 1206271292 180 1206271587 180 104 1206271290 182 1206271586 182 105 1206271289 183 1206271584 183 106 1206271287 185 1206271584 185 107 1206271287 185 1206271582 185 108 1206271285 187 1206271582 187 109 1206271285 187 1206271580 187 110 1206271283 189 1206271580 189	1206271899 1206271897 1206271897 1206271895 1206271895 1206271892 1206271892 1206271890 1206271888 1206271888 1206271886 1206271886	174 177 178 180 180 182 182 184 184	1206272284 1206272283 1206272281 1206272279 1206272276 1206272276 1206272275 1206272275 1206272273 1206272271	176 177 179 181 181 184 185 185	1206272613 1206272611 1206272610 1206272608 1206272607 1206272605 1206272605 1206272603	174 176 177 179 180 182 182
102 1206271293 179 1206271590 179 103 1206271292 180 1206271587 180 104 1206271290 182 1206271586 182 105 1206271289 183 1206271584 183 106 1206271287 185 1206271584 185 107 1206271287 185 1206271582 185 108 1206271285 187 1206271582 187 109 1206271285 187 1206271580 187 110 1206271283 189 1206271580 189	1206271897 1206271897 1206271895 1206271894 1206271892 1206271890 1206271888 1206271888 1206271886 1206271886	174 177 178 180 180 182 182 184 184	1206272283 1206272281 1206272279 1206272279 1206272276 1206272275 1206272275 1206272273 1206272271	177 179 181 181 184 185 185	1206272611 1206272610 1206272608 1206272607 1206272605 1206272605 1206272603	176 177 179 180 182 182
103 1206271292 180 1206271587 180 104 1206271290 182 1206271586 182 105 1206271289 183 1206271584 183 106 1206271287 185 1206271584 185 107 1206271287 185 1206271582 185 108 1206271285 187 1206271582 187 109 1206271285 187 1206271580 187 110 1206271283 189 1206271580 189	1206271897 1206271895 1206271894 1206271892 1206271892 1206271890 1206271888 1206271888 1206271886 1206271886	177 178 180 180 182 182 184 184	1206272281 1206272279 1206272279 1206272276 1206272275 1206272275 1206272273 1206272271	179 181 181 184 185 185	1206272610 1206272608 1206272607 1206272605 1206272605 1206272603	177 179 180 182 182 184
104 1206271290 182 1206271586 182 105 1206271289 183 1206271584 183 106 1206271287 185 1206271584 185 107 1206271287 185 1206271582 185 108 1206271285 187 1206271582 187 109 1206271285 187 1206271580 187 110 1206271283 189 1206271580 189	1206271894 1206271894 1206271892 1206271892 1206271890 1206271888 1206271888 1206271886 1206271886	178 180 180 182 182 184 184	1206272279 1206272279 1206272276 1206272275 1206272275 1206272273 1206272271	181 181 184 185 185	1206272608 1206272607 1206272605 1206272605 1206272603	179 180 182 182 184
105 1206271289 183 1206271584 183 106 1206271287 185 1206271584 185 107 1206271287 185 1206271582 185 108 1206271285 187 1206271582 187 109 1206271285 187 1206271580 187 110 1206271283 189 1206271580 189	1206271894 1206271892 1206271892 1206271890 1206271888 1206271888 1206271886 1206271886	180 180 182 182 184 184	1206272279 1206272276 1206272275 1206272275 1206272273 1206272271	181 184 185 185 187	1206272607 1206272605 1206272605 1206272603	180 182 182 184
106 1206271287 185 1206271584 185 107 1206271287 185 1206271582 185 108 1206271285 187 1206271582 187 109 1206271285 187 1206271580 187 110 1206271283 189 1206271580 189	1206271892 1206271892 1206271890 1206271888 1206271888 1206271886 1206271886	180 182 182 184 184 186	1206272276 1206272275 1206272275 1206272273 1206272271	184 185 185 187	1206272605 1206272605 1206272603	182 182 184
107 1206271287 185 1206271582 185 108 1206271285 187 1206271582 187 109 1206271285 187 1206271580 187 110 1206271283 189 1206271580 189	1206271892 1206271890 1206271888 1206271888 1206271886 1206271886 1206271884	182 182 184 184 186	1206272275 1206272275 1206272273 1206272271	185 185 187	1206272605 1206272603	182
108 1206271285 187 1206271582 187 109 1206271285 187 1206271580 187 110 1206271283 189 1206271580 189	1206271880 1206271888 1206271888 1206271886 1206271886 1206271884	182 184 184 186	1206272275 1206272273 1206272271	185 187	1206272603	184
109 1206271285 187 1206271580 187 110 1206271283 189 1206271580 189	1206271888 1206271888 1206271886 1206271886 1206271884	184 184 186	1206272273 1206272271	187		
110 1206271283 189 1206271580 189	1206271888 1206271886 1206271886 1206271884	184 186	1206272271		1206272603	184
	1206271886 1206271886 1206271884	186		189		1 -0 -
111 1206271283 189 1206271578 189	1206271886 1206271884		1206272271		1206272601	186
	1206271884	186		189	1206272601	186
112 1206271281 191 1206271578 191			1206272270	190	1206272599	188
113 1206271281 191 1206271576 191		188	1206272269	191	1206272599	188
114 1206271279 193 1206271575 193	1206271884	189	1206272267	193	1206272597	190
115 1206271279 193 1206271574 193	1206271882	190	1206272267	193	1206272597	190
				195		192
	1206271882	190	1206272265		1206272595	
117 1206271277 195 1206271572 195	1206271880	192	1206272265	195	1206272594	193
118 1206271274 198 1206271572 198	1206271880	192	1206272263	197	1206272593	194
119 1206271274 198 1206271570 198	1206271878	194	1206272263	197	1206272592	195
120 1206271272 200 1206271569 200	1206271878	195	1206272261	199	1206272590	197
121 1206271272 200 1206271568 200	1206271876	196	1206272260	200	1206272590	197
122 1206271270 202 1206271568 202	1206271876	196	1206272259	201	1206272588	199
123 1206271270 202 1206271565 202	1206271874	199	1206272259	201	1206272588	199
124 1206271268 204 1206271565 204	1206271874	199	1206272257	203	1206272586	201
125 1206271268 204 1206271563 204	1206271872	201	1206272257	203	1206272586	201
126 1206271266 206 1206271563 206	1206271871	201	1206272255	205	1206272584	203
127 1206271266 206 1206271561 206	1206271870	203	1206272254	206	1206272584	203
128 1206271264 208 1206271561 208	1206271870	203	1206272252	208	1206272582	205
129 1206271264 208 1206271559 208	1206271868	205	1206272252	208	1206272582	205
130 1206271262 210 1206271559 210	1206271867	205	1206272250	210	1206272580	207
131 1206271262 210 1206271557 210	1206271866	207	1206272250	210	1206272580	207
132 1206271260 212 1206271557 212	1206271865	207	1206272248	212	1206272578	209
133 1206271259 213 1206271555 213	1206271863	209	1206272248	212	1206272578	209
		209		214		211
	1206271863		1206272246		1206272576	
135 1206271257 215 1206271553 215	1206271861	211	1206272246	214	1206272576	211
136 1206271255 217 1206271553 217	1206271861	211	1206272244	216	1206272574	213
137 1206271255 217 1206271551 217	1206271859	213	1206272244	216	1206272574	213
138 1206271253 219 1206271551 219	1206271859	213	1206272242	218	1206272572	215
139 1206271253 219 1206271549 219	1206271857	215	1206272242	218	1206272572	215
140 1206271251 221 1206271549 221	1206271857	215	1206272240	220	1206272570	217
141 1206271251 221 1206271547 221	1206271855	217	1206272240	220	1206272570	217
142 1206271249 223 1206271546 223	1206271855	218	1206272238	222	1206272568	219
143 1206271249 223 1206271545 223	1206271853	219	1206272238	222	1206272568	219
144 1206271247 225 1206271545 225	1206271853	219	1206272236	224	1206272566	221
145 1206271247 225 1206271543 225	1206271851	221	1206272236	224	1206272566	221
146 1206271245 227 1206271542 227	1206271851	222	1206272234	226	1206272564	223
147 1206271245 227 1206271540 227	1206271849	224	1206272233	227	1206272564	223
148 1206271243 229 1206271540 229	1206271849	224	1206272232	228	1206272562	225
149 1206271243 229 1206271538 229	1206271847	226	1206272232	228	1206272562	225
150 1206271240 232 1206271538 232	1206271847	226	1206272230	230	1206272560	227
150 1200271240 232 1200271536 232 151 1206271240 232 1206271536 232	1206271847	228	1206272229	231	1206272560	227
152 1206271238 234 1206271536 234 1206271536 234 1206271534 234	1206271845	228	1206272228	232	1206272558	229
153 1206271238 234 1206271534 234	1206271843	230	1206272227	233	1206272557	230
154 1206271236 236 1206271534 236	1206271843	230	1206272225	235	1206272556	231
155 1206271236 236 1206271532 236	1206271841	232	1206272225	235	1206272556	231
156 1206271234 238 1206271532 238	1206271840	232	1206272224	236	1206272554	233

157	1206271234	238	1206271530	238	1206271839	234	1206272222	238	1206272553	234
158	1206271232	240	1206271529	240	1206271839	235	1206272221	239	1206272551	236
159	1206271232	240	1206271528	240	1206271836	236	1206272220	240	1206272551	236
160	1206271230	242	1206271526	242	1206271835	238	1206272219	241	1206272550	237

Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A
1	1206526420	0 Kun 1 A	1206526780	0 Run 2 A	1207203845	0 Kun 3 A	1207204253	0 Run 4 A	1207204597	0
2	1206526418	2	1206526777	2	1207203845	3	1207204253	0	1207204596	1
3	1206526417	3	1206526777	3	1207203843	3	1207204253	2	1207204595	2
4	1206526416	4	1206526776	4	1207203842	4	1207204251	2	1207204595	2
5	1206526416	4	1206526775	4	1207203842	5	1207204251	2	1207204593	4
6	1206526414	6	1206526774	6	1207203842	6	1207204251	3	1207204593	5
7	1206526413	7	1206526772	7	1207203840	8	1207204230	5	1207204591	6
8	1206526413	7	1206526771	7	1207203840	9	1207204248	5	1207204591	6
9	1206526410	10	1206526770	10	1207203840	10	1207204248	5	1207204591	7
10	1206526409	11	1206526768	11	1207203840	12	1207204245	8	1207204587	10
11	1206526408	12	1206526767	12	1207203837	13	1207204245	8	1207204587	10
12	1206526408	12	1206526766	12	1207203836	14	1207204243	9	1207204387	11
13	1206526406	14	1206526764	14	1207203833	16	1207204244	10	1207204586	11
14	1206526405	15	1206526763	15	1207203833	17	1207204243	12	1207204584	13
15	1206526404	16	1206526762	16	1207203833	18	1207204241	12	1207204582	15
16	1206526403		1206526760		1207203831		1207204239		1207204582	
17	1206526402	18	1206526759	18	1207203829	21	1207204238	15	1207204581	16
18	1206526402	18	1206526757	18	1207203829	23	1207204237	16	1207204581	16
19	1206526401	19	1206526756	19	1207203828	24	1207204237	16	1207204578	19
20	1206526398	22	1206526755	22	1207203826	25	1207204234	19	1207204578	19
21	1206526397	23	1206526755	23	1207203826	25	1207204233	20	1207204578	19
22	1206526397	23	1206526753	23	1207203823	27	1207204232	21	1207204576	21
23	1206526395	25	1206526752	25	1207203823	28	1207204232	21	1207204575	22
24	1206526393	27	1206526750	27	1207203823	30	1207204229	24	1207204575	22
25	1206526393	27	1206526750	27	1207203822	30	1207204228	25	1207204573	24
26	1206526391	29	1206526749	29	1207203820	31	1207204228	25	1207204572	25
27	1206526391	29	1206526747	29	1207203820	33	1207204225	28	1207204572	25
28	1206526389	31	1206526746	31	1207203817	34	1207204225	28	1207204570	27
29	1206526388	32	1206526744	32	1207203817	36	1207204224	29	1207204569	28
30	1206526388	32	1206526743	32	1207203816	37	1207204222	31	1207204568	29
31	1206526387	33	1206526741	33	1207203816	39	1207204222	31	1207204567	30
32	1206526386	34	1206526740	34	1207203814	40	1207204221	32	1207204566	31
33	1206526383	37	1206526738	37	1207203813	42	1207204218	35	1207204565	32
34	1206526382	38	1206526736	38	1207203812	44	1207204218	35	1207204563	34
35	1206526380	40	1206526734	40	1207203810	46	1207204216	37	1207204562	35
36	1206526378	42	1206526732	42	1207203808	48	1207204216	37	1207204558	39
37	1206526377	43	1206526730	43	1207203806	50	1207204216	37	1207204556	41
38	1206526376	44	1206526729	44	1207203806	51	1207204213	40	1207204556	41
39	1206526375	45	1206526727	45	1207203804	53	1207204211	42	1207204555	42
40	1206526373	47	1206526725	47	1207203802	55	1207204211	42	1207204553	44
41	1206526373	47	1206526723	47	1207203801	57	1207204211	42	1207204550	47
42	1206526370	50	1206526721	50	1207203800	59	1207204208	45	1207204550	47
43	1206526368	52	1206526719	52	1207203798	61	1207204206	47	1207204548	49
44	1206526368	52	1206526717	52	1207203796	63	1207204204	49	1207204547	50
45	1206526366	54	1206526716	54	1207203796	64	1207204204	49	1207204546	51
46	1206526364	56	1206526714	56	1207203795	66	1207204202	51	1207204544	53
47	1206526363	57	1206526711	57	1207203792	69	1207204201	52	1207204544	53
48	1206526362	58	1206526710	58	1207203790	70	1207204199	54	1207204541	56
49	1206526360	60	1206526708	60	1207203789	72	1207204198	55	1207204540	57
50	1206526358	62	1206526705	62	1207203789	75	1207204196	57	1207204538	59
51	1206526357	63	1206526704	63	1207203786	76	1207204196	57	1207204538	59
52	1206526355	65	1206526703	65	1207203784	77	1207204193	60	1207204536	61

	53	1206526352	68	1206526700	68	1207203782	80	1207204192	61	1207204534	63
	57	1206526345	75	1206526692	75	1207203774	88	1207204185	68	1207204529	68
	58	1206526344	76	1206526690	76	1207203774	90	1207204185	68	1207204525	72
	59	1206526343	77	1206526689	77	1207203770	91	1207204182	71	1207204524	73
	60	1206526342	78	1206526685	78	1207203769	95	1207204181	72	1207204523	74
	61	1206526338	82	1206526683	82	1207203768	97	1207204179	74	1207204522	75
	62	1206526337	83	1206526681	83	1207203767	99	1207204179	74	1207204519	78
1206.526.531 89	63	1206526336	84	1206526680	84	1207203763	100	1207204176	77	1207204517	80
	64	1206526334	86	1206526678	86	1207203762	102	1207204174	79	1207204517	80
	65	1206526331	89	1206526676	89	1207203761	104	1207204174	79	1207204517	80
	66	1206526331	89	1206526674	89	1207203761	106	1207204171	82	1207204513	84
	67	1206526329	91	1206526673	91	1207203758	107	1207204170	83	1207204512	85
	68	1206526328	92	1206526670	92	1207203757	110	1207204169	84	1207204508	89
1206526324 96											
1206526320 100											
1206526319 101 1206526659 101 1207203749 121 1207204157 96 1207204500 97											
Texas											
Table											
To	75	1206526317	103	1206526657	103	1207203747	123	1207204156	97	1207204498	99
1206526314 100	76	1206526316	104	1206526656	104	1207203747	124	1207204155	98	1207204497	100
1206526311 109	77	1206526315	105	1206526654	105	1207203746	126	1207204153	100	1207204496	101
80	78	1206526314	106	1206526653	106	1207203743	127	1207204152	101	1207204495	102
81 1206526310 110 1206526647 110 1207203740 133 1207204148 105 1207204492 105 82 1206526308 112 1206526646 112 1207203738 134 1207204146 107 1207204489 108 83 1206526307 113 1206526643 113 1207203738 135 1207204144 109 1207204488 109 84 1206526301 116 1206526642 116 1207203735 137 1207204144 109 1207204488 109 85 1206526303 117 1206526641 117 1207203735 138 1207204142 111 1207204488 112 86 1206526301 119 1206526640 119 1207203735 139 1207204142 111 1207204488 113 88 1206526301 119 1206526638 119 1207203732 142 1207204138 115 1207204484 113 89 1206526639	79	1206526311	109	1206526650	109	1207203742	130	1207204150	103	1207204493	104
82 1206526308 112 1206526466 112 1207203738 134 1207204146 107 1207204489 108 83 1206526307 113 1206526645 113 1207203738 135 1207204145 108 1207204488 109 84 1206526306 114 1206526643 114 1207203736 137 1207204144 109 1207204488 109 85 1206526304 116 1206526642 116 1207203735 138 1207204143 110 1207204488 111 86 1206526303 117 1206526640 119 1207203734 140 1207204141 111 1207204488 112 87 1206526301 119 1206526638 119 1207203734 142 1207204141 112 1207204484 113 89 1206526631 119 1206526637 122 1207203732 143 1207204138 115 1207204482 115 90 1206526629	80	1206526310	110	1206526649	110	1207203741	131	1207204148	105	1207204493	104
83 1206526907 113 1206526645 113 1207203738 135 1207204145 108 1207204488 109 84 1206526306 114 1206526643 114 1207203736 137 1207204144 109 1207204488 109 85 1206526303 116 1206526642 116 1207203736 138 1207204143 110 1207204485 111 86 1206526303 117 1206526641 117 1207203735 139 1207204142 111 1207204485 112 87 1206526301 119 1206526640 119 1207203734 140 1207204141 112 1207204484 113 89 1206526298 122 1206526637 122 1207203732 143 1207204138 115 1207204478 115 90 1206526296 124 1206526634 124 1207203731 144 1207204137 116 1207204479 118 91 1206526293	81	1206526310	110	1206526647	110	1207203740	133	1207204148	105	1207204492	105
84 1206526906 114 1206526431 114 1207203736 137 1207204144 109 1207204488 109 85 1206526304 116 1206526422 116 1207203736 138 1207204143 110 1207204486 111 86 1206526303 117 1206526641 117 1207203735 139 1207204142 111 1207204485 112 87 1206526301 119 1206526638 119 1207203734 140 1207204141 112 1207204484 113 89 120652698 122 1206526638 119 1207203734 142 1207204138 115 1207204484 113 89 1206526298 122 1206526637 122 1207203734 142 1207204138 115 1207204482 115 90 1206526296 124 1206526634 124 1207203737 146 1207204138 115 1207204479 118 91 12065266293	82	1206526308	112	1206526646	112	1207203738	134	1207204146	107	1207204489	108
85 1206526304 116 1206526642 116 1207203736 138 1207204143 110 1207204486 111 86 1206526303 117 1206526641 117 1207203735 139 1207204142 111 1207204485 112 87 1206526301 119 1206526638 119 1207203734 140 1207204141 112 1207204484 113 88 1206526301 119 1206526637 122 1207203734 142 1207204138 115 1207204482 115 90 1206526296 124 1206526636 124 1207203731 144 1207204138 115 1207204479 118 91 1206526296 124 1206526636 124 1207203731 144 1207204138 115 1207204479 118 91 1206526296 124 1206526631 124 1207203727 147 1207204137 116 1207204477 120 92 12065262293 <td>83</td> <td>1206526307</td> <td>113</td> <td>1206526645</td> <td>113</td> <td>1207203738</td> <td>135</td> <td>1207204145</td> <td>108</td> <td>1207204488</td> <td>109</td>	83	1206526307	113	1206526645	113	1207203738	135	1207204145	108	1207204488	109
86 1206526303 117 1206526641 117 1207203735 139 1207204142 111 1207204485 112 87 1206526301 119 1206526640 119 1207203734 140 1207204141 112 1207204484 113 88 1206526301 119 1206526638 119 1207203734 142 1207204181 112 1207204484 113 89 1206526296 124 1206526636 124 1207203732 143 1207204138 115 1207204479 118 90 1206526296 124 1206526634 124 1207203731 144 1207204137 116 1207204479 118 92 1206526294 126 1206526633 126 1207203727 147 1207204137 116 1207204477 120 93 1206526293 128 1206526630 127 1207203727 147 1207204137 117 1207204477 120 94 1206526290	84	1206526306	114	1206526643	114	1207203736	137	1207204144	109	1207204488	109
86 1206526303 117 1206526641 117 1207203735 139 1207204142 111 1207204485 112 87 1206526301 119 1206526640 119 1207203734 140 1207204141 112 1207204484 113 88 1206526301 119 1206526638 119 1207203734 142 1207204141 112 1207204484 113 89 1206526296 124 1206526636 124 1207203731 144 1207204138 115 1207204479 118 91 1206526296 124 1206526634 124 1207203731 146 1207204137 116 1207204477 128 92 1206526294 126 1206526633 126 1207203727 147 1207204137 116 1207204477 120 93 1206526293 127 1206526630 127 1207203722 150 1207204137 118 1207204477 120 94 1206526292	85	1206526304	116	1206526642	116	1207203736	138	1207204143	110	1207204486	111
87 1206526301 119 1206526640 119 1207203734 140 1207204141 112 1207204484 113 88 1206526301 119 120652638 119 1207203734 142 1207204141 112 1207204484 113 89 1206526298 122 1206526637 122 1207203732 143 1207204138 115 1207204479 118 90 1206526296 124 1206526636 124 1207203730 146 1207204137 116 1207204479 118 92 1206526294 126 1206526633 126 1207203727 147 1207204137 116 1207204477 120 93 1206526293 127 1206526633 127 1207203727 150 1207204135 118 1207204477 120 94 1206526293 128 1206526629 128 1207203727 150 1207204133 119 1207204477 120 95 1206526293		1206526303	117	1206526641	117	1207203735	139	1207204142	111		112
88 1206526301 119 1206526388 119 1207203734 142 1207204141 112 1207204484 113 89 1206526298 122 1206526637 122 1207203732 143 1207204138 115 1207204482 115 90 1206526296 124 1206526636 124 1207203731 144 1207204138 115 1207204479 118 91 1206526296 124 1206526634 124 1207203730 146 1207204137 116 1207204477 120 93 1206526294 126 1206526630 127 1207203727 147 1207204135 118 1207204477 120 94 1206526293 127 1206526630 127 1207203727 150 1207204134 119 1207204477 120 95 1206526291 129 1206526262 128 1207203724 153 1207204133 120 1207204477 120 95 12065262891 <td></td> <td>1206526301</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>		1206526301									
89 1206526298 122 120652637 122 1207203732 143 1207204138 115 1207204482 115 90 1206526296 124 1206526636 124 1207203731 144 1207204138 115 1207204479 118 91 1206526296 124 1206526634 124 1207203730 146 1207204137 116 1207204479 118 92 1206526294 126 1206526633 126 1207203727 147 1207204136 117 1207204477 120 93 1206526293 127 1206526630 127 1207203727 150 1207204135 118 1207204477 120 94 1206526292 128 1207203722 151 1207204134 119 1207204477 120 95 1206526291 129 1206526627 129 1207203722 154 120720413 120 1207204472 125 97 1206526288 132 12065262625											
90 1206526296 124 120652636 124 1207203731 144 1207204138 115 1207204479 118 91 1206526296 124 1206526634 124 1207203730 146 1207204137 116 1207204479 118 92 1206526294 126 1206526633 126 1207203727 147 1207204136 117 1207204477 120 93 1206526293 127 1206526630 127 1207203727 150 1207204135 118 1207204477 120 94 1206526292 128 1206526629 128 1207203725 151 1207204134 119 1207204477 120 95 1206526291 129 1206526627 129 1207203722 154 1207204133 120 1207204472 125 96 1206526288 132 1206526625 132 1207203722 155 1207204131 122 1207204472 125 98 1206526288											
1											
92 1206526294 126 120652633 126 1207203727 147 1207204136 117 1207204477 120 93 1206526293 127 1206526630 127 1207203727 150 1207204135 118 1207204477 120 94 1206526292 128 1206526629 128 1207203725 151 1207204134 119 1207204477 120 95 1206526291 129 1206526627 129 1207203724 153 1207204133 120 1207204474 123 96 1206526290 130 1206526626 130 1207203722 155 1207204131 122 1207204472 125 97 1206526288 132 1206526625 132 1207203722 155 1207204130 123 1207204472 125 98 1206526281 133 1206526623 133 1207203718 157 1207204130 123 1207204472 125 99 1206526286											
1206526293 127											
1206526292 128											
95 1206526291 129 1206526627 129 1207203724 153 1207204133 120 1207204474 123 96 1206526290 130 1206526626 130 1207203722 154 1207204131 122 1207204472 125 97 1206526288 132 120652625 132 1207203722 155 1207204130 123 1207204472 125 98 1206526287 133 1206526623 133 1207203721 157 1207204130 123 1207204472 125 99 1206526286 134 1206526621 134 1207203718 159 1207204128 125 1207204469 128 100 1206526284 136 1206526620 136 1207203718 160 1207204127 126 1207204469 128 101 1206526284 136 1206526619 136 1207203716 161 1207204126 127 1207204467 130 102 1206526281 <td></td> <td>1206526293</td> <td>127</td> <td></td> <td></td> <td>1207203727</td> <td>150</td> <td>1207204135</td> <td></td> <td>1207204477</td> <td></td>		1206526293	127			1207203727	150	1207204135		1207204477	
96 1206526290 130 1206526626 130 1207203722 154 1207204131 122 1207204472 125 97 1206526288 132 1206526625 132 1207203722 155 1207204130 123 1207204472 125 98 1206526287 133 1206526623 133 1207203721 157 1207204130 123 1207204472 125 99 1206526286 134 1206526621 134 1207203718 159 1207204128 125 1207204469 128 100 1206526284 136 1206526620 136 1207203718 160 1207204127 126 1207204469 128 101 1206526284 136 1206526619 136 1207203716 161 1207204126 127 1207204467 130 102 1206526281 139 1206526617 139 1207203716 163 1207204124 129 1207204466 131 103 1206526280<										1207204477	
97 1206526288 132 1206526625 132 1207203722 155 1207204130 123 1207204472 125 98 1206526287 133 1206526623 133 1207203721 157 1207204130 123 1207204472 125 99 1206526286 134 1206526621 134 1207203718 159 1207204128 125 1207204469 128 100 1206526284 136 1206526620 136 1207203718 160 1207204127 126 1207204469 128 101 1206526284 136 1206526619 136 1207203716 161 1207204126 127 1207204467 130 102 1206526281 139 1206526617 139 1207203716 163 1207204124 129 1207204466 131 103 1206526280 140 1206526615 140 1207203716 165 1207204124 129 1207204466 131 104 1206526279	95	1206526291	129	1206526627	129	1207203724	153	1207204133	120	1207204474	123
98 1206526287 133 1206526623 133 1207203721 157 1207204130 123 1207204472 125 99 1206526286 134 1206526621 134 1207203718 159 1207204128 125 1207204469 128 100 1206526284 136 1206526620 136 1207203718 160 1207204127 126 1207204469 128 101 1206526284 136 1206526619 136 1207203716 161 1207204126 127 1207204467 130 102 1206526281 139 1206526617 139 1207203716 163 1207204124 129 1207204466 131 103 1206526280 140 1206526615 140 1207203716 165 1207204124 129 1207204466 131 104 1206526279 141 1206526614 141 1207203712 166 1207204124 129 1207204464 133 105 120652627	96	1206526290	130	1206526626	130	1207203722	154	1207204131	122	1207204472	125
99 1206526286 134 1206526621 134 1207203718 159 1207204128 125 1207204469 128 100 1206526284 136 1206526620 136 1207203718 160 1207204127 126 1207204469 128 101 1206526284 136 1206526619 136 1207203716 161 1207204126 127 1207204467 130 102 1206526281 139 1206526617 139 1207203716 163 1207204124 129 1207204466 131 103 1206526280 140 1206526615 140 1207203716 165 1207204124 129 1207204466 131 104 1206526279 141 1206526614 141 1207203712 166 1207204124 129 1207204464 133 105 1206526277 143 1206526613 143 1207203711 167 1207204121 132 1207204462 135 106 12065262	97	1206526288	132	1206526625	132	1207203722	155	1207204130	123	1207204472	125
100 1206526284 136 1206526620 136 1207203718 160 1207204127 126 1207204469 128 101 1206526284 136 1206526619 136 1207203716 161 1207204126 127 1207204467 130 102 1206526281 139 1206526617 139 1207203716 163 1207204124 129 1207204466 131 103 1206526280 140 1206526615 140 1207203716 165 1207204124 129 1207204466 131 104 1206526279 141 1206526614 141 1207203712 166 1207204124 129 1207204464 133 105 1206526277 143 1206526613 143 1207203711 167 1207204124 129 1207204462 135 106 1206526276 144 1206526611 144 1207203711 167 1207204120 133 1207204462 135 107 1206526	98	1206526287	133	1206526623	133	1207203721	157	1207204130	123	1207204472	125
101 1206526284 136 1206526619 136 1207203716 161 1207204126 127 1207204467 130 102 1206526281 139 1206526617 139 1207203716 163 1207204124 129 1207204466 131 103 1206526280 140 1206526615 140 1207203716 165 1207204124 129 1207204466 131 104 1206526279 141 1206526614 141 1207203712 166 1207204124 129 1207204464 133 105 1206526277 143 1206526613 143 1207203711 167 1207204121 132 1207204462 135 106 1206526276 144 1206526611 144 1207203711 169 1207204120 133 1207204462 135 107 1206526275 145 1206526609 145 1207203711 171 1207204119 134 1207204462 135 108 1206526	99	1206526286	134	1206526621	134	1207203718	159	1207204128	125	1207204469	128
102 1206526281 139 1206526617 139 1207203716 163 1207204124 129 1207204466 131 103 1206526280 140 1206526615 140 1207203716 165 1207204124 129 1207204466 131 104 1206526279 141 1206526614 141 1207203712 166 1207204124 129 1207204464 133 105 1206526277 143 1206526613 143 1207203711 167 1207204121 132 1207204462 135 106 120652676 144 1206526611 144 1207203711 169 1207204120 133 1207204462 135 107 120652675 145 1206526609 145 1207203711 171 1207204119 134 1207204462 135 108 120652674 146 1206526608 146 1207203708 172 1207204119 134 1207204459 138 109 120652672<	100	1206526284	136	1206526620	136	1207203718	160	1207204127	126	1207204469	128
103 1206526280 140 1206526615 140 1207203716 165 1207204124 129 1207204466 131 104 1206526279 141 1206526614 141 1207203712 166 1207204124 129 1207204464 133 105 1206526277 143 1206526613 143 1207203711 167 1207204121 132 1207204462 135 106 1206526276 144 1206526611 144 1207203711 169 1207204120 133 1207204462 135 107 1206526275 145 1206526609 145 1207203711 171 1207204119 134 1207204462 135 108 1206526274 146 1206526608 146 1207203708 172 1207204119 134 1207204459 138 109 1206526273 147 1206526606 147 1207203708 174 1207204117 136 1207204459 138 110 1206526	101	1206526284	136	1206526619	136	1207203716	161	1207204126	127	1207204467	130
103 1206526280 140 1206526615 140 1207203716 165 1207204124 129 1207204466 131 104 1206526279 141 1206526614 141 1207203712 166 1207204124 129 1207204464 133 105 1206526277 143 1206526613 143 1207203711 167 1207204121 132 1207204462 135 106 1206526276 144 1206526611 144 1207203711 169 1207204120 133 1207204462 135 107 1206526275 145 1206526609 145 1207203711 171 1207204119 134 1207204462 135 108 1206526274 146 1206526608 146 1207203708 172 1207204119 134 1207204459 138 109 1206526273 147 1206526606 147 1207203708 174 1207204117 136 1207204459 138 110 1206526	102	1206526281	139	1206526617	139	1207203716	163	1207204124	129	1207204466	131
104 1206526279 141 1206526614 141 1207203712 166 1207204124 129 1207204464 133 105 1206526277 143 1206526613 143 1207203711 167 1207204121 132 1207204462 135 106 120652676 144 1206526611 144 1207203711 169 1207204120 133 1207204462 135 107 120652675 145 1206526609 145 1207203711 171 1207204119 134 1207204462 135 108 120652674 146 1206526608 146 1207203708 172 1207204119 134 1207204459 138 109 120652673 147 1206526606 147 1207203708 174 1207204117 136 1207204459 138 110 1206526272 148 1206526605 148 1207203706 175 1207204115 138 1207204458 139											
105 1206526277 143 1206526613 143 1207203711 167 1207204121 132 1207204462 135 106 1206526276 144 1206526611 144 1207203711 169 1207204120 133 1207204462 135 107 1206526275 145 1206526609 145 1207203711 171 1207204119 134 1207204462 135 108 1206526274 146 1206526608 146 1207203708 172 1207204119 134 1207204459 138 109 1206526273 147 1206526606 147 1207203708 174 1207204117 136 1207204459 138 110 1206526272 148 1206526605 148 1207203706 175 1207204115 138 1207204458 139											
106 1206526276 144 1206526611 144 1207203711 169 1207204120 133 1207204462 135 107 120652675 145 1206526609 145 1207203711 171 1207204119 134 1207204462 135 108 120652674 146 1206526608 146 1207203708 172 1207204119 134 1207204459 138 109 120652673 147 1206526606 147 1207203708 174 1207204117 136 1207204459 138 110 1206526272 148 1206526605 148 1207203706 175 1207204115 138 1207204458 139											
107 1206526275 145 1206526609 145 1207203711 171 1207204119 134 1207204462 135 108 1206526274 146 1206526608 146 1207203708 172 1207204119 134 1207204459 138 109 1206526273 147 1206526606 147 1207203708 174 1207204117 136 1207204459 138 110 1206526272 148 1206526605 148 1207203706 175 1207204115 138 1207204458 139											
108 1206526274 146 1206526608 146 1207203708 172 1207204119 134 1207204459 138 109 1206526273 147 1206526606 147 1207203708 174 1207204117 136 1207204459 138 110 1206526272 148 1206526605 148 1207203706 175 1207204115 138 1207204458 139											
109 1206526273 147 1206526606 147 1207203708 174 1207204117 136 1207204459 138 110 1206526272 148 1206526605 148 1207203706 175 1207204115 138 1207204458 139											
110 1206526272 148 1206526605 148 1207203706 175 1207204115 138 1207204458 139											
1111 1206526271 149 1206526603 149 1207203705 177 1207204115 129 1207204457 140											
11 12002001 17 12002000 177 12012010 171 120120113 130 1201204437 140	111	1206526271	149	1206526603	149	1207203705	177	1207204115	138	1207204457	140

1112	1206526271	140	1206526602	140	1207202704	170	1207204114	120	1207204456	L.,,
112	1206526271	149	1206526602	149	1207203704	178	1207204114	139	1207204456	141
113	1206526269	151	1206526601	151	1207203703	179	1207204114	139	1207204456	141
114	1206526267	153	1206526600	153	1207203701	180	1207204113	140	1207204453	144
115	1206526267	153	1206526598	153	1207203700	182	1207204111	142	1207204453	144
116	1206526266	154	1206526596	154	1207203699	184	1207204110	143	1207204452	145
117	1206526264	156	1206526595	156	1207203699	185	1207204110	143	1207204451	146
118	1206526263	157	1206526594	157	1207203698	186	1207204108	145	1207204450	147
119	1206526261	159	1206526593	159	1207203698	187	1207204107	146	1207204449	148
120	1206526261	159	1206526592	159	1207203695	188	1207204105	148	1207204448	149
121	1206526259	161	1206526590	161	1207203694	190	1207204105	148	1207204447	150
122	1206526258	162	1206526589	162	1207203694	191	1207204104	149	1207204446	151
123	1206526257	163	1206526588	163	1207203693	192	1207204104	149	1207204445	152
124	1206526257	163	1206526587	163	1207203693	193	1207204101	152	1207204445	152
125	1206526255	165	1206526586	165	1207203690	194	1207204100	153	1207204442	155
126	1206526254	166	1206526584	166	1207203689	196	1207204100	153	1207204442	155
127	1206526254	166	1206526583	166	1207203689	197	1207204099	154	1207204441	156
128	1206526251	169	1206526582	169	1207203687	198	1207204097	156	1207204440	157
129	1206526251	169	1206526581	169	1207203686	199	1207204096	157	1207204440	157
130	1206526249	171	1206526580	171	1207203685	200	1207204095	158	1207204438	159
131	1206526248	172	1206526579	172	1207203685	201	1207204094	159	1207204437	160
132	1206526247	173	1206526577	173	1207203683	203	1207204093	160	1207204436	161
133	1206526246	174	1206526576	174	1207203682	204	1207204092	161	1207204434	163
134	1206526245	175	1206526575	175	1207203681	205	1207204091	162	1207204434	163
135	1206526244	176	1206526573	176	1207203679	207	1207204091	162	1207204433	164
136	1206526243	177	1206526572	177	1207203678	208	1207204089	164	1207204431	166
137	1206526243	177	1206526571	177	1207203677	209	1207204088	165	1207204431	166
138	1206526241	179	1206526570	179	1207203677	210	1207204088	165	1207204430	167
139	1206526240	180	1206526568	180	1207203675	212	1207204087	166	1207204429	168
140	1206526240	180	1206526567	180	1207203675	213	1207204086	167	1207204428	169
141	1206526237	183	1206526566	183	1207203674	214	1207204084	169	1207204426	171
142	1206526237	183	1206526565	183	1207203673	215	1207204084	169	1207204426	171
143	1206526237	185	1206526563	185	1207203671	217	1207204084	169	1207204425	172
144	1206526234	186	1206526561	186	1207203671	219	1207204081	172	1207204423	174
145	1206526233	187	1206526561	187	1207203670	219	1207204081	172	1207204422	175
146	1206526232	188	1206526559	188	1207203669	221	1207204080	173	1207204422	175
147	1206526231	189	1206526557	189	1207203668	223	1207204030	175	1207204421	176
		190		190						
148	1206526230		1206526557		1207203668	223	1207204077	176	1207204419	178
149	1206526229	191	1206526554	191	1207203665	226	1207204077	176	1207204419	178
150	1206526227	193	1206526553	193	1207203665	227	1207204075	178	1207204418	179
151	1206526227	193	1206526551	193	1207203665	229	1207204074	179	1207204417	180
152	1206526226	194	1206526550	194	1207203662	230	1207204073	180	1207204415	182
153	1206526224	196	1206526548	196	1207203662	232	1207204072	181	1207204413	184
154	1206526223	197	1206526545	197	1207203661	235	1207204071	182	1207204412	185
155	1206526222	198	1206526543	198	1207203658	237	1207204071	182	1207204411	186
156	1206526222	198	1206526542	198	1207203658	238	1207204068	185	1207204410	187
157	1206526219	201	1206526540	201	1207203658	240	1207204068	185	1207204409	188
158	1206526218	202	1206526538	202	1207203655	242	1207204066	187	1207204408	189
159	1206526217	203	1206526537	203	1207203655	243	1207204066	187	1207204406	191
160				204		245				

CompTorrent Mandelbrot 8 Machines												
Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A		
1	1206527068	0	1206527300	0	1207205110	0	1207205468	0	1207206172	0		
2	1206527067	1	1206527300	1	1207205109	0	1207205468	0	1207206171	1		
3	1206527067	1	1206527299	1	1207205108	1	1207205466	2	1207206171	1		
4	1206527066	2	1206527298	2	1207205107	2	1207205465	3	1207206169	3		
5	1206527064	4	1206527296	4	1207205105	4	1207205464	4	1207206169	3		
6	1206527063	5	1206527295	5	1207205104	5	1207205463	5	1207206167	5		
7	1206527061	7	1206527294	7	1207205103	6	1207205462	6	1207206166	6		

l.		I .		l.		۱.		l <u>-</u>		١
8	1206527060	8	1206527294	8	1207205101	6	1207205461	7	1207206165	7
9	1206527059	9	1206527294	9	1207205100	6	1207205458	10	1207206164	8
10	1206527058	10	1206527292	10	1207205098	8	1207205457	11	1207206163	9
11	1206527057	11	1206527292	11	1207205097	8	1207205457	11	1207206162	10
12	1206527056	12	1206527289	12	1207205096	11	1207205456	12	1207206160	12
13	1206527055	13	1206527288	13	1207205095	12	1207205456	12	1207206159	13
14	1206527053	15	1206527286	15	1207205094	14	1207205456	12	1207206158	14
15	1206527052	16	1206527286	16	1207205093	14	1207205454	14	1207206157	15
16	1206527051	17	1206527285	17	1207205092	15	1207205453	15	1207206155	17
17	1206527050	18	1206527284	18	1207205091	16	1207205453	15	1207206154	18
18	1206527049	19	1206527283	19	1207205090	17	1207205451	17	1207206154	18
19	1206527047	21	1206527282	21	1207205089	18	1207205451	17	1207206153	19
20	1206527047	21	1206527281	21	1207205088	19	1207205451	17	1207206151	21
21	1206527046	22	1206527281	22	1207205087	19	1207205448	20	1207206150	22
22	1206527045	23	1206527279	23	1207205085	21	1207205447	21	1207206149	23
23	1206527043	25	1206527278	25	1207205083	22	1207205447	21	1207206148	24
24	1206527043	25	1206527278	25	1207205082	22	1207205444	24	1207206148	24
25	1206527042	26	1206527277	26	1207205082	23	1207205444	24	1207206147	25
26	1206527041	27	1206527276	27	1207205080	24	1207205443	25	1207206146	26
27	1206527040	28	1206527274	28	1207205080	26	1207205441	27	1207206145	27
28	1206527036	32	1206527273	32	1207205079	27	1207205441	27	1207206144	28
29	1206527035	33	1206527272	33	1207205078	28	1207205441	27	1207206143	29
30	1206527034	34	1206527271	34	1207205077	29	1207205439	29	1207206142	30
31	1206527034	34	1206527270	34	1207205075	30	1207205436	32	1207206141	31
32	1206527032	36	1206527269	36	1207205074	31	1207205435	33	1207206140	32
33	1206527031	37	1206527268	37	1207205073	32	1207205434	34	1207206139	33
34	1206527026	42	1206527268	42	1207205071	32	1207205434	34	1207206138	34
35	1206527025	43	1206527268	43	1207205070	32	1207205431	37	1207206136	36
36	1206527025	43	1206527267	43	1207205069	33	1207205431	37	1207206136	36
37	1206527024	44	1206527265	44	1207205068	35	1207205431	37	1207206133	39
38	1206527023	45	1206527264	45	1207205067	36	1207205430	38	1207206132	40
39	1206527020	48	1206527263	48	1207205066	37	1207205429	39	1207206130	42
40	1206527019	49	1206527260	49	1207205066	40	1207205428	40	1207206129	43
41	1206527017	51	1206527259	51	1207205063	41	1207205427	41	1207206129	43
42	1206527016	52	1206527258	52	1207205062	42	1207205426	42	1207206128	44
43	1206527014	54	1206527257	54	1207205061	43	1207205425	43	1207206126	46
44	1206527013	55	1206527255	55	1207205059	45	1207205424	44	1207206125	47
45	1206527009	59	1206527255	59	1207205059	45	1207205422	46	1207206125	47
46	1206527008	60	1206527254	60	1207205058	46	1207205421	47	1207206123	49
47	1206527007	61	1206527251	61	1207205056	49	1207205418	50	1207206122	50
48	1206527004	64	1206527250	64	1207205055	50	1207205418	50	1207206121	51
49	1206527003	65	1206527249	65	1207205054	51	1207205418	50	1207206120	52
50	1206527002	66	1206527249	66	1207205052	51	1207205416	52	1207206120	52
51	1206527001	67	1206527249	67	1207205051	51	1207205415	53	1207206118	54
52	1206527000	68	1206527246	68	1207205050	54	1207205415	53	1207206117	55
53	1206526998	70	1206527244	70	1207205048	56	1207205413	55	1207206114	57
54	1206526997	71	1206527243	71	1207205047	57	1207205412	56	1207206114	58
55	1206526996	72	1206527242	72	1207205046	58	1207205412	56	1207206114	58
56	1206526994	74	1206527242	74	1207205046	58	1207205410	58	1207206113	59
57	1206526993	75	1206527241	75	1207205043	59	1207205410	58	1207206110	62
58	1206526992	76	1206527240	76	1207205042	60	1207205408	60	1207206110	62
59	1206526990	78	1206527238	78	1207205041	62	1207205406	62	1207206109	63
60	1206526988	80	1206527238	80	1207205040	62	1207205406	62	1207206106	66
61	1206526987	81	1206527236	81	1207205039	64	1207205405	63	1207206106	66
62	1206526986	82	1206527234	82	1207205039	66	1207205404	64	1207206105	67
63	1206526985	83	1206527234	83	1207205035	66	1207205403	65	1207206104	68
64	1206526984	84	1206527234	84	1207205034	66	1207205403	65	1207206103	69
65	1206526983	85	1206527233	85	1207205033	67	1207205401	67	1207206102	70
66	1206526981	87	1206527232	87	1207205032	68	1207205400	68	1207206101	71

	12065255	07	12065255	0.7	1207222		1207227		12072255	١.,
67	1206526981	87	1206527232	87	1207205032	68	1207205398	70	1207206101	71
68	1206526980	88	1206527230	88	1207205029	70	1207205398	70	1207206099	73
69	1206526978	90	1206527230	90	1207205027	70	1207205396	72	1207206099	73
70	1206526977	91	1206527229	91	1207205026	71	1207205396	72	1207206097	75
71	1206526975	93	1206527226	93	1207205026	74	1207205395	73	1207206096	76
72	1206526974	94	1206527225	94	1207205025	75	1207205394	74	1207206095	77
73	1206526972	96	1206527224	96	1207205022	76	1207205393	75	1207206094	78
74	1206526972	96	1206527222	96	1207205021	78	1207205393	75	1207206094	78
75	1206526969	99	1206527222	99	1207205020	78	1207205392	76	1207206093	79
76	1206526968	100	1206527222	100	1207205020	78	1207205390	78	1207206093	79
77	1206526968	100	1206527220	100	1207205018	80	1207205389	79	1207206091	81
78	1206526966	102	1206527220	102	1207205017	80	1207205387	81	1207206089	83
79	1206526964	104	1206527217	104	1207205017	83	1207205387	81	1207206089	83
80	1206526963	105	1206527217	105	1207205015	83	1207205386	82	1207206087	85
81	1206526962	106	1206527216	106	1207205015	84	1207205385	83	1207206087	85
82	1206526961	107	1206527216	107	1207205013	84	1207205384	84	1207206085	87
	1206526960	108	1206527215	108	1207205014		1207205384	86	1207206085	87
83						85				
84	1206526960	108	1206527215	108	1207205010	85	1207205381	87	1207206084	88
85	1206526958	110	1206527211	110	1207205010	89	1207205381	87	1207206083	89
86	1206526956	112	1206527210	112	1207205010	90	1207205381	87	1207206082	90
87	1206526956	112	1206527209	112	1207205009	91	1207205377	91	1207206081	91
88	1206526955	113	1206527208	113	1207205008	92	1207205377	91	1207206081	91
89	1206526953	115	1206527208	115	1207205008	92	1207205375	93	1207206079	93
90	1206526952	116	1206527206	116	1207205007	94	1207205375	93	1207206077	95
91	1206526951	117	1206527206	117	1207205004	94	1207205375	93	1207206075	97
92	1206526951	117	1206527205	117	1207205004	95	1207205373	95	1207206074	98
93	1206526950	118	1206527204	118	1207205003	96	1207205371	97	1207206074	98
94	1206526948	120	1206527202	120	1207205003	98	1207205371	97	1207206073	99
95	1206526947	121	1206527202	121	1207205000	98	1207205370	98	1207206072	100
96	1206526946	122	1206527200	122	1207204999	100	1207205370	98	1207206072	100
97	1206526944	124	1206527198	124	1207204998	102	1207205369	99	1207206070	102
98	1206526943	125	1206527196	125	1207204998	104	1207205368	100	1207206070	102
99	1206526942	126	1206527196	126	1207204998	104	1207205366	102	1207206067	105
100	1206526941	127	1206527195	127	1207204995	105	1207205365	103	1207206067	105
101	1206526939	129	1206527195	129	1207204994	105	1207205365	103		106
102	1206526939	129	1206527194	129	1207204994	106		103	1207206066	107
							1207205364		1207206065	
103	1206526937	131	1206527193	131	1207204992	107	1207205363	105	1207206063	109
104	1206526936	132	1206527189	132	1207204990	111	1207205362	106	1207206061	111
105	1206526934	134	1206527189	134	1207204989	111	1207205360	108	1207206061	111
106	1206526933	135	1206527188	135	1207204989	112	1207205358	110	1207206059	113
107	1206526932	136	1206527186	136	1207204986	114	1207205357	111	1207206059	113
108	1206526931	137	1206527186	137	1207204986	114	1207205355	113	1207206057	115
109	1206526930	138	1206527185	138	1207204986	115	1207205354	114	1207206056	116
110	1206526930	138	1206527183	138	1207204985	117	1207205353	115	1207206056	116
111	1206526929	139	1206527183	139	1207204983	117	1207205351	117	1207206055	117
112	1206526928	140	1206527183	140	1207204982	117	1207205350	118	1207206054	118
113	1206526925	143	1206527181	143	1207204981	119	1207205349	119	1207206052	120
114	1206526925	143	1206527181	143	1207204980	119	1207205348	120	1207206050	122
115	1206526924	144	1206527180	144	1207204979	120	1207205348	120	1207206049	123
116	1206526923	145	1206527178	145	1207204978	122	1207205347	121	1207206048	124
117	1206526922	146	1206527177	146	1207204976	123	1207205344	124	1207206047	125
118	1206526921	147	1206527175	147	1207204976	125	1207205344	124	1207206046	126
119	1206526920	148	1206527174	148	1207204975	126	1207205344	126	1207206045	127
120	1206526919	149	1206527174	149	1207204973	129	1207205342	127	1207206045	127
121	1206526915	153	1206527171	153	1207204972	129	1207205339	129	1207206042	130
122	1206526915	153	1206527170	153	1207204971	130	1207205338	130	1207206042	130
123	1206526914	154	1206527169	154	1207204971	131	1207205338	130	1207206041	131
124	1206526913	155	1206527167	155	1207204968	133	1207205335	133	1207206039	133
125	1206526911	157	1206527166	157	1207204966	134	1207205334	134	1207206038	134

126	1206526910	158	1206527166	158	1207204966	134	1207205333	135	1207206037	135
127	1206526909	159	1206527164	159	1207204964	136	1207205333	135	1207206036	136
128	1206526908	160	1206527163	160	1207204963	137	1207205332	136	1207206035	137
129	1206526906	162	1206527162	162	1207204961	138	1207205331	137	1207206033	139
130	1206526906	162	1206527161	162	1207204961	139	1207205330	138	1207206032	140
131	1206526905	163	1206527160	163	1207204960	140	1207205328	140	1207206031	141
132	1206526901	167	1206527159	167	1207204958	141	1207205327	141	1207206030	142
133	1206526900	168	1206527158	168	1207204955	142	1207205327	141	1207206029	143
134	1206526899	169	1206527156	169	1207204954	144	1207205326	142	1207206028	144
135	1206526898	170	1206527156	170	1207204953	144	1207205324	144	1207206027	145
136	1206526896	172	1206527155	172	1207204952	145	1207205322	146	1207206026	146
137	1206526895	173	1206527154	173	1207204951	146	1207205322	146	1207206025	147
138	1206526894	174	1206527154	174	1207204951	146	1207205322	146	1207206024	148
139	1206526892	176	1206527150	176	1207204948	150	1207205321	147	1207206023	149
140	1206526892	176	1206527149	176	1207204948	151	1207205320	148	1207206022	150
141	1206526890	178	1206527147	178	1207204945	153	1207205318	150	1207206021	151
142	1206526886	182	1206527146	182	1207204944	154	1207205317	151	1207206019	153
143	1206526886	182	1206527146	182	1207204943	154	1207205317	151	1207206019	153
144	1206526884	184	1206527145	184	1207204942	155	1207205315	153	1207206018	154
145	1206526883	185	1206527143	185	1207204940	157	1207205315	153	1207206018	154
146	1206526882	186	1206527142	186	1207204940	158	1207205315	153	1207206016	156
147	1206526878	190	1206527142	190	1207204937	158	1207205312	156	1207206016	156
148	1206526876	192	1206527139	192	1207204937	161	1207205311	157	1207206014	158
149	1206526874	194	1206527139	194	1207204936	161	1207205310	158	1207206012	160
150	1206526873	195	1206527138	195	1207204934	162	1207205308	160	1207206012	160
151	1206526870	198	1206527137	198	1207204934	163	1207205308	160	1207206011	161
152	1206526869	199	1206527136	199	1207204931	164	1207205307	161	1207206009	163
153	1206526866	202	1206527135	202	1207204929	165	1207205307	161	1207206009	163
154	1206526864	204	1206527135	204	1207204928	165	1207205305	163	1207206006	166
155	1206526862	206	1206527132	206	1207204927	168	1207205303	165	1207206005	167
156	1206526860	208	1206527131	208	1207204925	169	1207205302	166	1207206004	168
157	1206526858	210	1206527131	210	1207204924	169	1207205300	168	1207206002	170
158	1206526857	211	1206527129	211	1207204923	171	1207205299	169	1207206001	171
159	1206526855	213	1206527127	213	1207204920	173	1207205298	170	1207205998	174
160	1206526853	215	1206527126	215	1207204920	174	1207205297	171	1207205998	174

CompTorrent Mandelbrot 16 Machines												
Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A		
1	1205801809	0	1205802395	0	1205802817	0	1205806273	0	1205806655	0		
2	1205801808	1	1205802394	1	1205802816	1	1205806272	1	1205806654	1		
3	1205801807	2	1205802393	2	1205802814	2	1205806272	1	1205806653	2		
4	1205801807	2	1205802393	2	1205802814	2	1205806271	2	1205806652	3		
5	1205801806	3	1205802389	3	1205802812	6	1205806270	3	1205806651	4		
6	1205801805	4	1205802387	4	1205802809	8	1205806269	4	1205806649	6		
7	1205801803	6	1205802383	6	1205802805	12	1205806268	5	1205806644	11		
8	1205801800	9	1205802380	9	1205802803	15	1205806267	6	1205806643	12		
9	1205801798	11	1205802378	11	1205802802	17	1205806266	7	1205806641	14		
10	1205801797	12	1205802377	12	1205802800	18	1205806265	8	1205806639	16		
11	1205801796	13	1205802377	13	1205802798	18	1205806264	9	1205806638	17		
12	1205801795	14	1205802376	14	1205802796	19	1205806264	9	1205806636	19		
13	1205801792	17	1205802374	17	1205802795	21	1205806259	14	1205806634	21		
14	1205801791	18	1205802373	18	1205802793	22	1205806254	19	1205806633	22		
15	1205801790	19	1205802372	19	1205802791	23	1205806244	29	1205806631	24		
16	1205801789	20	1205802371	20	1205802789	24	1205806243	30	1205806624	31		
17	1205801788	21	1205802369	21	1205802787	26	1205806241	32	1205806622	33		
18	1205801786	23	1205802364	23	1205802787	31	1205806238	35	1205806621	34		
19	1205801785	24	1205802364	24	1205802785	31	1205806238	35	1205806618	37		
20	1205801785	24	1205802362	24	1205802784	33	1205806234	39	1205806615	40		
21	1205801784	25	1205802361	25	1205802783	34	1205806231	42	1205806613	42		

22	1205801783	26	1205802360	26	1205802782	35	1205806229	44	1205806607	48
23	1205801783	36	1205802360	36	1205802782	36	1205806229	47	1205806607	49
24	1205801772	37	1205802357	37	1205802779	38	1205806223	50	1205806606	49
25	1205801768	41	1205802355	41	1205802777	40	1205806220	53	1205806604	51
26	1205801767	42	1205802354	42	1205802776	41	1205806218	55	1205806603	52
27	1205801766	43	1205802352	43	1205802776	43	1205806211	62	1205806601	54
28	1205801765	44	1205802351	44	1205802774	44	1205806210	63	1205806599	56
29	1205801764	45	1205802351	45	1205802771	44	1205806209	64	1205806597	58
30	1205801763	46	1205802350	46	1205802771	45	1205806207	66	1205806592	63
31	1205801763	46	1205802349	46	1205802770	46	1205806206	67	1205806591	64
32	1205801762	47	1205802348	47	1205802769	47	1205806204	69	1205806590	65
33	1205801762	47	1205802346	47	1205802768	49	1205806203	70	1205806588	67
34	1205801761	48	1205802345	48	1205802766	50	1205806202	71	1205806587	68
35	1205801760	49	1205802344	49	1205802765	51	1205806200	73	1205806586	69
36	1205801760	49	1205802342	49	1205802765	53	1205806199	74	1205806584	71
37	1205801758	51	1205802342	51	1205802763	53	1205806198	75	1205806583	72
38	1205801758	51	1205802340	51	1205802762	55	1205806197	76	1205806582	73
39	1205801755	54	1205802339	54	1205802762	56	1205806194	79	1205806582	73
40	1205801755	54	1205802337	54	1205802759	58	1205806191	82	1205806582	73
41	1205801754	55	1205802337	55	1205802758	58	1205806190	83	1205806580	75
42	1205801753	56	1205802336	56	1205802757	59	1205806188	85	1205806578	77
43	1205801752	57	1205802334	57	1205802755	61	1205806188	85	1205806577	78
44	1205801750	59	1205802333	59	1205802754	62	1205806185	88	1205806576	79
45	1205801749	60	1205802332	60	1205802752	63	1205806185	88	1205806575	80
46	1205801749	60	1205802332	60	1205802752	65	1205806183	91	1205806574	81
47	1205801747	62			1205802750		1205806182	92		84
			1205802330	62		65		94	1205806571	
48	1205801747	62	1205802329	62	1205802750	66	1205806179		1205806571	84
49	1205801745	64	1205802328	64	1205802748	67	1205806179	94	1205806569	86
50	1205801745	64	1205802326	64	1205802747	69	1205806177	96	1205806569	86
51	1205801744	65	1205802325	65	1205802746	70	1205806176	97	1205806568	87
52	1205801742	67	1205802323	67	1205802744	72	1205806175	98	1205806566	89
53	1205801742	67	1205802323	67	1205802744	72	1205806173	100	1205806566	89
54	1205801742	67	1205802321	67	1205802741	74	1205806172	101	1205806563	92
55	1205801740	69	1205802321	69	1205802740	74	1205806169	104	1205806563	92
56	1205801740	69	1205802318	69	1205802740	77	1205806168	105	1205806561	94
57	1205801737	72	1205802318	72	1205802738	77	1205806165	108	1205806560	95
58	1205801737	72	1205802317	72	1205802738	78	1205806165	108	1205806558	97
59	1205801737	72	1205802316	72	1205802735	79	1205806162	111	1205806557	98
60	1205801735	74	1205802314	74	1205802735	81	1205806161	112	1205806555	100
61	1205801734	75	1205802313	75	1205802732	82	1205806159	114	1205806553	102
62	1205801732	77	1205802311	77	1205802732	84	1205806159	114	1205806552	103
63	1205801732	77	1205802308	77	1205802731	87	1205806157	116	1205806552	103
64	1205801730	79	1205802307	79	1205802730	88	1205806155	118	1205806551	104
65	1205801729	80	1205802305	80	1205802727	90	1205806153	120	1205806549	106
66	1205801728	81	1205802304	81	1205802727	91	1205806152	121	1205806548	107
67	1205801728	81	1205802302	81	1205802725	93	1205806151	122	1205806547	108
68	1205801726	83	1205802301	83	1205802724	94	1205806150	123	1205806546	109
69	1205801726	83	1205802300	83	1205802723	95	1205806148	125	1205806544	111
70	1205801725	84	1205802300	84	1205802723	96	1205806148	125	1205806542	113
71	1205801725	84	1205802297	84	1205802721	98	1205806146	127	1205806541	114
72	1205801724	85	1205802297	85	1205802720	98	1205806143	130	1205806541	114
73	1205801724	85	1205802295	85	1205802718	100	1205806142	131	1205806539	116
74	1205801724	85	1205802293	85	1205802717	102	1205806141	132	1205806537	118
75	1205801722	87	1205802292	87	1205802715	103	1205806139	134	1205806534	121
76	1205801722	87	1205802292	87	1205802714	103	1205806138	135	1205806533	122
77	1205801720	89	1205802291	89	1205802713	104	1205806137	136	1205806531	124
78	1205801718	91	1205802290	91	1205802711	105	1205806136	137	1205806529	126
79	1205801717	92	1205802288	92	1205802711	107	1205806134	139	1205806529	126
	1205801717	92	1205802286	92	1205802710	109	1205806132	141	1205806528	127

81 82 83	1205801715 1205801715	94	1205802285	94	1205802708	110	1205806132	141	1205806525	
		94	1205802284	94	1205802708	111	1205806130	143	1205806525	130
	1205801713	95	1205802284	95	1205802706	111	1205806130	143	1205806524	131
84	1205801713	96	1205802281	96	1205802704	114	1205806129	144	1205806523	132
85	1205801710		1205802279		1205802704	116	1205806127	146	1205806521	
86	1205801708	101	1205802278	101	1205802702	117	1205806126	147	1205806520	135
87	1205801708	101	1205802277	101	1205802700	118	1205806125	148	1205806518	137
88	1205801707	102	1205802274	102	1205802699	121	1205806122	151	1205806518	137
89	1205801707	102	1205802272	102	1205802698	123	1205806120	153	1205806516	139
90	1205801705	104	1205802271	104	1205802697	124	1205806118	155	1205806515	140
91	1205801705	104	1205802271	104	1205802696	124	1205806117	156	1205806513	142
92	1205801704	105	1205802269	105	1205802693	126	1205806117	156	1205806512	143
93	1205801703	106	1205802267	106	1205802691	128	1205806114	159	1205806510	145
94	1205801703	106	1205802265	106	1205802690	130	1205806114	159	1205806510	145
95	1205801701	108	1205802263	108	1205802690	132	1205806112	161	1205806509	146
96	1205801699	110	1205802262	110	1205802687	133	1205806108	165	1205806507	148
97	1205801699	110	1205802260	110	1205802686	135	1205806107	166	1205806504	151
98	1205801698	111	1205802259	111	1205802684	136	1205806106	167	1205806503	152
99	1205801698	111	1205802258	111	1205802683	137	1205806104	169	1205806503	152
100	1205801696	113	1205802256	113	1205802682	139	1205806103	170	1205806502	153
101	1205801696	113	1205802254	113	1205802681	141	1205806100	173	1205806502	153
102	1205801693	116	1205802252	116	1205802679	143	1205806100	173	1205806500	155
103	1205801692	117	1205802251	117	1205802677	144	1205806098	175	1205806498	157
104	1205801690	119	1205802250	119	1205802676	145	1205806097	176	1205806498	157
105	1205801689	120	1205802248	120	1205802676	147	1205806096	177	1205806496	159
106	1205801689	120	1205802246	120	1205802674	149	1205806094	179	1205806495	160
107	1205801687	122	1205802245	122	1205802672	150	1205806093	180	1205806494	161
108	1205801685	124	1205802244	124	1205802672	151	1205806090	183	1205806491	164
109	1205801684	125	1205802243	125	1205802670	152	1205806088	185	1205806491	164
110	1205801683	126	1205802242	126	1205802670	153	1205806087	186	1205806490	165
111	1205801682	127	1205802240	127	1205802667	155	1205806087	186	1205806488	167
112	1205801680	129	1205802238	129	1205802664	157	1205806085	188	1205806486	169
113	1205801679	130	1205802238	130	1205802664	157	1205806084	189	1205806484	171
114	1205801676	133	1205802235	133	1205802663	160	1205806081	192	1205806484	171
115	1205801675	134	1205802233	134	1205802661	162	1205806080	193	1205806482	173
116	1205801673	136	1205802233	136	1205802659	162	1205806078	195	1205806482	173
117	1205801673	136	1205802231	136	1205802658	164	1205806077	196	1205806479	176
118	1205801671	138	1205802230	138	1205802656	165	1205806076	197	1205806479	176
119	1205801670	139	1205802228	139	1205802655	167	1205806075	198	1205806476	179
120	1205801670	139	1205802227	139	1205802654	168	1205806073	200	1205806476	179
121	1205801668	141	1205802227	141	1205802653	168	1205806071	202	1205806473	182
122	1205801667	142	1205802225	142	1205802652	170	1205806070	203	1205806471	184
123	1205801665	144	1205802223	144	1205802651	172	1205806077	206	1205806471	184
124	1205801665	144	1205802223	144	1205802650	172	1205806067	206	1205806470	185
125	1205801664	145	1205802223	145	1205802648	175	1205806067	208	1205806468	187
126	1205801661	148	1205802220	148	1205802648	175	1205806063	210	1205806468	187
127	1205801661	148	1205802219	148	1205802646	176	1205806063	210	1205806466	189
128	1205801660	149	1205802216	149	1205802644	179	1205806061	212	1205806466	189
129	1205801659	150	1205802214	150	1205802641	181	1205806060	213	1205806463	192
130	1205801657	152	1205802213	152	1205802640	182	1205806058	215	1205806462	193
131	1205801657	152	1205802212	152	1205802639	183	1205806056	217	1205806461	194
132	1205801656	153	1205802210	153	1205802637	185	1205806055	218	1205806459	196
133	1205801655	154	1205802206	154	1205802637	189	1205806053	220	1205806458	197
134	1205801652	157	1205802204	157	1205802635	191	1205806053	220	1205806456	199
135	1205801652	157	1205802204	157	1205802633	191	1205806051	222	1205806455	200
136	1205801652	157	1205802202	157	1205802632	193	1205806050	223	1205806453	202
137	1205801650	159	1205802202	159	1205802631	193	1205806049	224	1205806453	202
138	1205801648	161	1205802200	161	1205802629	195	1205806048	225	1205806450	205
				161	1205802627	196	1205806047	226	1205806450	205

140	1205801648	161	1205802196	161	1205802626	199	1205806045	228	1205806447	208
141	1205801645	164	1205802196	164	1205802624	199	1205806044	229	1205806444	211
142	1205801645	164	1205802193	164	1205802623	202	1205806041	232	1205806444	211
143	1205801644	165	1205802192	165	1205802621	203	1205806039	234	1205806444	211
144	1205801640	169	1205802190	169	1205802621	205	1205806037	236	1205806439	216
145	1205801640	169	1205802189	169	1205802619	206	1205806035	238	1205806438	217
146	1205801640	169	1205802185	169	1205802618	210	1205806034	239	1205806437	218
147	1205801637	172	1205802185	172	1205802617	210	1205806031	242	1205806437	218
148	1205801637	172	1205802184	172	1205802617	211	1205806029	244	1205806435	220
149	1205801635	174	1205802183	174	1205802614	212	1205806029	244	1205806433	222
150	1205801633	176	1205802178	176	1205802613	217	1205806025	248	1205806432	223
151	1205801632	177	1205802174	177	1205802612	221	1205806024	249	1205806429	226
152	1205801631	178	1205802174	178	1205802611	221	1205806021	252	1205806429	226
153	1205801630	179	1205802173	179	1205802608	222	1205806021	252	1205806428	227
154	1205801627	182	1205802169	182	1205802605	226	1205806017	256	1205806426	229
155	1205801627	182	1205802168	182	1205802603	227	1205806016	257	1205806423	232
156	1205801623	186	1205802167	186	1205802602	228	1205806014	259	1205806422	233
157	1205801623	186	1205802163	186	1205802598	232	1205806010	263	1205806421	234
158	1205801622	187	1205802163	187	1205802594	232	1205806010	263	1205806416	239
159	1205801613	196	1205802159	196	1205802594	236	1205806006	267	1205806412	243
160	1205801613	196	1205802157	196	1205802589	238	1205806006	267	1205806412	243

No Work

Condor

Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A
1	1201134141	0	1201134710	0	1201135282	0	1201135733	0	1201136141	0
2	1201134145	4	1201134714	4	1201135286	4	1201135737	4	1201136146	5
3	1201134149	8	1201134718	8	1201135290	8	1201135740	7	1201136150	9
4	1201134153	12	1201134722	12	1201135294	12	1201135745	12	1201136154	13
5	1201134156	15	1201134726	16	1201135297	15	1201135749	16	1201136158	17
6	1201134160	19	1201134729	19	1201135302	20	1201135753	20	1201136161	20
7	1201134163	22	1201134733	23	1201135305	23	1201135756	23	1201136165	24
8	1201134167	26	1201134736	26	1201135309	27	1201135760	27	1201136168	27
9	1201134170	29	1201134740	30	1201135312	30	1201135763	30	1201136172	31
10	1201134174	33	1201134743	33	1201135316	34	1201135767	34	1201136175	34
11	1201134177	36	1201134746	36	1201135319	37	1201135771	38	1201136178	37
12	1201134181	40	1201134750	40	1201135322	40	1201135774	41	1201136182	41
13	1201134184	43	1201134753	43	1201135325	43	1201135778	45	1201136185	44
14	1201134187	46	1201134757	47	1201135329	47	1201135781	48	1201136189	48
15	1201134191	50	1201134760	50	1201135332	50	1201135784	51	1201136192	51
16	1201134194	53	1201134764	54	1201135336	54	1201135788	55	1201136196	55
17	1201134198	57	1201134767	57	1201135339	57	1201135791	58	1201136199	58
18	1201134201	60	1201134771	61	1201135343	61	1201135795	62	1201136203	62
19	1201134205	64	1201134774	64	1201135347	65	1201135798	65	1201136206	65
20	1201134208	67	1201134778	68	1201135350	68	1201135802	69	1201136210	69
21	1201134212	71	1201134781	71	1201135354	72	1201135806	73	1201136214	73
22	1201134215	74	1201134785	75	1201135357	75	1201135809	76	1201136217	76
23	1201134218	77	1201134788	78	1201135361	79	1201135813	80	1201136221	80
24	1201134222	81	1201134792	82	1201135364	82	1201135816	83	1201136224	83
25	1201134225	84	1201134795	85	1201135368	86	1201135819	86	1201136228	87
26	1201134229	88	1201134798	88	1201135371	89	1201135822	89	1201136231	90
27	1201134232	91	1201134802	92	1201135375	93	1201135826	93	1201136235	94
28	1201134236	95	1201134805	95	1201135378	96	1201135830	97	1201136238	97
29	1201134239	98	1201134809	99	1201135381	99	1201135833	100	1201136242	101
30	1201134243	102	1201134812	102	1201135385	103	1201135837	104	1201136245	104

	1	1	l	1	1	1	l	1	1	1
31	1201134246	105	1201134816	106	1201135388	106	1201135841	108	1201136248	107
32	1201134249	108	1201134819	109	1201135392	110	1201135844	111	1201136252	111
33	1201134253	112	1201134823	113	1201135395	113	1201135847	114	1201136255	114
34	1201134256	115	1201134827	117	1201135399	117	1201135851	118	1201136259	118
35	1201134260	119	1201134831	121	1201135402	120	1201135854	121	1201136262	121
36	1201134263	122	1201134834	124	1201135406	124	1201135858	125	1201136266	125
37	1201134267	126	1201134838	128	1201135409	127	1201135861	128	1201136269	128
38	1201134270	129	1201134842	132	1201135413	131	1201135865	132	1201136273	132
39	1201134274	133	1201134845	135	1201135416	134	1201135868	135	1201136276	135
40	1201134277	136	1201134849	139	1201135420	138	1201135871	138	1201136280	139
41	1201134281	140	1201134852	142	1201135423	141	1201135874	141	1201136283	142
42	1201134284	143	1201134856	146	1201135427	145	1201135878	145	1201136287	146
43	1201134288	147	1201134859	149	1201135430	148	1201135881	148	1201136290	149
44	1201134291	150	1201134863	153	1201135434	152	1201135884	151	1201136294	153
45	1201134294	153	1201134867	157	1201135437	155	1201135888	155	1201136297	156
46	1201134297	156	1201134870	160	1201135441	159	1201135892	159	1201136300	159
47	1201134300	159	1201134874	164	1201135445	163	1201135896	163	1201136304	163
48	1201134304	163	1201134877	167	1201135448	166	1201135899	166	1201136307	166
49	1201134307	166	1201134881	171	1201135451	169	1201135902	169	1201136311	170
50	1201134311	170	1201134884	174	1201135455	173	1201135906	173	1201136314	173
51	1201134314	173	1201134888	178	1201135458	176	1201135909	176	1201136318	177
52	1201134317	176	1201134891	181	1201135462	180	1201135913	180	1201136321	180
53	1201134321	180	1201134895	185	1201135465	183	1201135916	183	1201136325	184
54	1201134324	183	1201134898	188	1201135469	187	1201135920	187	1201136328	187
55	1201134328	187	1201134902	192	1201135472	190	1201135923	190	1201136332	191
56	1201134331	190	1201134905	195	1201135476	194	1201135927	194	1201136335	194
57	1201134335	194	1201134908	198	1201135479	197	1201135930	197	1201136338	197
58	1201134338	197	1201134912	202	1201135483	201	1201135934	201	1201136342	201
59	1201134342	201	1201134916	206	1201135486	204	1201135937	204	1201136345	204
60	1201134345	204	1201134919	209	1201135490	208	1201135940	207	1201136349	208
61	1201134349	208	1201134923	213	1201135493	211	1201135944	211	1201136352	211
62	1201134352	211	1201134926	216	1201135496	214	1201135947	214	1201136356	215
63	1201134356	215	1201134930	220	1201135500	218	1201135951	218	1201136359	218
64	1201134359	218	1201134933	223	1201135503	221	1201135954	221	1201136363	222
65	1201134363	222	1201134937	227	1201135507	225	1201135958	225	1201136367	226
66	1201134366	225	1201134940	230	1201135510	228	1201135961	228	1201136370	229
67	1201134370	229	1201134944	234	1201135514	232	1201135964	231	1201136373	232
68	1201134373	232	1201134947	237	1201135517	235	1201135968	235	1201136377	236
69	1201134377	236	1201134951	241	1201135521	239	1201135971	238	1201136381	240
70	1201134381	240	1201134954	244	1201135524	242	1201135975	242	1201136384	243
71	1201134384	243	1201134958	248	1201135528	246	1201135978	245	1201136388	247
72	1201134387	246	1201134961	251	1201135531	249	1201135982	249	1201136391	250
73	1201134391	250	1201134964	254	1201135535	253	1201135985	252	1201136395	254
74	1201134394	253	1201134968	258	1201135538	256	1201135989	256	1201136398	257
75	1201134398	257	1201134971	261	1201135536	259	1201135989	259	1201136398	261
76	1201134398	260	1201134975	265	1201135545	263	1201135995	262	1201136405	264
77	1201134401	264	1201134973	268	1201135548	266	1201135999	266	1201136408	267
78	1201134403	267	1201134978	272	1201135548	270	1201133999	269	1201136408	271
79	1201134408	271	1201134982	275	1201135555	273	1201136002	273	1201136412	274
80	1201134412	274	1201134985	279	1201135559	277	1201136006	276	1201136415	274
81	1201134418	274	1201134989	282	1201135562	280	1201136009	280	1201136419	281
82			1201134992		1201135565	283	1201136013			
83	1201134422	281	1201134995	285				283	1201136426	285
	1201134425	284		289	1201135569	287	1201136020		1201136429	
84	1201134429	288	1201135002	292	1201135572	290	1201136023	290	1201136432	291
85	1201134432	291	1201135006	296	1201135576	294	1201136027	294	1201136436	295
86	1201134436	295	1201135009	299	1201135579	297	1201136030	297	1201136439	298
87	1201134439	298	1201135012	302	1201135583	301	1201136033	300	1201136443	302
88	1201134443	302	1201135016	306	1201135586	304	1201136037	304	1201136446	305
89	1201134446	305	1201135019	309	1201135589	307	1201136041	308	1201136450	309

90	1201134449	308	1201135023	313	1201135592	310	1201136044	311	1201136454	313
91	1201134453	312	1201135026	316	1201135595	313	1201136048	315	1201136457	316
92	1201134456	315	1201135030	320	1201135599	317	1201136051	318	1201136460	319
93	1201134460	319	1201135033	323	1201135602	320	1201136054	321	1201136464	323
94	1201134463	322	1201135037	327	1201135606	324	1201136058	325	1201136467	326
95	1201134467	326	1201135040	330	1201135609	327	1201136061	328	1201136471	330
96	1201134470	329	1201135044	334	1201135613	331	1201136065	332	1201136474	333
97	1201134474	333	1201135047	337	1201135616	334	1201136068	335	1201136477	336
98	1201134477	336	1201135051	341	1201135620	338	1201136072	339	1201136481	340
99	1201134480	339	1201135054	344	1201135624	342	1201136075	342	1201136484	343
100	1201134484	343	1201135058	348	1201135627	345	1201136079	346	1201136488	347

Condo	r No Work 2	2 Machine	es							
Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A
1	1201165398	0	1201165767	0	1201166141	0	1201166469	0	1201166821	0
2	1201165404	6	1201165771	4	1201166145	4	1201166475	6	1201166826	5
3	1201165408	10	1201165774	7	1201166149	8	1201166479	10	1201166831	10
4	1201165409	11	1201165777	10	1201166152	11	1201166483	14	1201166836	15
5	1201165411	13	1201165782	15	1201166155	14	1201166488	19	1201166842	21
6	1201165414	16	1201165785	18	1201166159	18	1201166490	21	1201166843	22
7	1201165415	17	1201165787	20	1201166160	19	1201166494	25	1201166847	26
8	1201165418	20	1201165790	23	1201166162	21	1201166496	27	1201166849	28
9	1201165419	21	1201165791	24	1201166164	23	1201166501	32	1201166852	31
10	1201165422	24	1201165794	27	1201166167	26	1201166501	32	1201166855	34
11	1201165423	25	1201165795	28	1201166169	28	1201166505	36	1201166859	38
12	1201165426	28	1201165797	30	1201166172	31	1201166508	39	1201166861	40
13	1201165427	29	1201165799	32	1201166173	32	1201166511	42	1201166863	42
14	1201165430	32	1201165801	34	1201166176	35	1201166512	43	1201166864	43
15	1201165431	33	1201165804	37	1201166177	36	1201166515	46	1201166867	46
16	1201165434	36	1201165806	39	1201166180	39	1201166516	47	1201166868	47
17	1201165435	37	1201165809	42	1201166181	40	1201166519	50	1201166870	49
18	1201165437	39	1201165810	43	1201166184	43	1201166520	51	1201166872	51
19	1201165439	41	1201165813	46	1201166185	44	1201166523	54	1201166874	53
20	1201165442	44	1201165814	47	1201166187	46	1201166524	55	1201166876	55
21	1201165443	45	1201165817	50	1201166189	48	1201166527	58	1201166878	57
22	1201165445	47	1201165818	51	1201166192	51	1201166528	59	1201166880	59
23	1201165447	49	1201165820	53	1201166193	52	1201166531	62	1201166882	61
24	1201165450	52	1201165822	55	1201166195	54	1201166532	63	1201166884	63
25	1201165451	53	1201165824	57	1201166197	56	1201166535	66	1201166887	66
26	1201165454	56	1201165826	59	1201166199	58	1201166536	67	1201166888	67
27	1201165455	57	1201165828	61	1201166202	61	1201166538	69	1201166891	70
28	1201165457	59	1201165831	64	1201166203	62	1201166540	71	1201166892	71
29	1201165459	61	1201165832	65	1201166206	65	1201166542	73	1201166895	74
30	1201165461	63	1201165835	68	1201166207	66	1201166544	75	1201166896	75
31	1201165464	66	1201165836	69	1201166210	69	1201166546	77	1201166899	78
32	1201165467	69	1201165838	71	1201166212	71	1201166548	79	1201166900	79
33	1201165468	70	1201165840	73	1201166215	74	1201166551	82	1201166902	81
34	1201165471	73	1201165842	75	1201166216	75	1201166553	84	1201166905	84
35	1201165472	74	1201165844	77	1201166219	78	1201166556	87	1201166907	86
36	1201165474	76	1201165846	79	1201166220	79	1201166557	88	1201166909	88
37	1201165477	79	1201165849	82	1201166222	81	1201166560	91	1201166911	90
38	1201165479	81	1201165850	83	1201166224	83	1201166561	92	1201166913	92
39	1201165482	84	1201165853	86	1201166227	86	1201166564	95	1201166915	94
40	1201165483	85	1201165854	87	1201166228	87	1201166565	96	1201166917	96
41	1201165486	88	1201165857	90	1201166231	90	1201166568	99	1201166919	98
42	1201165487	89	1201165858	91	1201166232	91	1201166569	100	1201166921	100
43	1201165490	92	1201165861	94	1201166235	94	1201166572	103	1201166923	102
44	1201165491	93	1201165862	95	1201166236	95	1201166573	104	1201166926	105
45	1201165494	96	1201165864	97	1201166238	97	1201166576	107	1201166928	107

46	1201165495	97	1201165867	100	1201166240	99	1201166577	108	1201166931	110
47	1201165498	100	1201165868	101	1201166242	101	1201166580	111	1201166932	111
48	1201165499	101	1201165870	103	1201166245	104	1201166581	112	1201166935	114
49	1201165502	104	1201165873	106	1201166246	105	1201166584	115	1201166936	115
50	1201165503	105	1201165875	108	1201166249	108	1201166585	116	1201166939	118
51	1201165505	107	1201165877	110	1201166250	109	1201166588	119	1201166940	119
52	1201165508	110	1201165880	113	1201166253	112	1201166589	120	1201166942	121
53	1201165509	111	1201165881	114	1201166254	113	1201166592	123	1201166944	123
54	1201165511	113	1201165883	116	1201166257	116	1201166593	124	1201166946	125
55	1201165514	116	1201165885	118	1201166258	117	1201166596	127	1201166949	128
56	1201165516	118	1201165888	121	1201166261	120	1201166597	128	1201166950	129
57	1201165518	120	1201165889	121	1201166262	120	1201166597	131	1201166953	132
58	1201165521	123	1201165892	125	1201166264	123	1201166601	132	1201166954	133
59	1201165522	123	1201165892	123	1201166264	125	1201166604	135	1201166954	136
		127		129						
60	1201165525	127	1201165896	130	1201166268	127	1201166605	136	1201166958	137
61	1201165526		1201165897			130	1201166608		1201166961	
62	1201165528	130	1201165900	133	1201166273	132	1201166609	140	1201166962	141
63	1201165530	132	1201165901	134	1201166276	135	1201166612	143	1201166964	143
64	1201165532	134	1201165903	136	1201166278	137	1201166613	144	1201166966	145
65	1201165534	136	1201165906	139	1201166281	140	1201166615	146	1201166969	148
66	1201165536	138	1201165908	141	1201166284	143	1201166618	149	1201166970	149
67	1201165539	141	1201165911	144	1201166285	144	1201166620	151	1201166973	152
68	1201165540	142	1201165912	145	1201166287	146	1201166623	154	1201166974	153
69	1201165542	144	1201165914	147	1201166289	148	1201166625	156	1201166976	155
70	1201165544	146	1201165916	149	1201166292	151	1201166627	158	1201166978	157
71	1201165547	149	1201165918	151	1201166293	152	1201166629	160	1201166980	159
72	1201165548	150	1201165920	153	1201166296	155	1201166632	163	1201166983	162
73	1201165550	152	1201165922	155	1201166297	156	1201166633	164	1201166984	163
74	1201165552	154	1201165925	158	1201166299	158	1201166636	167	1201166986	165
75	1201165554	156	1201165927	160	1201166301	160	1201166637	168	1201166988	167
76	1201165557	159	1201165930	163	1201166303	162	1201166639	170	1201166991	170
77	1201165559	161	1201165932	165	1201166306	165	1201166641	172	1201166992	171
78	1201165561	163	1201165935	168	1201166307	166	1201166643	174	1201166994	173
79	1201165563	165	1201165938	171	1201166310	169	1201166645	176	1201166996	175
80	1201165565	167	1201165939	172	1201166311	170	1201166647	178	1201166998	177
81	1201165567	169	1201165942	175	1201166314	173	1201166650	181	1201167000	179
82	1201165570	172	1201165943	176	1201166315	174	1201166651	182	1201167002	181
83	1201165571	173	1201165946	179	1201166317	176	1201166654	185	1201167005	184
84	1201165574	176	1201165947	180	1201166320	179	1201166655	186	1201167007	186
85	1201165575	177	1201165950	183	1201166322	181	1201166658	189	1201167010	189
86	1201165578	180	1201165951	184	1201166325	184	1201166659	190	1201167012	191
87	1201165579	181	1201165953	186	1201166326	185	1201166662	193	1201167014	193
88	1201165582	184	1201165955	188	1201166329	188	1201166663	194	1201167017	196
89	1201165583	185	1201165957	190	1201166330	189	1201166666	197	1201167018	197
90	1201165586	188	1201165960	193	1201166333	192	1201166667	198	1201167020	199
91	1201165587	189	1201165961	194	1201166334	193	1201166669	200	1201167022	201
92	1201165590	192	1201165964	197	1201166336	195	1201166671	202	1201167024	203
93	1201165591	193	1201165965	198	1201166338	197	1201166673	204	1201167027	206
94	1201165594	196	1201165968	201	1201166341	200	1201166675	206	1201167028	207
95	1201165595	197	1201165969	202	1201166342	201	1201166678	209	1201167031	210
96	1201165598	200	1201165971	204	1201166345	204	1201166679	210	1201167032	211
97	1201165599	201	1201165973	206	1201166346	205	1201166681	212	1201167035	214
98	1201165601	203	1201165976	209	1201166349	208	1201166684	215	1201167036	215
99	1201165604	206	1201165977	210	1201166350	209	1201166686	217	1201167038	217
100	1201165605	207	1201165980	213	1201166352	211	1201166689	220	1201167040	219

Condor N	Condor No Work 4 Machines											
Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A		
1	1201166821	0	1201167536	0	1201167896	0	1201168212	0	1201168545	0		

2	1201166826	5	1201167541	5	1201167902	6	1201168216	4	1201168550	5
3	1201166831	10	1201167547	11	1201167907	11	1201168222	10	1201168555	10
4	1201166836	15	1201167554	18	1201167913	17	1201168227	15	1201168560	15
5	1201166842	21	1201167556	20	1201167914	18	1201168232	20	1201168565	20
6	1201166843	22	1201167557	21	1201167916	20	1201168234	22	1201168569	24
7	1201166847	26	1201167560	24	1201167919	23	1201168236	24	1201168569	24
8	1201166849	28	1201167562	26	1201167920	24	1201168238	26	1201168571	26
9	1201166852	31	1201167564	28	1201167922	26	1201168240	28	1201168574	29
10	1201166855	34	1201167567	31	1201167926	30	1201168242	30	1201168575	30
11	1201166859	38	1201167568	32	1201167926	30	1201168245	33	1201168577	32
12	1201166861	40	1201167570	34	1201167929	33	1201168246	34	1201168580	35
13	1201166863	42	1201167573	37	1201167930	34	1201168249	37	1201168581	36
14	1201166864	43	1201167576	40	1201167932	36	1201168250	38	1201168583	38
15	1201166867	46	1201167576	40	1201167935	39	1201168253	41	1201168585	40
16	1201166868	47	1201167579	43	1201167936	40	1201168254	42	1201168588	43
17	1201166870	49	1201167580	44	1201167938	42	1201168254	43	1201168589	44
	1201166872	51		45	1201167939	43	1201168255	45	1201168589	44
18			1201167581							
19	1201166874	53	1201167582	46	1201167940	44	1201168258	46	1201168592	47
20	1201166876	55	1201167586	50	1201167943	47	1201168260	48	1201168593	48
21	1201166878	57	1201167587	51	1201167944	48	1201168262	50	1201168596	51
22	1201166880	59	1201167588	52	1201167947	51	1201168264	52	1201168597	52
23	1201166882	61	1201167590	54	1201167948	52	1201168266	54	1201168600	55
24	1201166884	63	1201167593	57	1201167951	55	1201168269	57	1201168601	56
25	1201166887	66	1201167594	58	1201167952	56	1201168270	58	1201168604	59
26	1201166888	67	1201167597	61	1201167954	58	1201168273	61	1201168605	60
27	1201166891	70	1201167598	62	1201167956	60	1201168276	64	1201168608	63
28	1201166892	71	1201167600	64	1201167958	62	1201168277	65	1201168609	64
29	1201166895	74	1201167603	67	1201167960	64	1201168279	67	1201168612	67
30	1201166896	75	1201167605	69	1201167962	66	1201168282	70	1201168613	68
31	1201166899	78	1201167608	72	1201167964	68	1201168284	72	1201168615	70
32	1201166900	79	1201167609	73	1201167966	70	1201168286	74	1201168617	72
33	1201166902	81	1201167611	75	1201167969	73	1201168289	77	1201168620	75
34	1201166905	84	1201167613	77	1201167970	74	1201168290	78	1201168621	76
35	1201166907	86	1201167616	80	1201167972	76	1201168293	81	1201168624	79
36	1201166909	88	1201167618	82	1201167974	78	1201168294	82	1201168625	80
37	1201166911	90	1201167621	85	1201167976	80	1201168296	84	1201168628	83
38	1201166913	92	1201167623	87	1201167979	83	1201168298	86	1201168629	84
39	1201166915	94	1201167626	90	1201167980	84	1201168301	89	1201168631	86
40	1201166917	96	1201167629	93	1201167982	86	1201168302	90	1201168633	88
41	1201166919	98	1201167630	94	1201167985	89	1201168304	92	1201168635	90
42	1201166921	100	1201167633	97	1201167987	91	1201168306	94	1201168638	93
43	1201166923	102	1201167634	98	1201167990	94	1201168308	96	1201168641	96
44	1201166926	105	1201167637	101	1201167991	95	1201168311	99	1201168642	97
45	1201166928	107	1201167638	102	1201167994	98	1201168313	101	1201168644	99
46	1201166931	110	1201167641	105	1201167995	99	1201168316	104	1201168646	101
47	1201166932	111	1201167642	106	1201167998	102	1201168318	106	1201168648	103
48	1201166935	114	1201167645	109	1201167999	103	1201168321	109	1201168651	106
49	1201166936	115	1201167646	110	1201168002	106	1201168322	110	1201168653	108
50	1201166939	118	1201167649	113	1201168003	107	1201168324	112	1201168656	111
51	1201166940	119	1201167650	114	1201168005	109	1201168326	114	1201168657	112
52	1201166942	121	1201167652	116	1201168007	111	1201168328	116	1201168660	115
53	1201166944	123	1201167654	118	1201168010	114	1201168331	119	1201168661	116
54	1201166946	125	1201167656	120	1201168011	115	1201168332	120	1201168663	118
55	1201166949	128	1201167659	123	1201168013	117	1201168334	122	1201168665	120
56	1201166950	129	1201167662	126	1201168015	119	1201168334	124	1201168667	120
57	1201166950	132	1201167663	127	1201168013	121	1201168338	124	1201168669	124
58	1201166954	133	1201167666	130	1201168017	124	1201168338	128	1201168669	124
59	1201166954	136	1201167666	130	1201168020	124	1201168340	131	1201168671	126
60	1201166958	137	1201167669	133	1201168024	128	1201168344	132	1201168675	130

61	1201166961	140	1201167672	136	1201168025	129	1201168346	134	1201168678	133
62	1201166962	141	1201167673	137	1201168028	132	1201168348	136	1201168679	134
63	1201166964	143	1201167675	139	1201168029	133	1201168350	138	1201168682	137
64	1201166966	145	1201167677	141	1201168031	135	1201168353	141	1201168683	138
65	1201166969	148	1201167680	144	1201168033	137	1201168354	142	1201168685	140
66	1201166970	149	1201167681	145	1201168035	139	1201168357	145	1201168688	143
67	1201166973	152	1201167683	147	1201168039	143	1201168359	147	1201168691	146
68	1201166974	153	1201167685	149	1201168039	143	1201168361	149	1201168692	147
69	1201166976	155	1201167688	152	1201168042	146	1201168362	150	1201168695	150
70	1201166978	157	1201167689	153	1201168043	147	1201168365	153	1201168696	151
71	1201166980	159	1201167692	156	1201168046	150	1201168366	154	1201168698	153
72	1201166983	162	1201167693	157	1201168047	151	1201168369	157	1201168700	155
73	1201166984	163	1201167696	160	1201168050	154	1201168370	158	1201168702	157
74	1201166986	165	1201167697	161	1201168051	155	1201168373	161	1201168705	160
75	1201166988	167	1201167699	163	1201168054	158	1201168374	162	1201168706	161
76	1201166991	170	1201167701	165	1201168055	159	1201168376	164	1201168709	164
77	1201166992	171	1201167703	167	1201168057	161	1201168378	166	1201168710	165
78	1201166994	173	1201167706	170	1201168059	163	1201168380	168	1201168712	167
79	1201166996	175	1201167707	171	1201168061	165	1201168383	171	1201168714	169
80	1201166998	177	1201167710	174	1201168064	168	1201168384	172	1201168716	171
81	1201167000	179	1201167711	175	1201168065	169	1201168387	175	1201168718	173
82	1201167002	181	1201167713	177	1201168067	171	1201168388	176	1201168720	175
83	1201167005	184	1201167716	180	1201168069	173	1201168390	178	1201168723	178
84	1201167007	186	1201167717	181	1201168072	176	1201168392	180	1201168724	179
85	1201167010	189	1201167720	184	1201168073	177	1201168395	183	1201168727	182
86	1201167012	191	1201167721	185	1201168076	180	1201168396	184	1201168728	183
87	1201167014	193	1201167724	188	1201168077	181	1201168398	186	1201168731	186
88	1201167017	196	1201167725	189	1201168079	183	1201168401	189	1201168732	187
89	1201167018	197	1201167728	192	1201168082	186	1201168403	191	1201168735	190
90	1201167020	199	1201167729	193	1201168084	188	1201168406	194	1201168736	191
91	1201167022	201	1201167731	195	1201168087	191	1201168407	195	1201168738	193
92	1201167024	203	1201167733	197	1201168088	192	1201168410	198	1201168741	196
93	1201167027	206	1201167735	199	1201168091	195	1201168411	199	1201168742	197
94	1201167028	207	1201167738	202	1201168092	196	1201168413	201	1201168745	200
95	1201167031	210	1201167739	203	1201168095	199	1201168416	204	1201168746	201
96	1201167032	211	1201167742	206	1201168096	200	1201168418	206	1201168749	204
97	1201167035	214	1201167743	207	1201168099	203	1201168421	209	1201168750	205
98	1201167036	215	1201167745	209	1201168100	204	1201168424	212	1201168753	208
99	1201167038	217	1201167747	211	1201168103	207	1201168425	213	1201168754	209
100	1201167040	219	1201167750	214	1201168104	208	1201168428	216	1201168757	212

Condor	No Work 8	Machine	S							
Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A
1	1201168963	0	1201169290	0	1201169594	0	1201169929	0	1201170236	0
2	1201168968	5	1201169292	2	1201169597	3	1201169935	6	1201170240	4
3	1201168973	10	1201169295	5	1201169602	8	1201169940	11	1201170242	6
4	1201168978	15	1201169297	7	1201169605	11	1201169944	15	1201170244	8
5	1201168981	18	1201169300	10	1201169609	15	1201169947	18	1201170246	10
6	1201168985	22	1201169302	12	1201169612	18	1201169949	20	1201170247	11
7	1201168986	23	1201169305	15	1201169615	21	1201169951	22	1201170249	13
8	1201168989	26	1201169309	19	1201169616	22	1201169953	24	1201170251	15
9	1201168990	27	1201169309	19	1201169619	25	1201169955	26	1201170253	17
10	1201168992	29	1201169313	23	1201169620	26	1201169957	28	1201170256	20
11	1201168995	32	1201169315	25	1201169623	29	1201169959	30	1201170258	22
12	1201168996	33	1201169316	26	1201169624	30	1201169962	33	1201170260	24
13	1201168998	35	1201169318	28	1201169627	33	1201169963	34	1201170261	25
14	1201169000	37	1201169320	30	1201169628	34	1201169965	36	1201170263	27
15	1201169003	40	1201169323	33	1201169630	36	1201169968	39	1201170267	31
16	1201169005	42	1201169324	34	1201169632	38	1201169969	40	1201170268	32

18	1201169006	43	1201169329	39	1201169636	42	1201169972	43	1201170278	42
19	1201169009	46	1201169332	42	1201169638	44	1201169974	45	1201170278	42
20	1201169010	47	1201169333	43	1201169640	46	1201169975	46	1201170280	44
21	1201169013	50	1201169334	44	1201169642	48	1201169978	49	1201170282	46
22	1201169014	51	1201169336	46	1201169644	50	1201169979	50	1201170285	49
23	1201169017	54	1201169337	47	1201169646	52	1201169982	53	1201170288	52
24	1201169018	55	1201169339	49	1201169648	54	1201169983	54	1201170289	53
25	1201169021	58	1201169341	51	1201169651	57	1201169986	57	1201170292	56
26	1201169022	59	1201169343	53	1201169653	59	1201169987	58	1201170293	57
27	1201169025	62	1201169346	56	1201169656	62	1201169989	60	1201170294	58
28	1201169026	63	1201169347	57	1201169658	64	1201169991	62	1201170297	61
29	1201169029	66	1201169349	59	1201169660	66	1201169993	64	1201170299	63
30	1201169030	67	1201169352	62	1201169662	68	1201169995	66	1201170310	74
31	1201169033	70	1201169353	63	1201169664	70	1201169998	69	1201170312	76
32	1201169034	71	1201169356	66	1201169666	72	1201169999	70	1201170313	77
33	1201169037	74	1201169357	67	1201169669	75	1201170001	72	1201170314	78
34	1201169038	75	1201169359	69	1201169671	77	1201170003	74	1201170315	79
35	1201169041	78	1201169361	71	1201169674	80	1201170006	77	1201170318	82
36	1201169042	79	1201169363	73	1201169676	82	1201170008	79	1201170319	83
37	1201169045	82	1201169366	76	1201169677	83	1201170011	82	1201170323	87
38	1201169046	83	1201169368	78	1201169680	86	1201170011	83	1201170323	88
39	1201169049	86	1201169371	81	1201169681	87	1201170012	86	1201170324	89
40	1201169049	87	1201169371	83	1201169684	90	1201170013	87	1201170323	92
41	1201169053	90	1201169376	86	1201169685	91	1201170010	90	1201170328	93
42	1201169054	91	1201169377	87	1201169688	94	1201170019	91	1201170323	97
43	1201169057	94	1201169377	90	1201169689	95	1201170020	93	1201170333	98
44	1201169058	95	1201169381	91	1201169691	97	1201170024	95	1201170335	99
45	1201169062	99	1201169384	94	1201169693		1201170027	98	1201170338	102
46	1201169062	99	1201169385	95	1201169695	101	1201170029	100	1201170339	103
47	1201169065	102	1201169388	98	1201169697	103	1201170032	103	1201170343	107
48	1201169066	103	1201169389	99	1201169700	106	1201170033	104	1201170345	109
49	1201169068	105	1201169391	101	1201169701	107	1201170036	107	1201170347	111
50	1201169070	107	1201169393	103	1201169704	110	1201170037	108	1201170348	112
51	1201169073	110	1201169395	105	1201169705	111	1201170040	111	1201170351	115
52	1201169076	113	1201169398	108	1201169707	113	1201170041	112	1201170352	116
53	1201169077	114	1201169399	109	1201169709	115	1201170044	115	1201170355	119
54	1201169080	117	1201169401	111	1201169711	117	1201170045	116	1201170356	120
55	1201169081	118	1201169404	114	1201169714	120	1201170048	119	1201170359	123
56	1201169083	120	1201169408	118	1201169716	122	1201170049	120	1201170360	124
57	1201169085	122	1201169409	119	1201169719	125	1201170051	122	1201170363	127
58	1201169087	124	1201169410	120	1201169720	126	1201170054	125	1201170364	128
59	1201169089	126	1201169413	123	1201169722	128	1201170056	127	1201170367	131
60	1201169092	129	1201169414	124	1201169724	130	1201170059	130	1201170368	132
61	1201169093	130	1201169416	126	1201169726	132	1201170060	131	1201170370	134
62	1201169096	133	1201169418	128	1201169728	134	1201170062	133	1201170372	136
63	1201169097	134	1201169421	131	1201169731	137	1201170064	135	1201170374	138
64	1201169099	136	1201169422	132	1201169733	139	1201170067	138	1201170377	141
65	1201169102	139	1201169424	134	1201169736	142	1201170068	139	1201170378	142
66	1201169103	140	1201169426	136	1201169737	143	1201170070	141	1201170381	145
67	1201169105	142	1201169429	139	1201169740	146	1201170072	143	1201170382	146
68	1201169107	144	1201169430	140	1201169741	147	1201170074	145	1201170385	149
69	1201169109	146	1201169432	142	1201169744	150	1201170076	147	1201170386	150
70	1201169112	149	1201169434	144	1201169745	151	1201170078	149	1201170389	153
71	1201169113	150	1201169436	146	1201169747	153	1201170080	151	1201170390	154
72	1201169115	152	1201169438	148	1201169749	155	1201170082	153	1201170392	156
73	1201169117	154	1201169440	150	1201169752	158	1201170085	156	1201170395	159
74	1201169120	157	1201169443	153	1201169753	159	1201170088	159	1201170397	161

76	1201169124	161	1201169448	158	1201169757	163	1201170092	163	1201170403	167
77	1201169125	162	1201169449	159	1201169759	165	1201170093	164	1201170404	168
78	1201169128	165	1201169452	162	1201169762	168	1201170095	166	1201170406	170
79	1201169129	166	1201169453	163	1201169763	169	1201170098	169	1201170408	172
80	1201169131	168	1201169456	166	1201169766	172	1201170100	171	1201170410	174
81	1201169134	171	1201169457	167	1201169767	173	1201170103	174	1201170412	176
82	1201169136	173	1201169459	169	1201169770	176	1201170104	175	1201170414	178
83	1201169138	175	1201169461	171	1201169771	177	1201170107	178	1201170417	181
84	1201169141	178	1201169463	173	1201169774	180	1201170108	179	1201170419	183
85	1201169142	179	1201169466	176	1201169775	181	1201170110	181	1201170422	186
86	1201169144	181	1201169467	177	1201169778	184	1201170112	183	1201170423	187
87	1201169146	183	1201169470	180	1201169779	185	1201170115	186	1201170426	190
88	1201169148	185	1201169471	181	1201169782	188	1201170116	187	1201170427	191
89	1201169151	188	1201169474	184	1201169783	189	1201170118	189	1201170430	194
90	1201169152	189	1201169475	185	1201169786	192	1201170120	191	1201170431	195
91	1201169154	191	1201169478	188	1201169787	193	1201170123	194	1201170434	198
92	1201169156	193	1201169479	189	1201169789	195	1201170124	195	1201170435	199
93	1201169158	195	1201169482	192	1201169791	197	1201170127	198	1201170438	202
94	1201169161	198	1201169483	193	1201169793	199	1201170128	199	1201170439	203
95	1201169164	201	1201169486	196	1201169796	202	1201170130	201	1201170442	206
96	1201169166	203	1201169487	197	1201169799	205	1201170133	204	1201170443	207
97	1201169167	204	1201169490	200	1201169800	206	1201170135	206	1201170445	209
98	1201169169	206	1201169491	201	1201169803	209	1201170138	209	1201170447	211
99	1201169172	209	1201169493	203	1201169804	210	1201170139	210	1201170449	213
100	1201169173	210	1201169495	205	1201169806	212	1201170142	213	1201170452	216

Condor 1	No Work 1	6 Machin	es							
Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A
1	1201170723	0	1201226362	0	1201226702	0	1201227104	0	1201227337	0
2	1201170728	5	1201226366	4	1201226704	2	1201227108	4	1201227339	2
3	1201170734	11	1201226370	8	1201226706	4	1201227112	8	1201227341	4
4	1201170740	17	1201226375	13	1201226708	6	1201227116	12	1201227343	6
5	1201170745	22	1201226380	18	1201226710	8	1201227120	16	1201227345	8
6	1201170748	25	1201226383	21	1201226712	10	1201227125	21	1201227347	10
7	1201170749	26	1201226385	23	1201226714	12	1201227127	23	1201227349	12
8	1201170752	29	1201226387	25	1201226716	14	1201227129	25	1201227351	14
9	1201170754	31	1201226389	27	1201226718	16	1201227130	26	1201227354	17
10	1201170755	32	1201226390	28	1201226720	18	1201227133	29	1201227357	20
11	1201170758	35	1201226393	31	1201226722	20	1201227134	30	1201227359	22
12	1201170759	36	1201226396	34	1201226724	22	1201227136	32	1201227361	24
13	1201170762	39	1201226397	35	1201226726	24	1201227139	35	1201227363	26
14	1201170763	40	1201226399	37	1201226728	26	1201227140	36	1201227364	27
15	1201170765	42	1201226402	40	1201226729	27	1201227143	39	1201227366	29
16	1201170767	44	1201226403	41	1201226732	30	1201227144	40	1201227369	32
17	1201170768	45	1201226405	43	1201226735	33	1201227147	43	1201227370	33
18	1201170770	47	1201226408	46	1201226736	34	1201227148	44	1201227372	35
19	1201170771	48	1201226409	47	1201226738	36	1201227151	47	1201227375	38
20	1201170774	51	1201226411	49	1201226741	39	1201227153	49	1201227378	41
21	1201170775	52	1201226414	52	1201226742	40	1201227155	51	1201227379	42
22	1201170778	55	1201226415	53	1201226745	43	1201227158	54	1201227382	45
23	1201170779	56	1201226418	56	1201226748	46	1201227159	55	1201227383	46
24	1201170782	59	1201226419	57	1201226749	47	1201227162	58	1201227386	49
25	1201170783	60	1201226422	60	1201226752	50	1201227163	59	1201227387	50
26	1201170786	63	1201226423	61	1201226755	53	1201227166	62	1201227390	53
27	1201170787	64	1201226426	64	1201226756	54	1201227167	63	1201227392	55
28	1201170790	67	1201226427	65	1201226759	57	1201227169	65	1201227394	57
29	1201170792	69	1201226430	68	1201226761	59	1201227172	68	1201227397	60
30	1201170795	72	1201226433	71	1201226762	60	1201227173	69	1201227398	61
31	1201170798	75	1201226434	72	1201226764	62	1201227176	72	1201227401	64

1		l =		l	l	l		l ==	l	ا
32	1201170799	76	1201226436	74	1201226767	65	1201227177	73	1201227402	65
33	1201170802	79	1201226439	77	1201226768	66	1201227179	75	1201227405	68
34	1201170803	80	1201226440	78	1201226771	69	1201227182	78	1201227406	69
35	1201170806	83	1201226443	81	1201226772	70	1201227183	79	1201227408	71
36	1201170807	84	1201226444	82	1201226774	72	1201227186	82	1201227412	75
37	1201170810	87	1201226447	85	1201226777	75	1201227187	83	1201227413	76
38	1201170811	88	1201226449	87	1201226778	76	1201227190	86	1201227414	77
39		91	1201226450	91	1201226781	79	1201227191	87	1201227417	80
40	1201170815 1201170817	92	1201226453	91	1201226782	80	1201227194	90	1201227418	81
42	1201170817	97	1201226454	95	1201226785	84	1201227193	94	1201227421	85
43	1201170820	99	1201226457	97	1201226786	87	1201227198	95	1201227422	88
44	1201170822	102	1201226461	99	1201226790	88	1201227199	98	1201227423	89
45	1201170825	103	1201226464	102	1201226793	91	1201227202	99	1201227428	91
46	1201170829	106	1201226466	104	1201226794	92	1201227206	102	1201227420	93
47	1201170830	107	1201226467	105	1201226797	95	1201227207	103	1201227433	96
48	1201170833	110	1201226470	108	1201226798	96	1201227210	106	1201227434	97
49	1201170834	111	1201226471	109	1201226800	98	1201227211	107	1201227437	100
50	1201170836	113	1201226474	112	1201226803	101	1201227214	110	1201227438	101
51	1201170838	115	1201226475	113	1201226804	102	1201227215	111	1201227441	104
52	1201170840	117	1201226478	116	1201226806	104	1201227218	114	1201227442	105
53	1201170842	119	1201226479	117	1201226808	106	1201227219	115	1201227444	107
54	1201170844	121	1201226482	120	1201226810	108	1201227222	118	1201227446	109
55	1201170846	123	1201226483	121	1201226812	110	1201227223	119	1201227448	111
56	1201170848	125	1201226486	124	1201226814	112	1201227226	122	1201227450	113
57	1201170851	128	1201226487	125	1201226817	115	1201227227	123	1201227454	117
58	1201170853	130	1201226490	128	1201226818	116	1201227230	126	1201227455	118
59	1201170856	133	1201226491	129	1201226821	119	1201227231	127	1201227457	120
60	1201170857	134	1201226494	132	1201226822	120	1201227234	130	1201227458	121
61	1201170860	137	1201226497	135	1201226825	123	1201227235	131	1201227461	124
62	1201170861	138	1201226498	136	1201226826	124	1201227238	134	1201227462	125
63	1201170864	141	1201226501	139	1201226829	127	1201227240	136	1201227465	128
64	1201170865	142	1201226502	140	1201226830	128	1201227241	137	1201227466	129
65	1201170868	145	1201226505	143	1201226832	130	1201227244	140	1201227469	132
66	1201170869	146	1201226506	144	1201226834	132	1201227245	141	1201227470	133
67	1201170872	149	1201226509	147	1201226836	134	1201227248	144	1201227473	136
68	1201170873	150	1201226511	149	1201226839	137	1201227250	146	1201227474	137
69	1201170876	153	1201226512	150	1201226840	138	1201227251	147	1201227477	140
70	1201170877	154	1201226515	153	1201226843	141	1201227254	150	1201227478	141
71	1201170879	156	1201226516	154	1201226845	143	1201227255	151	1201227481	144
72	1201170881	158	1201226519	157	1201226846	144	1201227258	154	1201227482	145
73	1201170883	160	1201226520	158	1201226849	147	1201227259	155	1201227484	147
74	1201170886	163	1201226522	160	1201226850	148	1201227262	158	1201227487	150
75	1201170887	164	1201226524	162	1201226853	151	1201227263	159	1201227488	151
76	1201170890	167	1201226527	165	1201226854	152	1201227266	162	1201227490	153
77	1201170891	168	1201226528	166	1201226857	155	1201227267	163	1201227493	156
78	1201170894	171	1201226531	169	1201226858	156	1201227270	166	1201227495	158
79	1201170895	172	1201226532	170	1201226861	159	1201227271	167	1201227496	159
80	1201170898	175	1201226535	173	1201226862	160	1201227274	170	1201227501	164
81	1201170899	176	1201226536	174	1201226864	162	1201227275	171	1201227501	164
82	1201170902	179	1201226539	177	1201226866	164	1201227277	173	1201227502	165
83	1201170903	180	1201226540	178	1201226869	167	1201227280	176	1201227506	169
84	1201170905	182	1201226543	181	1201226870	168	1201227281	177	1201227508	171
85	1201170907	184	1201226544	182	1201226873	171	1201227284	180	1201227511	174
86	1201170909	186	1201226547	185	1201226876	174	1201227285	181	1201227512	175
87	1201170912	189	1201226548	186	1201226877	175	1201227288	184	1201227515	178
88	1201170913	190	1201226551	189	1201226880	178	1201227289	185	1201227516	179
89	1201170916	193	1201226552	190	1201226881	179	1201227292	188	1201227519	182
90	1201170917	194	1201226555	193	1201226884	182	1201227293	189	1201227520	183

91	1201170920	197	1201226556	194	1201226885	183	1201227296	192	1201227523	186
92	1201170921	198	1201226559	197	1201226888	186	1201227297	193	1201227525	188
93	1201170924	201	1201226560	198	1201226889	187	1201227300	196	1201227526	189
94	1201170925	202	1201226563	201	1201226892	190	1201227301	197	1201227529	192
95	1201170928	205	1201226564	202	1201226893	191	1201227304	200	1201227530	193
96	1201170929	206	1201226567	205	1201226895	193	1201227307	203	1201227533	196
97	1201170931	208	1201226568	206	1201226897	195	1201227309	205	1201227534	197
98	1201170934	211	1201226571	209	1201226899	197	1201227310	206	1201227536	199
99	1201170936	213	1201226572	210	1201226901	199	1201227312	208	1201227538	201
100	1201170939	216	1201226575	213	1201226904	202	1201227314	210	1201227541	204

BOINC

BOINC	No Work	l Machine	;							
Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A
1	1203462728	0	1203464632	0	1203472013	0	1203477277	0	1203482877	0
2	1203462743	15	1203464648	16	1203472028	15	1203477305	28	1203482892	15
3	1203462758	30	1203464662	30	1203472043	30	1203477321	44	1203482908	31
4	1203462773	45	1203464678	46	1203472058	45	1203477336	59	1203482923	46
5	1203462788	60	1203464692	60	1203472073	60	1203477350	73	1203482938	61
6	1203462804	76	1203464707	75	1203472088	75	1203477367	90	1203482953	76
7	1203462819	91	1203464722	90	1203472104	91	1203477382	105	1203482969	92
8	1203462835	107	1203464738	106	1203472118	105	1203477397	120	1203482983	106
9	1203462848	120	1203464753	121	1203472134	121	1203477412	135	1203482998	121
10	1203462864	136	1203464769	137	1203472149	136	1203477427	150	1203483013	136
11	1203462880	152	1203464782	150	1203472164	151	1203477442	165	1203483029	152
12	1203462895	167	1203464797	165	1203472179	166	1203477458	181	1203483044	167
13	1203462911	183	1203464813	181	1203472194	181	1203477473	196	1203483059	182
14	1203462925	197	1203464829	197	1203472209	196	1203477488	211	1203483073	196
15	1203462940	212	1203464845	213	1203472224	211	1203477502	225	1203483088	211
16	1203462954	226	1203464859	227	1203472239	226	1203477517	240	1203483104	227
17	1203462970	242	1203464874	242	1203472255	242	1203477532	255	1203483120	243
18	1203462986	258	1203464889	257	1203472270	257	1203477549	272	1203483135	258
19	1203463002	274	1203464905	273	1203472285	272	1203477564	287	1203483150	273
20	1203463016	288	1203464920	288	1203472300	287	1203477644	367	1203483166	289
21	1203463031	303	1203464935	303	1203472315	302	1203477644	367	1203483180	303
22	1203463045	317	1203464950	318	1203472331	318	1203477644	367	1203483195	318
23	1203463061	333	1203464965	333	1203472345	332	1203477674	397	1203483210	333
24	1203463076	348	1203464980	348	1203472361	348	1203477689	412	1203483225	348
25	1203463092	364	1203464995	363	1203472376	363	1203477705	428	1203483240	363
26	1203463107	379	1203465010	378	1203472390	377	1203477720	443	1203483255	378
27	1203463121	393	1203465026	394	1203472406	393	1203477735	458	1203483271	394
28	1203463137	409	1203465041	409	1203472421	408	1203477750	473	1203483286	409
29	1203463153	425	1203465056	424	1203472436	423	1203477765	488	1203483301	424
30	1203463166	438	1203465071	439	1203472452	439	1203477781	504	1203483316	439
31	1203463182	454	1203465087	455	1203472467	454	1203477795	518	1203483330	453
32	1203463197	469	1203465102	470	1203472483	470	1203477811	534	1203483345	468
33	1203463213	485	1203465117	485	1203472498	485	1203477826	549	1203483362	485
34	1203463229	501	1203465133	501	1203472527	514	1203477841	564	1203483377	500
35	1203463243	515	1203465147	515	1203472543	530	1203477857	580	1203483392	515
36	1203463258	530	1203465162	530	1203472558	545	1203477870	593	1203483407	530
37	1203463273	545	1203465177	545	1203472572	559	1203477886	609	1203483423	546
38	1203463288	560	1203465192	560	1203472587	574	1203477901	624	1203483437	560
39	1203463303	575	1203465208	576	1203472603	590	1203477917	640	1203483453	576
40	1203463319	591	1203465223	591	1203472618	605	1203477932	655	1203483468	591
41	1203463334	606	1203465239	607	1203472633	620	1203477947	670	1203483483	606
42	1203463349	621	1203465253	621	1203472649	636	1203477962	685	1203483499	622

1.000100000000000000000000000000000000	42	1202462264		1202465269		1202472664	651	1202477077	700	1202402512	
	43	1203463364	636	1203465268	636	1203472664	651	1203477977	700	1203483513	636
100464425 097											
120461446 712											
S1 100445485 757 101445389 757 101445394 772 101445394 772 101445394 772 101445394 772 101445394 772 101445494 772 101445394 788 1014472874 88 1014473818 88 1024483494 772 S4 103445346 88 102445453 88 1024472874 88 102447315 88 102445450 88 1024472874 88 102447315 82 102447315 82 102447315 82 102447315 82 102447311 82 102447310 82 102447311 82 102447310 82 102447311 82 102447310 82 102447310 82 102447310 82 102447310 82 102447310 82 102447310 82 102447310 82 102447310 82 102447310 82 102447310 82 102447310 82 102447310 82 102447310 82 102447310 82 102447310											
1,003463500 772											
1201463560 802											
1201461546 818											
56 1203463561 83 1203465468 83 1203472985 892 12034781874 897 1203483710 83 57 1203463798 88 1203463898 83 1203472989 92 1203473189 92 12034837325 818 59 1203463608 88 1203467298 82 1203473149 422 1203483735 878 60 1203463631 83 1203465521 893 1203473344 93 1203483735 98 61 120346363 83 1203465556 99 1203473344 131 1203473249 97 1203483785 98 62 1203463666 983 1203465572 90 1203473344 131 1203473280 105 1203483810 92 63 1203467872 99 120346572 90 1203473491 136 1203478381 103 120348381 92 64 1203467733 183 1203465660 99 1203473491 146											
\$\begin{array}{c c c c c c c c c c c c c c c c c c c											
Section 1203463591 863 1203465495 863 1203472955 922 1203473204 927 1203437307 863 939 1203463606 878 1203465515 879 1203472965 937 1203473214 932 1203437355 578 1203463602 938 1203463525 939 1203473496 932 1203473234 937 1203437375 938 1203463606 938 1203465526 944 1203473344 1331 1203473204 937 1203437355 938 1203463606 938 1203465526 944 1203473344 1331 1203473205 1003 1203433816 934 1203463607 969 1203465526 944 1203473344 1331 1203473205 1015 1203433816 936 1203463607 969 1203465526 944 1203473404 1391 1203473205 1015 1203438316 934 1203463737 999 1203465606 944 1203473404 1391 1203473205 1015 1203438316 946 1203463737 999 120346616 949 1203473449 1491 1203473325 1018 120348381 948 1203463737 1015 1203463734 1491 12034637335 1018 120348381 948 1203463733 1015 1203463731 1015 1203473449 1406 1203473355 1018 120348381 949 1203473449 1406 1203463737 1015 1203463734 1406 1203465607 1015 1203473459 1406 1203473831 1015 1203463737 1015 1203473495 1406 1203463737 1015 1203463											
59 1203463666 878 1203463511 879 120347290 937 1203473210 942 1203433515 878 60 1203463621 893 1203463521 893 1203463521 893 1203463521 99 1203473344 131 1203473234 97 1203483705 98 62 120346362 944 1203463525 94 1203473344 1331 1203473234 97 1203483816 92 63 120346362 944 1203463572 90 1203473340 1301 1203473240 103 1203483816 99 64 1203463772 99 1203466616 96 120347340 1301 1301 1203483816 99 66 1203463772 99 120346661 94 120347340 1420 1203473330 1065 1203483861 98 67 1203463733 1015 1203466671 1015 1203473461 1420 1203443881 109 1203483891 194 <td></td>											
100											
120346563 225											
62 1203465666 938 1203465550 924 1203473344 1331 1203478264 987 1203483801 924 63 1203465697 969 1203465588 954 1203473349 1376 1203478280 1003 1203483816 99 65 1203465712 984 1203465601 969 1203473414 1301 1203478205 1018 120348386 99 66 120346712 984 1203465016 984 1203473414 1406 1203478100 1033 1203483861 994 67 1203467373 1015 120346661 99 1203473481 1421 1203478340 1063 1203483861 994 60 1203467378 1016 1203473494 1466 1203478355 1078 1204488892 1015 70 1203467878 1001 1203465677 1045 1203473495 1482 1203478361 1109 1203483922 1045 71 1203468803 1075 1											
63											
65 1203463712											
1203463727 999											
1203463743 1015	65	1203463712	984	1203465601	969	1203473419	1406	1203478310	1033	1203483846	969
1203463758 1030	66	1203463727	999	1203465616	984	1203473434	1421	1203478325	1048	1203483861	984
69 1203463773 1045 1203465662 1030 1203473479 1466 1203478371 1094 1203483907 1030 70 1203463789 1061 1203465677 1045 1203473495 1482 1203478386 1109 1203483922 1045 71 1203463818 1090 1203465707 1075 1203473525 1512 1203478416 1124 1203483937 1060 72 1203463818 1106 1203465707 1075 1203473521 1497 1203478416 1139 1203483952 1075 73 1203463849 1106 1203465739 1017 1203473541 1528 1203478431 1154 1203483967 1090 74 1203463849 1161 1203465783 1107 1203473555 1542 1203478431 1144 1203483982 1105 75 1203463849 1166 1203465788 1136 1203473560 1557 1203478466 1199 1203484013 1136	67	1203463743	1015	1203465631	999	1203473449	1436	1203478340	1063	1203483876	999
70 1203463789 1061 1203465677 1045 1203473495 1482 1203478386 1109 1203483922 1045 71 1203463803 1075 1203465693 1061 1203473510 1497 1203478401 1124 1203483937 1060 72 1203463818 1090 1203465723 1091 1203473525 1512 1203478416 1139 1203483952 1075 74 1203463849 1121 1203465739 1107 1203473555 1512 1203478446 1169 1203483987 1105 75 1203463849 1121 1203465753 1121 1203473555 1557 1203478461 1184 1203483988 1126 76 1203463879 1151 1203465788 1136 1203473570 1557 1203478461 1184 12034840898 1121 77 1203463894 1166 1203465784 1152 1203473500 1587 1203478462 1215 12034840428 1151 <t< td=""><td>68</td><td>1203463758</td><td>1030</td><td>1203465647</td><td>1015</td><td>1203473465</td><td>1452</td><td>1203478355</td><td>1078</td><td>1203483892</td><td>1015</td></t<>	68	1203463758	1030	1203465647	1015	1203473465	1452	1203478355	1078	1203483892	1015
71 1203463803 1075 1203465693 1061 1203473510 1497 1203478401 1124 1203483937 1060 72 1203463818 1090 1203465707 1075 1203473525 1512 1203478416 1139 1203483952 1075 73 1203463841 1106 1203465723 1091 1203473541 1528 1203478461 1139 1203483967 1090 74 1203463840 1121 1203465739 1107 1203473555 1542 1203478461 1169 1203483982 1105 75 1203463840 1151 1203465753 1121 12034765768 1136 1203473588 1557 1203478461 1199 1203484013 1116 76 1203463894 1166 1203465784 1152 1203473588 1575 1203478462 1215 1203484028 1151 78 1203463990 1181 1203465813 1181 1203473682 1619 1203478522 1245 1203484088	69	1203463773	1045	1203465662	1030	1203473479	1466	1203478371	1094	1203483907	1030
72 1203463818 1090 1203465707 1075 1203473525 1512 1203478416 1139 1203483952 1075 73 1203463834 1106 1203465723 1091 1203473541 1528 1203478446 1169 1203483967 1090 74 1203463849 1121 1203465739 1107 1203473555 1542 1203478446 1169 1203483982 1105 75 1203463879 1151 1203465768 1136 1203473507 1557 1203478461 1184 1203484033 1136 77 1203463899 1181 12034565768 1136 1203473600 1887 1203478402 1215 1203484033 1166 79 1203463999 1181 1203465798 1166 1203473616 1603 1203478507 1230 1203484038 1181 80 1203463999 1271 1203465813 1181 1203473648 1659 1203478507 1230 1203484058 1181 <td< td=""><td>70</td><td>1203463789</td><td>1061</td><td>1203465677</td><td>1045</td><td>1203473495</td><td>1482</td><td>1203478386</td><td>1109</td><td>1203483922</td><td>1045</td></td<>	70	1203463789	1061	1203465677	1045	1203473495	1482	1203478386	1109	1203483922	1045
73 1203463834 1106 1203465723 1091 1203473541 1528 1203478431 1154 1203483967 1090 74 1203463849 1121 1203465739 1107 1203473555 1542 1203478446 1169 120348382 1105 75 1203463864 1136 1203465753 1121 1203473570 1557 1203478461 1184 1203483998 1121 76 1203463879 1151 1203465784 1152 1203473600 1587 1203478476 1199 1203484013 1136 77 1203463999 1181 1203465784 1152 1203473600 1587 1203478492 1215 1203484028 1151 78 1203463999 1181 1203465784 1152 1203473602 1619 1203478522 1245 1203484038 1166 79 1203463999 1271 1203465813 1181 1203473632 1619 1203478522 1245 1203484058 1181 8	71	1203463803	1075	1203465693	1061	1203473510	1497	1203478401	1124	1203483937	1060
74 1203463849 1121 1203465739 1107 1203473555 1542 1203478446 1169 1203483982 1105 75 1203463864 1136 1203465753 1121 1203473570 1557 1203478461 1184 1203483998 1121 76 1203463879 1151 1203465768 1136 1203473580 1557 1203478476 1199 1203484013 1136 77 1203463894 1166 1203465784 1152 1203473600 1587 1203478492 1215 1203484028 1151 78 1203463999 1181 1203465798 1166 1203473616 1603 1203478507 1230 120348403 1166 79 1203463926 1198 1203465813 1181 1203473662 1619 1203478537 1260 1203484058 1181 80 1203463999 1271 1203465841 1212 1203473663 1650 1203478557 1290 1203484088 1211 8	72	1203463818	1090	1203465707	1075	1203473525	1512	1203478416	1139	1203483952	1075
75 1203463864 1136 1203465753 1121 1203473570 1557 1203478461 1184 1203433998 1121 76 1203463879 1151 1203465768 1136 1203473588 1575 1203478476 1199 1203484013 1136 77 1203463894 1166 1203465784 1152 1203473600 1587 1203478492 1215 1203484028 1151 78 1203463909 1181 1203465798 1166 1203473616 1603 1203478507 1230 1203484033 1166 79 1203463926 1198 1203465813 1181 1203473652 1619 1203478522 1245 1203484088 1181 80 1203463999 1271 1203465828 1196 1203473663 1650 1203478557 1275 1203484088 1211 81 1203463999 1271 1203465859 1227 1203473677 1664 1203478582 1305 1203484103 1226	73	1203463834	1106	1203465723	1091	1203473541	1528	1203478431	1154	1203483967	1090
76 1203463879 1151 1203465768 1136 1203473588 1575 1203478476 1199 1203484013 1136 77 1203463894 1166 1203465784 1152 1203473600 1587 1203478492 1215 1203484028 1151 78 1203463990 1181 1203465798 1166 1203473616 1603 1203478507 1230 1203484043 1166 79 1203463926 1198 1203465813 1181 1203473632 1619 1203478522 1245 1203484058 1181 80 1203463999 1271 1203465828 1196 1203473663 1650 1203478537 1260 1203484074 1197 81 1203463999 1271 1203465874 1212 1203473667 1664 1203478567 1290 1203484108 1211 82 1203460299 1301 1203465874 1242 1203473691 1678 1203478567 1290 1203484118 1241	74	1203463849	1121	1203465739	1107	1203473555	1542	1203478446	1169	1203483982	1105
77 1203463894 1166 1203465784 1152 1203473600 1587 1203478492 1215 1203484028 1151 78 1203463909 1181 1203465798 1166 1203473616 1603 1203478507 1230 1203484043 1166 79 1203463926 1198 1203465813 1181 1203473632 1619 1203478522 1245 1203484058 1181 80 1203463999 1271 1203465828 1196 1203473663 1650 1203478552 125 1203484074 1197 81 1203463999 1271 1203465844 1212 1203473667 1664 1203478552 1275 1203484088 1211 82 1203463099 1271 1203465874 1242 1203473607 1664 1203478552 1290 1203484088 1211 84 1203464029 1301 1203465889 1257 1203473707 1694 1203478582 1305 1203484118 1241 8	75	1203463864	1136	1203465753	1121	1203473570	1557	1203478461	1184	1203483998	1121
78 1203463999 1181 1203465798 1166 1203473616 1603 1203478507 1230 1203484043 1166 79 1203463926 1198 1203465813 1181 1203473632 1619 1203478522 1245 1203484058 1181 80 1203463999 1271 1203465828 1196 1203473648 1635 1203478537 1260 1203484074 1197 81 1203463999 1271 1203465844 1212 1203473663 1650 1203478552 1275 1203484088 1211 82 1203463999 1271 1203465874 1242 1203473677 1664 1203478567 1290 1203484103 1226 83 1203464029 1301 1203465874 1242 1203473691 1678 1203478582 1305 1203484118 1241 84 1203464040 1316 1203465894 1272 1203473707 1694 1203478597 1320 1203484134 1257	76	1203463879	1151	1203465768	1136	1203473588	1575	1203478476	1199	1203484013	1136
79 1203463926 1198 1203465813 1181 1203473632 1619 1203478522 1245 1203484058 1181 80 1203463999 1271 1203465828 1196 1203473648 1635 1203478537 1260 1203484074 1197 81 1203463999 1271 1203465844 1212 1203473677 1664 1203478552 1275 1203484088 1211 82 1203464029 1301 1203465874 1242 1203473691 1678 1203478582 1305 1203484103 1226 83 1203464044 1316 1203465889 1257 1203473707 1694 1203478582 1305 1203484118 1241 84 1203464060 1332 1203465904 1272 1203473737 1694 1203478613 1336 1203484149 1272 86 1203464075 1347 1203465919 1287 1203473737 1724 1203478628 1351 1203484164 1287	77	1203463894	1166	1203465784	1152	1203473600	1587	1203478492	1215	1203484028	1151
80 1203463999 1271 1203465828 1196 1203473648 1635 1203478537 1260 1203484074 1197 81 1203463999 1271 1203465844 1212 1203473663 1650 1203478552 1275 1203484088 1211 82 1203463999 1271 1203465859 1227 1203473677 1664 1203478567 1290 1203484103 1226 83 1203464029 1301 1203465874 1242 1203473691 1678 1203478592 1305 1203484118 1241 84 1203464044 1316 1203465894 1257 1203473707 1694 1203478597 1320 1203484134 1257 85 1203464060 1332 1203465919 1287 1203473737 1710 1203478628 1351 1203484144 1272 86 1203464075 1347 1203465935 1303 1203473737 1724 1203478634 1366 1203484149 1272	78	1203463909	1181	1203465798	1166	1203473616	1603	1203478507	1230	1203484043	1166
81 1203463999 1271 1203465844 1212 1203473663 1650 1203478552 1275 1203484088 1211 82 1203463999 1271 1203465859 1227 1203473677 1664 1203478567 1290 1203484103 1226 83 1203464029 1301 1203465874 1242 1203473691 1678 1203478582 1305 1203484118 1241 84 1203464029 1301 1203465889 1257 1203473707 1694 1203478597 1320 1203484134 1257 85 1203464060 1332 1203465904 1272 1203473733 1710 1203478613 1336 1203484149 1272 86 1203464075 1347 1203465919 1287 1203473737 1724 1203478628 1351 1203484144 1287 87 1203464090 1362 1203465935 1303 1203473737 1724 1203478643 1366 1203484149 1372	79	1203463926	1198	1203465813	1181	1203473632	1619	1203478522	1245	1203484058	1181
82 1203463999 1271 1203465859 1227 1203473677 1664 1203478567 1290 1203484103 1226 83 1203464029 1301 1203465874 1242 1203473691 1678 1203478582 1305 1203484118 1241 84 1203464044 1316 1203465889 1257 1203473707 1694 1203478597 1320 1203484134 1257 85 1203464060 1332 1203465904 1272 1203473723 1710 1203478613 1336 1203484149 1272 86 1203464075 1347 1203465919 1287 1203473737 1724 1203478628 1351 1203484164 1287 87 1203464090 1362 1203465950 1318 1203473737 1724 120347863 1366 1203484194 1317 89 1203464105 1377 1203465965 1333 1203473768 1755 1203478673 1396 1203484208 1331 9	80	1203463999	1271	1203465828	1196	1203473648	1635	1203478537	1260	1203484074	1197
83 1203464029 1301 1203465874 1242 1203473691 1678 1203478582 1305 1203484118 1241 84 1203464044 1316 1203465889 1257 1203473707 1694 1203478597 1320 1203484134 1257 85 1203464060 1332 1203465904 1272 1203473723 1710 1203478613 1336 1203484149 1272 86 1203464075 1347 1203465919 1287 1203473737 1724 1203478628 1351 1203484164 1287 87 1203464090 1362 1203465935 1303 1203473737 1724 1203478643 1366 1203484179 1302 88 1203464105 1377 1203465950 1318 1203473753 1740 1203478678 1381 1203484194 1317 89 1203464120 1392 1203465950 1333 1203473768 1755 1203478673 1396 1203484240 1331	81	1203463999	1271	1203465844	1212	1203473663	1650	1203478552	1275	1203484088	1211
84 1203464044 1316 1203465889 1257 1203473707 1694 1203478597 1320 1203484134 1257 85 1203464060 1332 1203465904 1272 1203473723 1710 1203478613 1336 1203484149 1272 86 1203464075 1347 1203465919 1287 1203473737 1724 1203478628 1351 1203484164 1287 87 1203464090 1362 1203465935 1303 1203473737 1724 1203478643 1366 1203484179 1302 88 1203464105 1377 1203465950 1318 1203473753 1740 1203478658 1381 1203484194 1317 89 1203464120 1392 1203465965 1333 1203473768 1755 1203478673 1396 1203484208 1331 90 1203464136 1408 1203465995 1363 1203473783 1770 1203478688 1411 1203484240 1363	82	1203463999	1271	1203465859	1227	1203473677	1664	1203478567	1290	1203484103	1226
85 1203464060 1332 1203465904 1272 1203473723 1710 1203478613 1336 1203484149 1272 86 1203464075 1347 1203465919 1287 1203473737 1724 1203478628 1351 1203484164 1287 87 1203464090 1362 1203465935 1303 1203473737 1724 1203478643 1366 1203484179 1302 88 1203464105 1377 1203465950 1318 1203473753 1740 1203478658 1381 1203484194 1317 89 1203464120 1392 1203465965 1333 1203473788 1755 1203478673 1396 1203484208 1331 90 1203464136 1408 1203465980 1348 1203473783 1770 1203478688 1411 1203484224 1347 91 1203464150 1422 1203465995 1363 1203473799 1786 1203478718 1441 1203484240 1363	83	1203464029	1301	1203465874	1242	1203473691	1678	1203478582	1305	1203484118	1241
86 1203464075 1347 1203465919 1287 1203473737 1724 1203478628 1351 1203484164 1287 87 1203464090 1362 1203465935 1303 1203473737 1724 1203478643 1366 1203484179 1302 88 1203464105 1377 1203465950 1318 1203473753 1740 1203478658 1381 1203484194 1317 89 1203464120 1392 1203465965 1333 1203473768 1755 1203478673 1396 1203484208 1331 90 1203464136 1408 1203465980 1348 1203473783 1770 1203478688 1411 1203484224 1347 91 1203464150 1422 1203465995 1363 1203473799 1786 1203478703 1426 1203484240 1363 92 1203464166 1438 1203466011 1379 1203473814 1801 1203478718 1441 1203484255 1378	84	1203464044	1316	1203465889	1257	1203473707	1694	1203478597	1320	1203484134	1257
87 1203464090 1362 1203465935 1303 1203473737 1724 1203478643 1366 1203484179 1302 88 1203464105 1377 1203465950 1318 1203473753 1740 1203478658 1381 1203484194 1317 89 1203464120 1392 1203465965 1333 1203473768 1755 1203478673 1396 1203484208 1331 90 1203464136 1408 1203465980 1348 1203473783 1770 1203478688 1411 1203484224 1347 91 1203464150 1422 1203465995 1363 1203473799 1786 1203478703 1426 1203484240 1363 92 1203464166 1438 1203466011 1379 1203473814 1801 1203478718 1441 1203484255 1378 93 1203464181 1453 1203466025 1393 1203473844 1831 1203478734 1457 1203484271 1394	85	1203464060	1332	1203465904	1272	1203473723	1710	1203478613	1336	1203484149	1272
88 1203464105 1377 1203465950 1318 1203473753 1740 1203478658 1381 1203484194 1317 89 1203464120 1392 1203465965 1333 1203473768 1755 1203478673 1396 1203484208 1331 90 1203464136 1408 1203465980 1348 1203473783 1770 1203478688 1411 1203484224 1347 91 1203464150 1422 1203465995 1363 1203473799 1786 1203478703 1426 1203484240 1363 92 1203464166 1438 1203466011 1379 1203473814 1801 1203478718 1441 1203484255 1378 93 1203464181 1453 1203466025 1393 1203473830 1817 1203478734 1457 1203484271 1394 94 1203464196 1468 1203466041 1409 1203473844 1831 1203478749 1472 1203484285 1408	86	1203464075	1347	1203465919	1287	1203473737	1724	1203478628	1351	1203484164	1287
89 1203464120 1392 1203465965 1333 1203473768 1755 1203478673 1396 1203484208 1331 90 1203464136 1408 1203465980 1348 1203473783 1770 1203478688 1411 1203484224 1347 91 1203464150 1422 1203465995 1363 1203473799 1786 1203478703 1426 1203484240 1363 92 1203464166 1438 1203466011 1379 1203473814 1801 1203478718 1441 1203484255 1378 93 1203464181 1453 1203466025 1393 1203473830 1817 1203478734 1457 1203484271 1394 94 1203464196 1468 1203466041 1409 1203473844 1831 1203478749 1472 1203484285 1408 95 1203464221 1483 1203466055 1423 1203473879 1846 1203478765 1488 1203484300 1423	87	1203464090	1362	1203465935	1303	1203473737	1724	1203478643	1366	1203484179	1302
90 1203464136 1408 1203465980 1348 1203473783 1770 1203478688 1411 1203484224 1347 91 1203464150 1422 1203465995 1363 1203473799 1786 1203478703 1426 1203484240 1363 92 1203464166 1438 1203466011 1379 1203473814 1801 1203478718 1441 1203484255 1378 93 1203464181 1453 1203466025 1393 1203473830 1817 1203478734 1457 1203484271 1394 94 1203464196 1468 1203466041 1409 1203473844 1831 1203478749 1472 1203484285 1408 95 1203464211 1483 1203466055 1423 1203473859 1846 1203478765 1488 1203484300 1423 96 1203464227 1499 1203466086 1454 1203473890 1877 1203478794 1517 1203484315 1438	88	1203464105	1377	1203465950	1318	1203473753	1740	1203478658	1381	1203484194	1317
91 1203464150 1422 1203465995 1363 1203473799 1786 1203478703 1426 1203484240 1363 92 1203464166 1438 1203466011 1379 1203473814 1801 1203478718 1441 1203484255 1378 93 1203464181 1453 1203466025 1393 1203473830 1817 1203478734 1457 1203484271 1394 94 1203464196 1468 1203466041 1409 1203473844 1831 1203478749 1472 1203484285 1408 95 1203464211 1483 1203466055 1423 1203473859 1846 1203478765 1488 1203484300 1423 96 1203464227 1499 1203466071 1439 1203473874 1861 1203478779 1502 1203484315 1438 97 1203464242 1514 1203466086 1454 1203473890 1877 1203478809 1532 1203484331 1454	89	1203464120	1392	1203465965	1333	1203473768	1755	1203478673	1396	1203484208	1331
92 1203464166 1438 1203466011 1379 1203473814 1801 1203478718 1441 1203484255 1378 93 1203464181 1453 1203466025 1393 1203473830 1817 1203478734 1457 1203484271 1394 94 1203464196 1468 1203466041 1409 1203473844 1831 1203478749 1472 1203484285 1408 95 1203464211 1483 1203466055 1423 1203473859 1846 1203478765 1488 1203484300 1423 96 1203464227 1499 1203466071 1439 1203473874 1861 1203478779 1502 1203484315 1438 97 1203464242 1514 1203466086 1454 1203473890 1877 1203478794 1517 1203484331 1454 98 120346256 1528 1203466102 1470 1203473905 1892 1203478809 1532 1203484346 1469	90	1203464136	1408	1203465980	1348	1203473783	1770	1203478688	1411	1203484224	1347
93 1203464181 1453 1203466025 1393 1203473830 1817 1203478734 1457 1203484271 1394 94 1203464196 1468 1203466041 1409 1203473844 1831 1203478749 1472 1203484285 1408 95 1203464211 1483 1203466055 1423 1203473859 1846 1203478765 1488 1203484300 1423 96 1203464227 1499 1203466071 1439 1203473874 1861 1203478779 1502 1203484315 1438 97 1203464242 1514 1203466086 1454 1203473890 1877 1203478794 1517 1203484331 1454 98 1203464256 1528 1203466102 1470 1203473905 1892 1203478809 1532 1203484346 1469	91	1203464150	1422	1203465995	1363	1203473799	1786	1203478703	1426	1203484240	1363
94 1203464196 1468 1203466041 1409 1203473844 1831 1203478749 1472 1203484285 1408 95 1203464211 1483 1203466055 1423 1203473859 1846 1203478765 1488 1203484300 1423 96 1203464227 1499 1203466071 1439 1203473874 1861 1203478779 1502 1203484315 1438 97 1203464242 1514 1203466086 1454 1203473890 1877 1203478794 1517 1203484331 1454 98 1203464256 1528 1203466102 1470 1203473905 1892 1203478809 1532 1203484346 1469	92	1203464166	1438	1203466011	1379	1203473814	1801	1203478718	1441	1203484255	1378
95 1203464211 1483 1203466055 1423 1203473859 1846 1203478765 1488 1203484300 1423 96 1203464227 1499 1203466071 1439 1203473874 1861 1203478779 1502 1203484315 1438 97 1203464242 1514 1203466086 1454 1203473890 1877 1203478794 1517 1203484331 1454 98 1203464256 1528 1203466102 1470 1203473905 1892 1203478809 1532 1203484346 1469	93	1203464181	1453	1203466025	1393	1203473830	1817	1203478734	1457	1203484271	1394
96 1203464227 1499 1203466071 1439 1203473874 1861 1203478779 1502 1203484315 1438 97 1203464242 1514 1203466086 1454 1203473890 1877 1203478794 1517 1203484331 1454 98 1203464256 1528 1203466102 1470 1203473905 1892 1203478809 1532 1203484346 1469	94	1203464196	1468	1203466041	1409	1203473844	1831	1203478749	1472	1203484285	1408
97 1203464242 1514 1203466086 1454 1203473890 1877 1203478794 1517 1203484331 1454 98 1203464256 1528 1203466102 1470 1203473905 1892 1203478809 1532 1203484346 1469	95	1203464211	1483	1203466055	1423	1203473859	1846	1203478765	1488	1203484300	1423
98 1203464256 1528 1203466102 1470 1203473905 1892 1203478809 1532 1203484346 1469	96	1203464227	1499	1203466071	1439	1203473874	1861	1203478779	1502	1203484315	1438
	97	1203464242	1514	1203466086	1454	1203473890	1877	1203478794	1517	1203484331	1454
99 1203464271 1543 1203466116 1484 1203473920 1907 1203478824 1547 1203484361 1484	98	1203464256	1528	1203466102	1470	1203473905	1892	1203478809	1532	1203484346	1469
	99	1203464271	1543	1203466116	1484	1203473920	1907	1203478824	1547	1203484361	1484
100 1203464337 1609 1203466181 1549 1203473985 1972 1203478889 1612 1203484426 1549	100	1203464337	1609	1203466181	1549	1203473985	1972	1203478889	1612	1203484426	1549

BOING	No Work	2 Machine	es .							
Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A
1	1203416092	0	1203417220	0	1203418327	0	1203460687	0	1203461650	0
2	1203416096	4	1203417235	15	1203418334	7	1203460694	7	1203461653	3
3	1203416108	16	1203417249	29	1203418343	16	1203460703	16	1203461665	15
4	1203416111	19	1203417265	45	1203418349	22	1203460709	22	1203461668	18
5	1203416123	31	1203417282	62	1203418358	31	1203460718	31	1203461681	31
6	1203416126	34	1203417297	77	1203418365	38	1203460723	36	1203461683	33
7	1203416138	46	1203417311	91	1203418372	45	1203460734	47	1203461695	45
8	1203416141	49	1203417325	105	1203418379	52	1203460739	52	1203461698	48
9	1203416153	61	1203417341	121	1203418386	59	1203460749	62	1203461711	61
10	1203416156	64	1203417357	137	1203418394	67	1203460754	67	1203461713	63
11	1203416168	76	1203417371	151	1203418402	75	1203460764	77	1203461725	75
12	1203416171	79	1203417371	167	1203418409	82	1203460770	83	1203461728	78
						90				92
13	1203416183	91	1203417402	182	1203418417		1203460779	92	1203461742	
14	1203416186	94	1203417417	197	1203418425	98	1203460785	98	1203461744	94
15	1203416199	107	1203417432	212	1203418433	106	1203460794	107	1203461757	107
16	1203416201	109	1203417447	227	1203418440	113	1203460799	112	1203461759	109
17	1203416215	123	1203417462	242	1203418448	121	1203460809	122	1203461772	122
18	1203416218	126	1203417477	257	1203418455	128	1203460815	128	1203461774	124
19	1203416229	137	1203417493	273	1203418463	136	1203460824	137	1203461787	137
20	1203416232	140	1203417510	290	1203418470	143	1203460830	143	1203461789	139
21	1203416244	152	1203417538	318	1203418478	151	1203460840	153	1203461801	151
22	1203416247	155	1203417552	332	1203418485	158	1203460845	158	1203461804	154
23	1203416259	167	1203417567	347	1203418494	167	1203460856	169	1203461816	166
24	1203416263	171	1203417579	359	1203418500	173	1203460860	173	1203461819	169
25	1203416273	181	1203417584	364	1203418509	182	1203460869	182	1203461833	183
26	1203416278	186	1203417596	376	1203418516	189	1203460876	189	1203461835	185
27	1203416291	199	1203417599	379	1203418524	197	1203460885	198	1203461848	198
28	1203416292	200	1203417611	391	1203418530	203	1203460891	204	1203461850	200
29	1203416304	212	1203417613	393	1203418540	213	1203460900	213	1203461863	213
30	1203416306	214	1203417626	406	1203418545	218	1203460906	219	1203461865	215
31	1203416321	229	1203417628	408	1203418554	227	1203460916	229	1203461878	228
32	1203416323	231	1203417640	420	1203418561	234	1203460920	233	1203461880	230
33	1203416334	242	1203417643	423	1203418569	242	1203460931	244	1203461892	242
34		246	1203417656	436	1203418576	249	1203460936	249	1203461895	245
	1203416338					-				
35	1203416351	259	1203417660	440	1203418585	258	1203460953	266	1203461909	259
36	1203416353	261	1203417671	451	1203418591	264	1203460960	273	1203461910	260
37	1203416365	273	1203417675	455	1203418600	273	1203460966	279	1203461923	273
38	1203416368	276	1203417687	467	1203418606	279	1203460976	289	1203461925	275
39	1203416382	290	1203417689	469	1203418615	288	1203460983	296	1203461938	288
40	1203416384	292	1203417702	482	1203418621	294	1203460990	303	1203461941	291
41	1203416396	304	1203417705	485	1203418630	303	1203460997	310	1203461954	304
42	1203416398	306	1203417717	497	1203418636	309	1203461005	318	1203461956	306
43	1203416411	319	1203417720	500	1203418645	318	1203461012	325	1203461970	320
44	1203416413	321	1203417731	511	1203418651	324	1203461021	334	1203461971	321
45	1203416427	335	1203417735	515	1203418661	334	1203461026	339	1203461984	334
46	1203416428	336	1203417747	527	1203418667	340	1203461037	350	1203461986	336
47	1203416442	350	1203417749	529	1203418676	349	1203461042	355	1203462000	350
48	1203416445	353	1203417762	542	1203418683	356	1203461052	365	1203462001	351
49	1203416457	365	1203417766	546	1203418691	364	1203461058	371	1203462015	365
50	1203416459	367	1203417777	557	1203418696	369	1203461067	380	1203462016	366
51	1203416472	380	1203417777	561	1203418706	379	1203461073	386	1203462031	381
52	1203416474	382	1203417781	572	1203418700	385	1203461073	395	1203462031	381
		395			1203418712	393		401		394
53	1203416487		1203417796	576			1203461088		1203462044	
54	1203416489	397	1203417808	588	1203418727	400	1203461096	409	1203462047	397
55	1203416501	409	1203417823	603	1203418738	411	1203461103	416	1203462061	411
56	1203416504	412	1203417838	618	1203418742	415	1203461112	425	1203462062	412
57	1203416518	426	1203417852	632	1203418751	424	1203461119	432	1203462076	426

58	1203416519	427	1203417870	650	1203418757	430	1203461128	441	1203462077	427
59	1203416532	440	1203417876	656	1203418768	441	1203461134	447	1203462092	442
60	1203416534	442	1203417876	656	1203418772	445	1203461143	456	1203462092	
61	1203416549	457	1203417876	656	1203418781	454	1203461148	461	1203462106	456
62		457	1203417885	665	1203418787	460	1203461158	471	1203462107	457
63	1203416564	472	1203417899	679	1203418797	470	1203461163	476	1203462121	471
64	1203416565	473	1203417906	686	1203418804	477	1203461173	486	1203462122	472
65	1203416579	487	1203417915	695	1203418814	487	1203461179	492	1203462136	486
66	1203416580	488	1203417921	701	1203418818	491	1203461188	501	1203462138	488
67	1203416594	502	1203417929	709	1203418828	501	1203461194	507	1203462152	502
68	1203416595	503	1203417937	717	1203418834	507	1203461203	516	1203462153	503
69	1203416608	516	1203417944	724	1203418842	515	1203461211	524	1203462167	517
70	1203416610	518	1203417952	732	1203418848	521	1203461218	531	1203462168	518
71	1203416624	532	1203417960	740	1203418858	531	1203461226	539	1203462182	532
72	1203416625	533	1203417967	747	1203418863	536	1203461239	552	1203462183	533
73	1203416639	547	1203417975	755	1203418874	547	1203461255	568	1203462197	547
74	1203416640	548	1203417982	762	1203418879	552	1203461270	583	1203462198	548
75	1203416654	562	1203417990	770	1203418888	561	1203461286	599	1203462213	563
76	1203416655	563	1203417998	778	1203418893	566	1203461299	612	1203462213	563
77	1203416669	577	1203418006	786	1203418904	577	1203461299	612	1203462228	578
78	1203416671	579	1203418013	793	1203418909	582	1203461299	612	1203462229	579
79	1203416685	593	1203418021	801	1203418919	592	1203461301	614	1203462243	593
80	1203416686	594	1203418028	808	1203418924	597	1203461316	629	1203462244	594
81	1203416701	609	1203418034	814	1203418934	607	1203461329	642	1203462259	609
82	1203416714	622	1203418043	823	1203418939	612	1203461330	643	1203462260	610
83	1203416717	625	1203418052	832	1203418949	622	1203461344	657	1203462273	623
84	1203416730	638	1203418058	838	1203418954	627	1203461347	660	1203462273	623
85	1203416731	639	1203418066	846	1203418965	638	1203461359	672	1203462289	639
86	1203416744	652	1203418073	853	1203418969	642	1203461360	673	1203462289	639
87	1203416746	654	1203418081	861	1203418979	652	1203461374	687	1203462304	654
88	1203416761	669	1203418088	868	1203418984	657	1203461378	691	1203462304	654
89	1203416762	670	1203418096	876	1203418996	669	1203461390	703	1203462318	668
90	1203416777	685	1203418103	883	1203418996	669	1203461391	704	1203462318	668
91	1203416777	685	1203418112	892	1203419000	673	1203461405	718	1203462335	685
92	1203416792	700	1203418119	899	1203419015	688	1203461407	720	1203462335	685
93	1203416792	700	1203418127	907	1203419025	698	1203461421	734	1203462350	700
94	1203416806	714	1203418134	914	1203419030	703	1203461421	734	1203462350	700
95	1203416807	715	1203418143	923	1203419041	714	1203461435	748	1203462363	713
96	1203416822	730	1203418148	928	1203419045	718	1203461438	751	1203462365	715
97	1203416822	730	1203418156	936	1203419056	729	1203461451	764	1203462380	730
98	1203416886	794	1203418163	943	1203419060	733	1203461451	764	1203462381	731
99	1203416887	795	1203418222	1002	1203419121	794	1203461515	828	1203462445	795
100	1203417085	993	1203418229	1009	1203419125	798	1203461518	831	1203462445	795
L	1	1		l .	1	1	1	l .	1	

BOINC	No Work 4	4 Machine	s							
Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A
1	1203411825	0	1203412573	0	1203413181	0	1203414291	0	1203415135	0
2	1203411826	1	1203412578	5	1203413181	0	1203414296	5	1203415142	7
3	1203411826	1	1203412579	6	1203413181	0	1203414296	5	1203415150	15
4	1203411826	1	1203412588	15	1203413187	6	1203414296	5	1203415158	23
5	1203411841	16	1203412592	19	1203413195	14	1203414307	16	1203415165	30
6	1203411842	17	1203412595	22	1203413195	14	1203414310	19	1203415173	38
7	1203411842	17	1203412604	31	1203413196	15	1203414310	19	1203415182	47
8	1203411844	19	1203412607	34	1203413210	29	1203414311	20	1203415187	52
9	1203411855	30	1203412610	37	1203413210	29	1203414322	31	1203415196	61
10	1203411856	31	1203412619	46	1203413212	31	1203414324	33	1203415203	68
11	1203411857	32	1203412623	50	1203413226	45	1203414325	34	1203415211	76
12	1203411870	45	1203412623	50	1203413226	45	1203414326	35	1203415216	81
13	1203411871	46	1203412634	61	1203413226	45	1203414338	47	1203415226	91

1	1	ĺ	I	I	I	I	I	I	1	ı
14	1203411872	47	1203412638	65	1203413239	58	1203414340	49	1203415233	98
15	1203411872	47	1203412638	65	1203413240	59	1203414341	50	1203415241	106
16	1203411885	60	1203412649	76	1203413241	60	1203414342	51	1203415248	113
17	1203411886	61	1203412653	80	1203413256	75	1203414351	60	1203415257	122
18	1203411887	62	1203412656	83	1203413256	75	1203414356	65	1203415263	128
19	1203411887	62	1203412665	92	1203413256	75	1203414356	65	1203415271	136
20	1203411902	77	1203412668	95	1203413270	89	1203414356	65	1203415278	143
21	1203411902	77	1203412670	97	1203413271	90	1203414369	78	1203415292	157
22	1203411903	78	1203412681	108	1203413273	92	1203414369	78	1203415293	158
23	1203411916	91	1203412683	110	1203413286	105	1203414370	79	1203415307	172
24	1203411916	91	1203412685	112	1203413286	105	1203414371	80	1203415308	173
25	1203411917	92	1203412696	123	1203413288	107	1203414383	92	1203415323	188
26	1203411917	92	1203412698	125	1203413301	120	1203414386	95	1203415324	189
27	1203411932	107	1203412700	127	1203413302	121	1203414386	95	1203415338	203
28	1203411932	107	1203412710	137	1203413316	135	1203414386	95	1203415340	205
29	1203411932	107	1203412713	140	1203413317	136	1203414398	107	1203415352	217
30	1203411932	107	1203412715	142	1203413318	137	1203414400	109	1203415355	220
31	1203411946	121	1203412726	153	1203413332	151	1203414402	111	1203415367	232
32	1203411947	122	1203412728	155	1203413332	151	1203414402	111	1203415369	234
33	1203411947	122	1203412731	158	1203413332	151	1203414414	123	1203415382	247
34	1203411949	124	1203412742	169	1203413346	165	1203414415	124	1203415385	250
35	1203411962	137	1203412744	171	1203413348	167	1203414416	125	1203415399	264
36	1203411962	137	1203412746	173	1203413349	168	1203414417	126	1203415399	264
37	1203411963	138	1203412756	183	1203413360	179	1203414429	138	1203415412	277
38	1203411976	151	1203412759	186	1203413362	181	1203414432	141	1203415416	281
39	1203411978	153	1203412762	189	1203413363	182	1203414432	141	1203415428	293
40	1203411978	153	1203412772	199	1203413376	195	1203414432	141	1203415429	294
41	1203411978	153	1203412774	201	1203413378	197	1203414442	151	1203415442	307
42	1203411992	167	1203412776	203	1203413379	198	1203414445	154	1203415446	311
43	1203411993	168	1203412785	212	1203413392	211	1203414447	156	1203415458	323
44	1203411993	168	1203412790	217	1203413393	212	1203414447	156	1203415462	327
45	1203412008	183	1203412792	219	1203413394	213	1203414459	168	1203415473	338
46	1203412008	183	1203412802	229	1203413406	225	1203414461	170	1203415489	354
47	1203412009	184	1203412804	231	1203413409	228	1203414462	171	1203415492	357
48	1203412022	197	1203412806	233	1203413422	241	1203414462	171	1203415499	364
49	1203412023	198	1203412817	244	1203413424	243	1203414473	182	1203415502	367
50	1203412023	198	1203412819	246	1203413436	255	1203414477	186	1203415504	369
51	1203412037	212	1203412821	248	1203413440	259	1203414477	186	1203415506	371
52	1203412038	213	1203412831	258	1203413453	272	1203414477	186	1203415512	377
53	1203412040	215	1203412835	262	1203413455	274	1203414489	198	1203415517	382
54	1203412054	229	1203412838	265	1203413468	287	1203414492	201	1203415520	385
55	1203412054	229	1203412847	274	1203413471	290	1203414492	201	1203415521	386
56	1203412054	229	1203412850	277	1203413482	301	1203414493	202	1203415529	394
57	1203412054	229	1203412853	280	1203413485	304	1203414505	214	1203415532	397
58	1203412034	244	1203412853	291	1203413498	317	1203414506	215	1203415534	399
59	1203412070	245	1203412865	292	1203413501	320	1203414509	218	1203415534	401
60	1203412070	258	1203412868	295	1203413501	333	1203414509	218	1203415536	410
61	1203412083	259	1203412808	305	1203413514	334	1203414509	227	1203415546	411
62	1203412084	259	1203412878	305	1203413515	348	1203414518	232	1203415549	411
63		273	1203412880	310	1203413529	350	1203414523	232	1203415549	414
64	1203412098	274	1203412883	320	1203413531	363	1203414524	233	1203415551	425
			1203412893							430
65	1203412114	275		322	1203413547	366 378	1203414534	243	1203415565	430
66	1203412114	289	1203412898	325	1203413559		1203414537	246	1203415567	
67	1203412114	289	1203412909	336	1203413561	380	1203414537	246	1203415574	439
68	1203412115	290	1203412911	338	1203413569	388	1203414538	247	1203415577	442
69	1203412129	304	1203412914	341	1203413569	388	1203414548	257	1203415579	444
70	1203412129	304	1203412925	352	1203413574	393	1203414553	262	1203415582	447
71	1203412130	305	1203412925	352	1203413576	395	1203414554	263	1203415589	454
72	1203412144	319	1203412929	356	1203413589	408	1203414554	263	1203415592	457

	ı	I.	ı	I.	ı	I.	I.	ı	I.	I.
73	1203412145	320	1203412934	361	1203413592	411	1203414565	274	1203415594	459
74	1203412146	321	1203412940	367	1203413604	423	1203414568	277	1203415597	462
75	1203412159	334	1203412941	368	1203413607	426	1203414569	278	1203415605	470
76	1203412160	335	1203412942	369	1203413615	434	1203414569	278	1203415607	472
77	1203412161	336	1203412950	377	1203413619	438	1203414580	289	1203415610	475
78	1203412173	348	1203412955	382	1203413622	441	1203414583	292	1203415612	477
79	1203412175	350	1203412956	383	1203413629	448	1203414584	293	1203415619	484
80	1203412175	350	1203412959	386	1203413635	454	1203414595	304	1203415622	487
81	1203412177	352	1203412965	392	1203413638	457	1203414599	308	1203415625	490
82	1203412190	365	1203412971	398	1203413644	463	1203414599	308	1203415627	492
83	1203412191	366	1203412971	398	1203413645	464	1203414599	308	1203415635	500
84	1203412206	381	1203412972	399	1203413650	469	1203414610	319	1203415637	502
85	1203412206	381	1203412980	407	1203413654	473	1203414613	322	1203415640	505
86	1203412219	394	1203412985	412	1203413661	480	1203414614	323	1203415643	508
87	1203412221	396	1203412986	413	1203413665	484	1203414616	325	1203415650	515
88	1203412235	410	1203412989	416	1203413668	487	1203414625	334	1203415653	518
89	1203412236	411	1203412996	423	1203413675	494	1203414628	337	1203415655	520
90	1203412248	423	1203413000	427	1203413680	499	1203414630	339	1203415658	523
91	1203412248	423	1203413001	428	1203413682	501	1203414640	349	1203415666	531
92	1203412250	425	1203413005	432	1203413692	511	1203414644	353	1203415668	533
93	1203412251	426	1203413012	439	1203413695	514	1203414645	354	1203415670	535
94	1203412262	437	1203413016	443	1203413748	567	1203414656	365	1203415672	537
95	1203412262	437	1203413016	443	1203413755	574	1203414709	418	1203415731	596
96	1203412262	437	1203413020	447	1203413760	579	1203414709	418	1203415733	598
97	1203412315	490	1203413076	503	1203413955	774	1203414721	430	1203415735	600
98	1203412317	492	1203413081	508	1203414020	839	1203414998	707	1203415743	608
99	1203412317	492	1203413081	508	1203414020	839	1203414998	707	1203415878	743
100	1203412382	557	1203413085	512	1203414086	905	1203415012	721	1203415935	800

BOING	No Work 8	8 Machin	es							
Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A
1	1202708459	0	1202709151	0	1202710169	0	1202710781	0	1202711553	0
2	1202708461	2	1202709154	3	1202710170	1	1202710782	1	1202711554	1
3	1202708461	2	1202709157	6	1202710171	2	1202710783	2	1202711555	2
4	1202708462	3	1202709158	7	1202710173	4	1202710783	2	1202711556	3
5	1202708462	3	1202709159	8	1202710173	4	1202710783	2	1202711556	3
6	1202708463	4	1202709159	8	1202710174	5	1202710786	5	1202711556	3
7	1202708463	4	1202709159	8	1202710176	7	1202710796	15	1202711557	4
8	1202708469	10	1202709165	14	1202710177	8	1202710796	15	1202711568	15
9	1202708473	14	1202709169	18	1202710184	15	1202710797	16	1202711569	16
10	1202708476	17	1202709172	21	1202710185	16	1202710798	17	1202711569	16
11	1202708476	17	1202709172	21	1202710186	17	1202710798	17	1202711570	17
12	1202708477	18	1202709173	22	1202710188	19	1202710801	20	1202711571	18
13	1202708478	19	1202709174	23	1202710188	19	1202710811	30	1202711571	18
14	1202708479	20	1202709175	24	1202710189	20	1202710812	31	1202711571	18
15	1202708483	24	1202709180	29	1202710190	21	1202710812	31	1202711583	30
16	1202708488	29	1202709184	33	1202710193	24	1202710813	32	1202711583	30
17	1202708490	31	1202709187	36	1202710200	31	1202710813	32	1202711585	32
18	1202708490	31	1202709188	37	1202710200	31	1202710817	36	1202711586	33
19	1202708491	32	1202709189	38	1202710202	33	1202710826	45	1202711586	33
20	1202708492	33	1202709190	39	1202710202	33	1202710827	46	1202711587	34
21	1202708493	34	1202709190	39	1202710204	35	1202710827	46	1202711595	42
22	1202708494	35	1202709195	44	1202710205	36	1202710828	47	1202711595	42
23	1202708498	39	1202709199	48	1202710205	36	1202710829	48	1202711597	44
24	1202708505	46	1202709202	51	1202710208	39	1202710832	51	1202711599	46
25	1202708506	47	1202709203	52	1202710215	46	1202710841	60	1202711600	47
26	1202708506	47	1202709203	52	1202710216	47	1202710842	61	1202711601	48
27	1202708506	47	1202709206	55	1202710217	48	1202710843	62	1202711602	49
28	1202708507	48	1202709206	55	1202710217	48	1202710844	63	1202711602	49

20	1202708507	18	1202700211	60	1202710220	51	1202710945	64	1202711612	59
29	1202708507	48	1202709211	60	1202710220	51	1202710845 1202710847	64	1202711612	-
30				63				66	1202711614	61
31	1202708515	56	1202709217	66	1202710221	52	1202710856	75	1202711616	63
32	1202708520	61	1202709218	67	1202710224	55	1202710857	76	1202711617	64
33	1202708521	62	1202709219	68	1202710230	61	1202710858	77	1202711617	64
34	1202708521	62	1202709219	68	1202710231	62	1202710859	78	1202711617	64
35	1202708521	62	1202709222	71	1202710232	63	1202710859	78	1202711625	72
36	1202708521	62	1202709226	75	1202710233	64	1202710862	81	1202711627	74
37	1202708523	64	1202709229	78	1202710235	66	1202710872	91	1202711631	78
38	1202708524	65	1202709232	81	1202710236	67	1202710872	91	1202711631	78
39	1202708530	71	1202709233	82	1202710238	69	1202710873	92	1202711631	78
40	1202708535	76	1202709233	82	1202710239	70	1202710874	93	1202711632	79
41	1202708536	77	1202709236	85	1202710245	76	1202710875	94	1202711641	88
42	1202708536	77	1202709236	85	1202710246	77	1202710878	97	1202711643	90
43	1202708537	78	1202709241	90	1202710248	79	1202710887	106	1202711643	90
44	1202708538	79	1202709246	95	1202710248	79	1202710887	106	1202711643	90
45	1202708538	79	1202709247	96	1202710249	80	1202710888	107	1202711645	92
46	1202708539	80	1202709248	97	1202710250	81	1202710889	108	1202711647	94
47	1202708545	86	1202709249	98	1202710254	85	1202710889	108	1202711647	94
48	1202708550	91	1202709250	99	1202710259	90	1202710893	112	1202711647	94
49	1202708551	92	1202709251	100	1202710262	93	1202710902	121	1202711655	102
50	1202708552	93	1202709256	105	1202710262	93	1202710903	122	1202711658	105
51	1202708553	94	1202709259	108	1202710263	94	1202710903	122	1202711661	108
52	1202708553	94	1202709262	111	1202710265	96	1202710905	124	1202711661	108
53	1202708554	95	1202709264	113	1202710266	97	1202710905	124	1202711662	109
54	1202708558	99	1202709265	114	1202710266	97	1202710908	127	1202711662	109
55	1202708565	106	1202709266	115	1202710269	100	1202710908	137	1202711670	117
56	1202708567	108	1202709267	116	1202710209	107	1202710918	137	1202711670	119
57	1202708568	109	1202709207	121	1202710276	107	1202710918	137	1202711672	120
58	1202708568	109	1202709272	123	1202710270	107	1202710918	139	1202711675	123
		109								123
59	1202708568		1202709279	128	1202710279	110	1202710921	140	1202711677	
60	1202708568	109	1202709279	128	1202710279	110	1202710923	142	1202711678	125
61	1202708575	116	1202709279	128	1202710280	111	1202710933	152	1202711678	125
62	1202708581	122	1202709281	130	1202710281	112	1202710933	152	1202711686	133
63	1202708583	124	1202709283	132	1202710290	121	1202710934	153	1202711687	134
64	1202708583	124	1202709288	137	1202710292	123	1202710935	154	1202711688	135
65	1202708583	124	1202709289	138	1202710293	124	1202710936	155	1202711690	137
66	1202708584	125	1202709294	143	1202710294	125	1202710939	158	1202711692	139
67	1202708585	126	1202709294	143	1202710296	127	1202710948	167	1202711693	140
68	1202708590	131	1202709295	144	1202710296	127	1202710948	167	1202711693	140
69	1202708597	138	1202709297	146	1202710297	128	1202710949	168	1202711700	147
70	1202708597	138	1202709297	146	1202710299	130	1202710950	169	1202711702	149
71	1202708597	138	1202709302	151	1202710306	137	1202710954	173	1202711703	150
72	1202708598	139	1202709305	154	1202710307	138	1202710963	182	1202711705	152
73	1202708599	140	1202709308	157	1202710308	139	1202710963	182	1202711707	154
74	1202708600	141	1202709309	158	1202710308	139	1202710964	183	1202711708	155
75	1202708604	145	1202709311	160	1202710311	142	1202710964	183	1202711708	155
76	1202708611	152	1202709311	160	1202710311	142	1202710965	184	1202711716	163
77	1202708612	153	1202709312	161	1202710313	144	1202710969	188	1202711719	166
78	1202708612	153	1202709317	166	1202710314	145	1202710979	198	1202711719	166
79	1202708614	155	1202709320	169	1202710321	152	1202710979	198	1202711722	169
80	1202708614	155	1202709324	173	1202710322	153	1202710980	199	1202711723	170
81	1202708614	155	1202709324	173	1202710323	154	1202710980	199	1202711723	170
82	1202708621	162	1202709326	175	1202710324	155	1202710980	199	1202711723	170
83	1202708627	168	1202709326	175	1202710324	157	1202710984	203	1202711723	180
84	1202708628	169	1202709328	177	1202710320	158	1202710984	212	1202711733	180
85	1202708628	169	1202709328	181	1202710327	158	1202710993	213	1202711733	181
86	1202708628	170	1202709332	184	1202710327	161	1202710994	213	1202711734	185
87	1202708630	171	1202709339	188	1202710336	167	1202710995	214	1202711738	185

88	1202708630	171	1202709339	188	1202710338	169	1202710997	216	1202711738	185
89	1202708635	176	1202709339	188	1202710339	170	1202711000	219	1202711738	185
90	1202708642	183	1202709342	191	1202710340	171	1202711009	228	1202711747	194
91	1202708693	234	1202709344	193	1202710391	222	1202711010	229	1202711748	195
92	1202708693	234	1202709348	197	1202710391	222	1202711059	278	1202711750	197
93	1202708694	235	1202709401	250	1202710392	223	1202711061	280	1202711802	249
94	1202708694	235	1202709404	253	1202710395	226	1202711061	280	1202711803	250
95	1202708694	235	1202709405	254	1202710402	233	1202711070	289	1202711803	250
96	1202708700	241	1202709405	254	1202710403	234	1202711074	293	1202711803	250
97	1202708707	248	1202709407	256	1202710405	236	1202711075	294	1202711812	259
98	1202708842	383	1202709408	257	1202710405	236	1202711191	410	1202711813	260
99	1202708918	459	1202709414	263	1202710657	488	1202711196	415	1202711814	261
100	1202708918	459	1202709561	410	1202710664	495	1202711418	637	1202711972	419

Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A
1	1202629982	0	1202635618	0	1202636229	0	1202705843	0	1202706992	0
2	1202629983	1	1202635618	0	1202636230	1	1202705843	0	1202706993	1
3	1202629983	1	1202635618	0	1202636231	2	1202705844	1	1202706993	1
4	1202629983	1	1202635618	0	1202636231	2	1202705844	1	1202706993	1
5	1202629983	1	1202635618	0	1202636232	3	1202705846	3	1202706994	2
6	1202629983	1	1202635620	2	1202636232	3	1202705846	3	1202706995	3
7	1202629984	2	1202635620	2	1202636233	4	1202705846	3	1202706995	3
8	1202629984	2	1202635621	3	1202636234	5	1202705846	3	1202706995	3
9	1202629984	2	1202635622	4	1202636236	7	1202705847	4	1202706996	4
10	1202629984	2	1202635622	4	1202636237	8	1202705848	5	1202706997	5
11	1202629984	2	1202635623	5	1202636237	8	1202705848	5	1202706998	6
12	1202629984	2	1202635624	6	1202636237	8	1202705848	5	1202706998	6
13	1202629984	2	1202635625	7	1202636238	9	1202705848	5	1202706999	7
14	1202629985	3	1202635626	8	1202636238	9	1202705849	6	1202706999	7
15	1202629988	6	1202635626	8	1202636244	15	1202705850	7	1202707000	8
16	1202629988	6	1202635626	8	1202636244	15	1202705851	8	1202707002	10
17	1202629998	16	1202635632	14	1202636245	16	1202705857	14	1202707007	15
18	1202629998	16	1202635633	15	1202636247	18	1202705858	15	1202707007	15
19	1202629998	16	1202635633	15	1202636247	18	1202705858	15	1202707007	15
20	1202629998	16	1202635634	16	1202636248	19	1202705859	16	1202707008	16
21	1202629998	16	1202635634	16	1202636248	19	1202705862	19	1202707009	17
22	1202629999	17	1202635634	16	1202636249	20	1202705862	19	1202707010	18
23	1202629999	17	1202635635	17	1202636252	23	1202705862	19	1202707010	18
24	1202629999	17	1202635637	19	1202636252	23	1202705862	19	1202707011	19
25	1202629999	17	1202635637	19	1202636253	24	1202705862	19	1202707011	19
26	1202629999	17	1202635638	20	1202636253	24	1202705863	20	1202707012	20
27	1202629999	17	1202635638	20	1202636253	24	1202705863	20	1202707013	21
28	1202629999	17	1202635639	21	1202636253	24	1202705864	21	1202707014	22
29	1202629999	17	1202635639	21	1202636260	31	1202705864	21	1202707014	22
30	1202630004	22	1202635639	21	1202636260	31	1202705865	22	1202707014	22
31	1202630004	22	1202635641	23	1202636261	32	1202705869	26	1202707016	24
32	1202630013	31	1202635642	24	1202636261	32	1202705874	31	1202707017	25
33	1202630013	31	1202635648	30	1202636262	33	1202705874	31	1202707022	30
34	1202630013	31	1202635648	30	1202636262	33	1202705874	31	1202707023	31
35	1202630014	32	1202635648	30	1202636263	34	1202705874	31	1202707023	31
36	1202630014	32	1202635649	31	1202636263	34	1202705876	33	1202707024	32
37	1202630014	32	1202635649	31	1202636265	36	1202705877	34	1202707025	33
38	1202630014	32	1202635650	32	1202636267	38	1202705877	34	1202707025	33
39	1202630014	32	1202635651	33	1202636268	39	1202705877	34	1202707026	34
40	1202630014	32	1202635652	34	1202636268	39	1202705877	34	1202707028	36
41	1202630014	32	1202635652	34	1202636268	39	1202705878	35	1202707028	36
42	1202630015	33	1202635652	34	1202636268	39	1202705878	35	1202707028	36
43	1202630015	33	1202635652	34	1202636274	45	1202705879	36	1202707028	36

L44	1202(20018	126	1202625655	27	1202626274	45	1202705870	120	1202707020	27
44	1202630018	36	1202635655	37	1202636274	45	1202705879	36	1202707029	37
45	1202630018	36	1202635655	37	1202636277	48	1202705879	36	1202707029	37
46	1202630018	36	1202635656	38	1202636277	48	1202705879	36	1202707030	38
47	1202630019	37	1202635656	38	1202636277	48	1202705886	43	1202707031	39
48	1202630029	47	1202635657	39	1202636277	48	1202705888	45	1202707038	46
49	1202630029	47	1202635662	44	1202636278	49	1202705888	45	1202707039	47
50	1202630029	47	1202635663	45	1202636278	49	1202705888	45	1202707039	47
51	1202630029	47	1202635663	45	1202636280	51	1202705889	46	1202707039	47
52	1202630030	48	1202635663	45	1202636282	53	1202705891	48	1202707040	48
53	1202630030	48	1202635663	45	1202636283	54	1202705892	49	1202707040	48
54	1202630030	48	1202635665	47	1202636283	54	1202705892	49	1202707041	49
55	1202630030	48	1202635666	48	1202636283	54	1202705892	49	1202707041	49
56	1202630030	48	1202635667	49	1202636284	55	1202705892	49	1202707042	50
57	1202630031	49	1202635667	49	1202636289	60	1202705893	50	1202707044	52
58	1202630031	49	1202635667	49	1202636290	61	1202705893	50	1202707044	52
59	1202630034	52	1202635668	50	1202636291	62	1202705894	51	1202707044	52
60	1202630034	52	1202635669	51	1202636292	63	1202705895	52	1202707045	53
61	1202630036	54	1202635669	51	1202636293	64	1202705895	52	1202707045	53
62	1202630038	56	1202635671	53	1202636293	64	1202705895	52	1202707046	54
63	1202630038	56	1202635672	54	1202636293	64	1202705902	59	1202707052	60
64	1202630043	61	1202635672	54	1202636294	65	1202705902	59	1202707053	61
65	1202630044	62	1202635678	60	1202636297	68	1202705902	59	1202707054	62
66	1202630045	63	1202635678	60	1202636298	69	1202705904	61	1202707054	62
67	1202630045	63	1202635679	61	1202636298	69	1202705904	61	1202707055	63
68	1202630051	69	1202635679	61	1202636298	69	1202705906	63	1202707055	63
69	1202630051	69	1202635679	61	1202636299	70	1202705906	63	1202707057	65
70	1202630051	69	1202635680	62	1202636305	76	1202705907	64	1202707058	66
71	1202630052	70	1202635680	62	1202636305	76	1202705907	64	1202707058	66
72	1202630053	71	1202635682	64	1202636307	78	1202705908	65	1202707059	67
73	1202630054	72	1202635683	65	1202636307	78	1202705908	65	1202707059	67
74	1202630054	72	1202635683	65	1202636307	78	1202705909	66	1202707059	67
75	1202630055	73	1202635685	67	1202636308	79	1202705909	66	1202707060	68
76	1202630057	75	1202635685	67	1202636309	80	1202705910	67	1202707060	68
77	1202630058	76	1202635687	69	1202636309	80	1202705910	67	1202707060	68
78	1202630059	77	1202635687	69	1202636311	82	1202705911	68	1202707061	69
79	1202630059	77	1202635687	69	1202636312	83	1202705916	73	1202707067	75
80	1202630059	77	1202635692	74	1202636313	84	1202705917	74	1202707068	76
81	1202630061	79	1202635693	75	1202636314	85	1202705918	75	1202707068	76
82	1202630061	79	1202635693	75	1202636314	85	1202705919	76	1202707069	77
83	1202630111	129	1202635694	76	1202636315	86	1202705920	77	1202707069	77
84	1202630115	133	1202635743	125	1202636371	142	1202705920	77	1202707120	128
85	1202630117	135	1202635744	126	1202636371	142	1202705971	128	1202707121	129
86	1202630119	137	1202635745	127	1202636371	142	1202705972	129	1202707122	130
87	1202630119	137	1202635746	128	1202636372	143	1202705972	129	1202707122	130
88	1202630121	139	1202635747	129	1202636374	145	1202705973	130	1202707124	132
89	1202630121	139	1202635748	130	1202636375	146	1202705973	130	1202707125	133
90	1202630122	140	1202635748	130	1202636377	148	1202705973	130	1202707126	134
91	1202630123	141	1202635749	131	1202636377	148	1202705974	131	1202707126	134
92	1202630124	142	1202635750	132	1202636378	149	1202705974	131	1202707133	141
93	1202630124	142	1202635751	133	1202636378	149	1202705975	132	1202707133	141
94	1202630124	142	1202635751	133	1202636378	149	1202705976	133	1202707133	141
95	1202630124	142	1202635758	140	1202636378	149	1202705976	133	1202707134	142
96	1202630125	143	1202635758	140	1202636379	150	1202705981	138	1202707134	142
97	1202630125	143	1202635759	141	1202636641	412	1202705984	141	1202707410	418
98	1202630128	146	1202635763	145	1202636642	413	1202705984	141	1202707410	418
99	1202630460	478	1202636087	469	1202636712	483	1202705988	145	1202707458	466
100	1202630467	485	1202636094	476	1202636712	483	1202706296	453	1202707610	618
		1	1	l .		l .		1	1	1

CompTorrent

CompT	Forrent No V	Vork 1 Ma	achine							
Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A
1	1206520611	0	1206521176	0	1206521318	0	1206521480	0	1206521621	0
<u> </u>	1206520612	1	1206521178	2	1206521319	1	1206521482	2	1206521622	1
3	1206520613	2	1206521178	2	1206521320	2	1206521483	3	1206521623	2
4	1206520615	4	1206521179	3	1206521321	3	1206521483	3	1206521625	4
5	1206520616	5	1206521181	5	1206521321	3	1206521485	5	1206521626	5
6	1206520616	5	1206521182	6	1206521323	5	1206521486	6	1206521627	6
7	1206520618	7	1206521183	7	1206521324	6	1206521487	7	1206521628	7
8	1206520618	7	1206521184	8	1206521325	7	1206521488	8	1206521629	8
))	1206520619	8	1206521185	9	1206521326	8	1206521489	9	1206521630	9
10	1206520621	10	1206521186	10	1206521327	9	1206521490	10	1206521631	10
11	1206520622	11		11	1206521328	10	1206521491	11		11
12		12	1206521187	12		11		12	1206521632	12
	1206520623		1206521188		1206521329		1206521492		1206521633	
13	1206520624	13	1206521189	13	1206521330	12	1206521493	13	1206521634	13
14	1206520624	13	1206521190	14	1206521331	13	1206521494	14	1206521635	14
15	1206520626	15	1206521191	15	1206521333	15	1206521495	15	1206521635	14
16	1206520627	16	1206521192	16	1206521334	16	1206521496	16	1206521637	16
17	1206520627	16	1206521193	17	1206521335	17	1206521497	17	1206521638	17
18	1206520629	18	1206521195	19	1206521336	18	1206521498	18	1206521638	17
.9	1206520630	19	1206521195	19	1206521337	19	1206521499	19	1206521640	19
20	1206520631	20	1206521196	20	1206521339	21	1206521501	21	1206521641	20
21	1206520632	21	1206521197	21	1206521340	22	1206521502	22	1206521642	21
22	1206520633	22	1206521198	22	1206521341	23	1206521503	23	1206521643	22
23	1206520634	23	1206521199	23	1206521342	24	1206521504	24	1206521644	23
24	1206520635	24	1206521201	25	1206521343	25	1206521505	25	1206521645	24
25	1206520636	25	1206521202	26	1206521344	26	1206521506	26	1206521646	25
26	1206520637	26	1206521202	26	1206521345	27	1206521507	27	1206521648	27
27	1206520638	27	1206521204	28	1206521346	28	1206521508	28	1206521648	27
28	1206520639	28	1206521205	29	1206521347	29	1206521509	29	1206521649	28
29	1206520640	29	1206521206	30	1206521348	30	1206521511	31	1206521650	29
30	1206520641	30	1206521207	31	1206521349	31	1206521512	32	1206521651	30
31	1206520641	30	1206521208	32	1206521350	32	1206521513	33	1206521652	31
32	1206520643	32	1206521209	33	1206521352	34	1206521514	34	1206521653	32
33	1206520644	33	1206521210	34	1206521353	35	1206521515	35	1206521654	33
34	1206520644	33	1206521211	35	1206521353	35	1206521516	36	1206521655	34
35	1206520646	35	1206521212	36	1206521355	37	1206521517	37	1206521657	36
36	1206520646	35	1206521213	37	1206521355	37	1206521518	38	1206521658	37
37	1206520648	37	1206521214	38	1206521356	38	1206521519	39	1206521658	37
38	1206520649	38	1206521215	39	1206521357	39	1206521520	40	1206521660	39
39	1206520650	39	1206521216	40	1206521358	40	1206521521	41	1206521661	40
10	1206520651	40	1206521217	41	1206521359	41	1206521522	42	1206521662	41
41	1206520652	41	1206521218	42	1206521360	42	1206521523	43	1206521663	42
12	1206520653	42	1206521219	43	1206521360	44	1206521524	44	1206521664	43
13	1206520654	43	1206521220	44	1206521362	44	1206521525	45	1206521665	44
14	1206520655	44	1206521221	45	1206521363	45	1206521526	46	1206521666	45
14 15	1206520656	45		45		46		47	1206521666	45
			1206521221		1206521364		1206521527			
6	1206520657	46	1206521223	47	1206521365	47	1206521528	48	1206521667	46
7	1206520658	47	1206521224	48	1206521366	48	1206521529	49	1206521668	47
8	1206520659	48	1206521225	49	1206521368	50	1206521530	50	1206521669	48
19	1206520660	49	1206521226	50	1206521369	51	1206521531	51	1206521670	49
50	1206520660	49	1206521227	51	1206521370	52	1206521533	53	1206521671	50
51	1206520662	51	1206521228	52	1206521371	53	1206521533	53	1206521672	51
52	1206520662	51	1206521229	53	1206521372	54	1206521534	54	1206521674	53
53	1206520664	53	1206521230	54	1206521373	55	1206521536	56	1206521675	54
54	1206520665	54	1206521231	55	1206521374	56	1206521537	57	1206521676	55
55	1206520665	54	1206521231	55	1206521375	57	1206521538	58	1206521677	56

56	1206520667	56	1206521233	57	1206521375	57	1206521539	59	1206521678	57
57	1206520668	57	1206521234	58	1206521377	59	1206521539	59	1206521679	58
58	1206520669	58	1206521235	59	1206521377	59	1206521539	61	1206521680	59
59	1206520670	59	1206521236	60	1206521379	61	1206521541	62	1206521681	60
60	1206520671	60	1206521237	61	1206521380	62	1206521542	62	1206521682	61
61	1206520672	61	1206521238	62	1206521380	62	1206521543	63	1206521684	63
62	1206520673	62	1206521239	63	1206521382	64		64	1206521685	64
							1206521544			
63	1206520674	63	1206521240	64	1206521383	65	1206521545	65	1206521685	64
64	1206520675	64	1206521242	66	1206521383	65	1206521546	66	1206521687	66
65	1206520676	65	1206521243	67	1206521385	67	1206521548	68	1206521688	67
66	1206520677	66	1206521244	68	1206521386	68	1206521549	69	1206521689	68
67	1206520678	67	1206521245	69	1206521387	69	1206521550	70	1206521690	69
68	1206520679	68	1206521246	70	1206521388	70	1206521551	71	1206521691	70
69	1206520680	69	1206521247	71	1206521389	71	1206521552	72	1206521692	71
70	1206520681	70	1206521248	72	1206521390	72	1206521553	73	1206521693	72
71	1206520682	71	1206521249	73	1206521391	73	1206521554	74	1206521694	73
72	1206520683	72	1206521250	74	1206521392	74	1206521555	75	1206521695	74
73	1206520685	74	1206521251	75	1206521393	75	1206521556	76	1206521697	76
74	1206520685	74	1206521252	76	1206521394	76	1206521558	78	1206521697	76
75	1206520686	75	1206521253	77	1206521395	77	1206521559	79	1206521698	77
76	1206520688	77	1206521254	78	1206521396	78	1206521560	80	1206521700	79
77	1206520688	77	1206521256	80	1206521397	79	1206521561	81	1206521701	80
78	1206520690	79	1206521257	81	1206521398	80	1206521562	82	1206521702	81
79	1206520690	79	1206521258	82	1206521399	81	1206521563	83	1206521703	82
80	1206520691	80	1206521259	83	1206521400	82	1206521564	84	1206521704	83
81	1206520693	82	1206521260	84	1206521401	83	1206521565	85	1206521705	84
82	1206520693	82	1206521261	85	1206521402	84	1206521566	86	1206521706	85
83	1206520695	84	1206521262	86	1206521404	86	1206521567	87	1206521707	86
84	1206520696	85	1206521263	87	1206521405	87	1206521568	88	1206521708	87
85	1206520697	86	1206521264	88	1206521405	87	1206521569	89	1206521710	89
86	1206520698	87	1206521265	89	1206521406	88	1206521570	90	1206521711	90
87	1206520699	88	1206521267	91	1206521408	90	1206521571	91	1206521712	91
88	1206520700	89	1206521268	92	1206521409	91	1206521572	92	1206521712	91
89	1206520701	90	1206521269	93	1206521410	92	1206521573	93	1206521714	93
90	1206520702	91	1206521270	94	1206521411	93	1206521574	94	1206521714	93
91	1206520703	92	1206521270	95	1206521411	94	1206521574	95	1206521715	94
92	1206520704	93	1206521272	96	1206521413	95	1206521576	96	1206521717	96
93	1206520705	94	1206521273	97	1206521415	97	1206521570	97	1206521717	96
94	1206520706	95	1206521274	98	1206521416	98	1206521578	98	1206521717	97
95	1206520707	96	1206521274	100	1206521417	99	1206521579	99	1206521718	99
96	1206520707	98	1206521277	101	1206521417	100	1206521580	100	1206521721	100
97	1206520710 1206520711	99	1206521278	102	1206521419	101	1206521581	101	1206521722	101
98		100		103	1206521421	103	1206521582	102	1206521723	102
99	1206520712	101	1206521280	104	1206521421	103	1206521583	103	1206521724	103
100	1206520713	102	1206521281	105	1206521422	104	1206521584	104	1206521725	104

CompTo	rrent No V	Vork 2 Ma	chines							
Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A
1	1206521784	0	1206521886	0	1206522023	0	1206522125	0	1206522219	0
2	1206521784	0	1206521887	1	1206522023	0	1206522125	0	1206522219	0
3	1206521785	1	1206521888	2	1206522024	1	1206522126	1	1206522220	1
4	1206521785	1	1206521889	3	1206522024	1	1206522126	1	1206522221	2
5	1206521786	2	1206521890	4	1206522025	2	1206522126	1	1206522221	2
6	1206521786	2	1206521890	4	1206522026	3	1206522127	2	1206522222	3
7	1206521787	3	1206521892	6	1206522026	3	1206522127	2	1206522223	4
8	1206521788	4	1206521893	7	1206522027	4	1206522128	3	1206522223	4
9	1206521788	4	1206521893	7	1206522028	5	1206522129	4	1206522224	5
10	1206521788	4	1206521894	8	1206522028	5	1206522129	4	1206522224	5
11	1206521789	5	1206521895	9	1206522029	6	1206522130	5	1206522225	6

12	1206521700	5	1206521007	l.,	1206522020	6	1206522121	6	120652222	6
12	1206521789	5	1206521897	11	1206522029	6	1206522131	6	1206522225	6
13	1206521790	6	1206521898	12	1206522030	7	1206522131	7	1206522226	7
14	1206521790	6	1206521898	12	1206522031	8	1206522132		1206522226	7
15	1206521791 1206521791	7	1206521899	13	1206522031	9	1206522132	8	1206522227	9
16			1206521900		1206522032	9	1206522133		1206522228	9
17	1206521792 1206521792	8	1206521900 1206521901	14	1206522032	10	1206522133 1206522134	9	1206522228 1206522229	10
		9						9	1206522229	
19	1206521793	9	1206521901	15	1206522033	10	1206522134			10
20	1206521793 1206521794	10	1206521903 1206521902	17	1206522034	11	1206522135 1206522135	10	1206522230	11
22	1206521794	10	1206521902	17	1206522034	12		11		12
23	1206521794	11	1206521905	19	1206522035	12	1206522136 1206522136	11	1206522231 1206522231	12
24	1206521796	12	1206521904	18	1206522035	12	1206522137	12	1206522232	13
25	1206521796	12	1206521905	19	1206522036	13	1206522138	13	1206522232	13
26	1206521797	13	1206521906	20	1206522037	14	1206522138	13	1206522233	14
27	1206521798	14	1206521906	20	1206522037	15	1206522138	13	1206522234	15
28	1206521798	14	1206521907	21	1206522038	15	1206522140	15	1206522234	15
29	1206521799	15	1206521908	22	1206522039	16	1206522140	15	1206522235	16
30	1206521799	15	1206521908	22	1206522039	16	1206522140	15	1206522236	17
31	1206521800	16	1206521909	23	1206522040	17	1206522141	16	1206522236	17
32	1206521800	16	1206521909	24	1206522040	18	1206522141	17	1206522237	18
33	1206521801	17	1206521910	24	1206522041	18	1206522142	17	1206522237	18
34	1206521801	17	1206521911	25	1206522041	19	1206522142	18	1206522238	19
35	1206521802	18	1206521912	26	1206522043	20	1206522143	18	1206522239	20
36	1206521803	19	1206521913	27	1206522043	20	1206522144	19	1206522239	20
37	1206521803	19	1206521913	27	1206522044	21	1206522144	19	1206522240	21
38	1206521804	20	1206521914	28	1206522044	21	1206522145	20	1206522240	21
39	1206521804	20	1206521914	28	1206522045	22	1206522145	20	1206522241	22
40	1206521805	21	1206521915	29	1206522046	23	1206522147	22	1206522242	23
41	1206521805	21	1206521916	30	1206522046	23	1206522147	22	1206522242	23
42	1206521806	22	1206521916	30	1206522047	24	1206522148	23	1206522243	24
43	1206521807	23	1206521917	31	1206522048	25	1206522148	23	1206522244	25
44	1206521807	23	1206521917	31	1206522048	25	1206522149	24	1206522244	25
45	1206521808	24	1206521918	32	1206522049	26	1206522149	24	1206522245	26
46	1206521808	24	1206521919	33	1206522049	26	1206522150	25	1206522245	26
47	1206521809	25	1206521919	33	1206522050	27	1206522150	25	1206522246	27
48	1206521809	25	1206521920	34	1206522050	27	1206522151	26	1206522247	28
49	1206521810	26	1206521920	34	1206522051	28	1206522152	27	1206522247	28
50	1206521810	26	1206521921	35	1206522052	29	1206522153	28	1206522248	29
51	1206521811	27	1206521922	36	1206522052	29	1206522153	28	1206522248	29
52	1206521811	27	1206521922	36	1206522053	30	1206522154	29	1206522249	30
53	1206521812	28	1206521923	37	1206522053	30	1206522154	29	1206522250	31
54	1206521813	29	1206521923	37	1206522054	31	1206522155	30	1206522250	31
55	1206521813	29	1206521924	38	1206522055	32	1206522155	30	1206522251	32
56	1206521814	30	1206521924	38	1206522055	32	1206522156	31	1206522251	32
57	1206521814	30	1206521926	40	1206522056	33	1206522157	32	1206522252	33
58	1206521815	31	1206521925	39	1206522056	33	1206522157	32	1206522253	34
59	1206521815	31	1206521926	40	1206522057	34	1206522158	33	1206522253	34
60	1206521816	32	1206521927	41	1206522057	34	1206522158	33	1206522254	35
61	1206521816	32	1206521928	42	1206522058	35	1206522159	34	1206522254	35
62	1206521817	33	1206521928	42	1206522058	35	1206522160	35	1206522255	36
63	1206521817	33	1206521929	43	1206522059	36	1206522160	35	1206522255	36
64	1206521818	34	1206521930	44	1206522059	36	1206522161	36	1206522256	37
65	1206521819	35	1206521930	44	1206522060	37	1206522162	37	1206522256	37
66	1206521819	35	1206521932	46	1206522060	37	1206522162	37	1206522257	38
67	1206521820	36	1206521931	45	1206522061	38	1206522163	38	1206522258	39
68	1206521821	37	1206521933	47	1206522061	38	1206522164	39	1206522258	39
69	1206521821	37	1206521933	47	1206522063	40	1206522164	39	1206522259	40
70	1206521822	38	1206521934	48	1206522063	40	1206522165	40	1206522260	41
	-	1		l		l		<u> </u>		L

100	1206521839	55	1206521952	66	1206522080	57	1206522183	58	1206522277	58
99	1206521839	55	1206521952	66	1206522080	57	1206522182	57	1206522277	58
98	1206521838	54	1206521951	65	1206522079	56	1206522182	57	1206522276	57
97	1206521837	53	1206521950	64	1206522078	55	1206522181	56	1206522276	57
96	1206521837	53	1206521950	64	1206522078	55	1206522180	55	1206522275	56
95	1206521836	52	1206521949	63	1206522077	54	1206522180	55	1206522274	55
94	1206521836	52	1206521949	63	1206522077	54	1206522179	54	1206522274	55
93	1206521835	51	1206521948	62	1206522076	53	1206522178	53	1206522273	54
92	1206521835	51	1206521947	61	1206522076	53	1206522178	53	1206522272	53
91	1206521834	50	1206521947	61	1206522075	52	1206522177	52	1206522272	53
90	1206521834	50	1206521946	60	1206522074	51	1206522176	51	1206522271	52
89	1206521833	49	1206521945	59	1206522074	51	1206522176	51	1206522271	52
88	1206521832	48	1206521945	59	1206522074	51	1206522175	50	1206522270	51
87	1206521832	48	1206521944	58	1206522073	50	1206522175	50	1206522269	50
86	1206521831	47	1206521944	58	1206522072	49	1206522174	49	1206522269	50
85	1206521831	47	1206521943	57	1206522072	49	1206522173	48	1206522268	49
84	1206521830	46	1206521942	56	1206522071	48	1206522173	48	1206522268	49
83	1206521829	45	1206521942	56	1206522071	48	1206522172	47	1206522267	48
82	1206521829	45	1206521941	55	1206522070	47	1206522171	46	1206522266	47
81	1206521828	44	1206521940	54	1206522069	46	1206522170	45	1206522266	47
80	1206521827	43	1206521940	54	1206522069	46	1206522171	46	1206522266	47
79	1206521827	43	1206521939	53	1206522068	45	1206522169	44	1206522265	46
78	1206521826	42	1206521939	53	1206522067	44	1206522169	44	1206522264	45
77	1206521826	42	1206521938	52	1206522067	44	1206522168	43	1206522264	45
76	1206521825	41	1206521937	51	1206522066	43	1206522168	43	1206522263	44
75	1206521825	41	1206521937	51	1206522066	43	1206522167	42	1206522262	43
74	1206521824	40	1206521936	50	1206522065	42	1206522167	42	1206522262	43
73	1206521824	40	1206521935	49	1206522065	42	1206522166	41	1206522261	42
72	1206521823	39	1206521935	49	1206522064	41	1206522166	41	1206522261	42
71	1206521822	38	1206521934	48	1206522064	41	1206522165	40	1206522259	40

СотрТо	rrent No V	Vork 4 Ma	chines							
Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A
1	1206522350	0	1206522440	0	1206522531	0	1206522617	0	1206522711	0
2	1206522350	0	1206522440	0	1206522531	0	1206522618	1	1206522711	0
3	1206522351	1	1206522442	2	1206522532	1	1206522619	2	1206522712	1
4	1206522351	1	1206522441	1	1206522532	1	1206522620	3	1206522712	1
5	1206522352	2	1206522443	3	1206522533	2	1206522620	3	1206522713	2
6	1206522353	3	1206522443	3	1206522534	3	1206522621	4	1206522713	2
7	1206522352	2	1206522444	4	1206522534	3	1206522620	3	1206522714	3
8	1206522353	3	1206522444	4	1206522534	3	1206522621	4	1206522714	3
9	1206522354	4	1206522445	5	1206522535	4	1206522621	4	1206522715	4
10	1206522355	5	1206522445	5	1206522536	5	1206522622	5	1206522715	4
11	1206522354	4	1206522446	6	1206522536	5	1206522622	5	1206522716	5
12	1206522356	6	1206522446	6	1206522537	6	1206522623	6	1206522717	6
13	1206522355	5	1206522447	7	1206522536	5	1206522623	6	1206522718	7
14	1206522355	5	1206522447	7	1206522537	6	1206522624	7	1206522717	6
15	1206522357	7	1206522448	8	1206522538	7	1206522624	7	1206522718	7
16	1206522356	6	1206522448	8	1206522538	7	1206522624	7	1206522719	8
17	1206522357	7	1206522449	9	1206522538	7	1206522625	8	1206522719	8
18	1206522358	8	1206522449	9	1206522539	8	1206522625	8	1206522720	9
19	1206522358	8	1206522450	10	1206522539	8	1206522625	8	1206522720	9
20	1206522359	9	1206522450	10	1206522539	8	1206522626	9	1206522721	10
21	1206522358	8	1206522451	11	1206522540	9	1206522626	9	1206522722	11
22	1206522359	9	1206522452	12	1206522540	9	1206522626	9	1206522723	12
23	1206522361	11	1206522452	12	1206522541	10	1206522627	10	1206522723	12
24	1206522360	10	1206522453	13	1206522541	10	1206522628	11	1206522723	12
25	1206522361	11	1206522454	14	1206522541	10	1206522627	10	1206522724	13

26	1206522261	l	1206522454	L ₁₄	1206522542	11	1206522620	12	1206522724	12
26	1206522361	11	1206522454	14	1206522542	11	1206522629	12	1206522724	13
27	1206522361	11	1206522453	13	1206522542	11	1206522628	11	1206522724	13
28	1206522362	12	1206522454	14	1206522543	12	1206522629	12	1206522724	13
29	1206522364	14	1206522455	15	1206522543	12	1206522629	12	1206522725	14
30	1206522363	13	1206522455	15	1206522544	13	1206522629	12	1206522726	15
31	1206522363	13	1206522456	16	1206522543	12	1206522631	14	1206522726	15
32	1206522364	14	1206522457	17	1206522545	14	1206522631	14	1206522726	15
33	1206522364	14	1206522456	16	1206522545	14	1206522631	14	1206522726	15
34	1206522364	14	1206522456	16	1206522547	16	1206522631	14	1206522727	16
35	1206522365	15	1206522457	17	1206522547	16	1206522633	16	1206522727	16
36	1206522366	16	1206522457	17	1206522547	16	1206522632	15	1206522728	17
37	1206522365	15	1206522458	18	1206522549	18	1206522632	15	1206522728	17
38	1206522366	16	1206522458	18	1206522549	18	1206522633	16	1206522729	18
39	1206522366	16	1206522460	20	1206522549	18	1206522634	17	1206522729	18
40	1206522367	17	1206522459	19	1206522550	19	1206522634	17	1206522729	18
41	1206522367	17	1206522459	19	1206522551	20	1206522634	17	1206522730	19
42	1206522367	17	1206522460	20	1206522551	20	1206522634	17	1206522730	19
43	1206522368	18	1206522461	21	1206522551	20	1206522635	18	1206522730	19
44	1206522368	18	1206522461	21	1206522551	20	1206522636	19	1206522730	19
45	1206522369	19	1206522461	21	1206522552	21	1206522636	19	1206522731	20
46	1206522369	19	1206522463	23	1206522553	22	1206522636	19	1206522731	20
47	1206522369	19	1206522461	21	1206522553	22	1206522636	19	1206522732	21
48	1206522371	21	1206522462	22	1206522554	23	1206522637	20	1206522732	21
49	1206522370	20	1206522463	23	1206522555	24	1206522637	20	1206522732	21
50	1206522371	21	1206522462	22	1206522554	23	1206522637	20	1206522733	22
51	1206522371	21	1206522464	24	1206522554	23	1206522638	21	1206522733	22
52	1206522371	21	1206522464	24	120652255	24	1206522638	21	1206522733	22
53	1206522372	22	1206522465	25	1206522555	24	1206522638	21	1206522733	22
54	1206522372	22	1206522466	26	1206522556	25	1206522639	22	1206522735	24
55	1206522373	23	1206522465	25	1206522556	25	1206522639	22	1206522734	23
						25		22		
56	1206522374	24	1206522466	26	1206522556	-	1206522639		1206522734	23
57		23	1206522466	26		26		23	1206522735	24
58	1206522375	25	1206522466	26	1206522557	26	1206522641	24	1206522735	24
59	1206522374	24	1206522467	27	1206522557	26	1206522641	24	1206522735	24
60	1206522374	24	1206522468	28	1206522558	27	1206522641	24	1206522736	25
61	1206522376	26	1206522467	27	1206522558	27	1206522642	25	1206522737	26
62	1206522375	25	1206522468	28	1206522559	28	1206522642	25	1206522738	27
63	1206522375	25	1206522468	28	1206522559	28	1206522642	25	1206522737	26
64	1206522377	27	1206522468	28	1206522560	29	1206522642	25	1206522737	26
65	1206522376	26	1206522469	29	1206522560	29	1206522643	26	1206522738	27
66	1206522377	27	1206522470	30	1206522560	29	1206522643	26	1206522738	27
67	1206522378	28	1206522469	29	1206522560	29	1206522644	27	1206522739	28
68	1206522377	27	1206522469	29	1206522561	30	1206522644	27	1206522740	29
69	1206522377	27	1206522470	30	1206522561	30	1206522645	28	1206522739	28
70	1206522378	28	1206522471	31	1206522563	32	1206522644	27	1206522739	28
71	1206522378	28	1206522472	32	1206522562	31	1206522645	28	1206522740	29
72	1206522379	29	1206522471	31	1206522563	32	1206522645	28	1206522740	29
73	1206522380	30	1206522472	32	1206522562	31	1206522645	28	1206522741	30
74	1206522381	31	1206522472	32	1206522563	32	1206522646	29	1206522741	30
75	1206522380	30	1206522473	33	1206522564	33	1206522646	29	1206522741	30
76	1206522381	31	1206522474	34	1206522565	34	1206522647	30	1206522742	31
77	1206522381	31	1206522473	33	1206522564	33	1206522647	30	1206522742	31
78	1206522382	32	1206522474	34	1206522565	34	1206522647	30	1206522743	32
79	1206522382	32	1206522474	34	1206522565	34	1206522648	31	1206522743	32
80	1206522383	33	1206522474	34	1206522566	35	1206522648	31	1206522743	32
81	1206522382	32	1206522475	35	1206522566	35	1206522648	31	1206522743	32
82	1206522383	33	1206522476	36	1206522566	35	1206522649	32	1206522744	33
83	1206522383	33	1206522475	35	1206522566	35	1206522649	32	1206522745	34
84	1206522384	34	1206522476	36	1206522567	36	1206522649	32	1206522745	34
		<u> </u>		1				<u> </u>		1

85	1206522386	36	1206522477	37	1206522567	36	1206522650	33	1206522745	34
86	1206522384	34	1206522477	37	1206522568	37	1206522650	33	1206522744	33
87	1206522385	35	1206522477	37	1206522568	37	1206522651	34	1206522746	35
88	1206522386	36	1206522477	37	1206522569	38	1206522650	33	1206522745	34
89	1206522385	35	1206522478	38	1206522569	38	1206522651	34	1206522747	36
90	1206522387	37	1206522478	38	1206522569	38	1206522652	35	1206522747	36
91	1206522386	36	1206522479	39	1206522570	39	1206522652	35	1206522747	36
92	1206522387	37	1206522479	39	1206522570	39	1206522652	35	1206522747	36
93	1206522388	38	1206522479	39	1206522570	39	1206522653	36	1206522749	38
94	1206522387	37	1206522480	40	1206522571	40	1206522653	36	1206522748	37
95	1206522388	38	1206522480	40	1206522571	40	1206522654	37	1206522748	37
96	1206522388	38	1206522480	40	1206522571	40	1206522654	37	1206522749	38
97	1206522390	40	1206522481	41	1206522572	41	1206522654	37	1206522749	38
98	1206522390	40	1206522482	42	1206522572	41	1206522654	37	1206522750	39
99	1206522389	39	1206522482	42	1206522572	41	1206522655	38	1206522750	39
100	1206522389	39	1206522482	42	1206522573	42	1206522655	38	1206522750	39

Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A
1	1206522824	0	1206522911	0	1206523000	0	1206523079	0	1206523159	0
2	1206522823	-1	1206522912	1	1206523001	1	1206523078	-1	1206523159	0
3	1206522824	0	1206522912	1	1206523003	3	1206523079	0	1206523161	2
4	1206522825	1	1206522913	2	1206523002	2	1206523079	0	1206523160	1
5	1206522825	1	1206522914	3	1206523003	3	1206523079	0	1206523161	2
6	1206522826	2	1206522915	4	1206523003	3	1206523082	3	1206523161	2
7	1206522827	3	1206522916	5	1206523004	4	1206523081	2	1206523162	3
8	1206522827	3	1206522914	3	1206523006	6	1206523081	2	1206523162	3
9	1206522828	4	1206522916	5	1206523004	4	1206523083	4	1206523163	4
10	1206522828	4	1206522916	5	1206523007	7	1206523083	4	1206523163	4
11	1206522829	5	1206522917	6	1206523006	6	1206523083	4	1206523163	4
12	1206522830	6	1206522917	6	1206523005	5	1206523083	4	1206523164	5
13	1206522830	6	1206522917	6	1206523007	7	1206523084	5	1206523166	7
14	1206522830	6	1206522918	7	1206523008	8	1206523084	5	1206523165	6
15	1206522831	7	1206522918	7	1206523006	6	1206523084	5	1206523166	7
16	1206522831	7	1206522919	8	1206523008	8	1206523085	6	1206523165	6
17	1206522831	7	1206522920	9	1206523008	8	1206523085	6	1206523165	6
18	1206522832	8	1206522921	10	1206523010	10	1206523086	7	1206523168	9
19	1206522831	7	1206522920	9	1206523008	8	1206523086	7	1206523167	8
20	1206522832	8	1206522921	10	1206523010	10	1206523086	7	1206523167	8
21	1206522832	8	1206522921	10	1206523009	9	1206523086	7	1206523168	9
22	1206522833	9	1206522922	11	1206523009	9	1206523087	8	1206523169	10
23	1206522833	9	1206522922	11	1206523010	10	1206523090	11	1206523168	9
24	1206522837	13	1206522922	11	1206523011	11	1206523088	9	1206523170	11
25	1206522835	11	1206522922	11	1206523011	11	1206523088	9	1206523169	10
26	1206522835	11	1206522923	12	1206523011	11	1206523088	9	1206523170	11
27	1206522834	10	1206522924	13	1206523011	11	1206523090	11	1206523170	11
28	1206522835	11	1206522923	12	1206523012	12	1206523089	10	1206523170	11
29	1206522835	11	1206522924	13	1206523012	12	1206523093	14	1206523172	13
30	1206522835	11	1206522924	13	1206523013	13	1206523089	10	1206523172	13
31	1206522835	11	1206522924	13	1206523016	16	1206523090	11	1206523171	12
32	1206522838	14	1206522926	15	1206523013	13	1206523091	12	1206523172	13
33	1206522838	14	1206522925	14	1206523012	12	1206523092	13	1206523172	13
34	1206522836	12	1206522925	14	1206523014	14	1206523092	13	1206523173	14
35	1206522837	13	1206522926	15	1206523014	14	1206523091	12	1206523173	14
36	1206522838	14	1206522927	16	1206523013	13	1206523092	13	1206523172	13
37	1206522839	15	1206522926	15	1206523015	15	1206523094	15	1206523173	14
38	1206522838	14	1206522927	16	1206523015	15	1206523093	14	1206523174	15
39	1206522838	14	1206522927	16	1206523014	14	1206523095	16	1206523173	14
40	1206522840	16	1206522928	17	1206523014	14	1206523094	15	1206523175	16

L	12065225 ::	1.5	12065225	1.5	1206522555	1,6	120652255	١,,,	1206522:-:	1.5
41	1206522841	17	1206522928	17	1206523016	16	1206523094	15	1206523174	15
42	1206522840	16	1206522929	18	1206523017	17	1206523095	16	1206523174	15
43	1206522843	19	1206522929	18	1206523016	16	1206523095	16	1206523176	17
44	1206522840	16	1206522930	19	1206523016	16	1206523096	17	1206523176	17
45	1206522840	16	1206522930	19	1206523017	17	1206523096	17	1206523176	17
46	1206522841	17	1206522930	19	1206523018	18	1206523096	17	1206523178	19
47	1206522843	19	1206522930	19	1206523016	16	1206523097	18	1206523176	17
48	1206522842	18	1206522930	19	1206523018	18	1206523099	20	1206523176	17
49	1206522844	20	1206522932	21	1206523018	18	1206523097	18	1206523176	17
50	1206522842	18	1206522931	20	1206523018	18	1206523098	19	1206523177	18
51	1206522842	18	1206522931	20	1206523019	19	1206523097	18	1206523178	19
52	1206522843	19	1206522933	22	1206523020	20	1206523097	18	1206523178	19
53	1206522843	19	1206522932	21	1206523019	19	1206523099	20	1206523178	19
54	1206522843	19	1206522931	20	1206523020	20	1206523099	20	1206523178	19
55	1206522846	22	1206522934	23	1206523019	19	1206523099	20	1206523178	19
56	1206522846	22	1206522933	22	1206523021	21	1206523102	23	1206523178	19
	1206522845	21	1206522934	23	1206523021	21	1206523102	21	1206523179	20
57										
58	1206522845	21	1206522933	22	1206523021	21	1206523105	26	1206523180	21
59	1206522846	22	1206522934	23	1206523022	22	1206523100	21	1206523181	22
60	1206522845	21	1206522934	23	1206523022	22	1206523101	22	1206523180	21
61	1206522846	22	1206522935	24	1206523021	21	1206523101	22	1206523180	21
62	1206522846	22	1206522934	23	1206523022	22	1206523102	23	1206523181	22
63	1206522848	24	1206522934	23	1206523022	22	1206523101	22	1206523181	22
64	1206522847	23	1206522936	25	1206523023	23	1206523104	25	1206523181	22
65	1206522850	26	1206522936	25	1206523023	23	1206523102	23	1206523182	23
66	1206522849	25	1206522936	25	1206523024	24	1206523102	23	1206523183	24
67	1206522848	24	1206522936	25	1206523023	23	1206523104	25	1206523183	24
68	1206522849	25	1206522936	25	1206523024	24	1206523104	25	1206523182	23
69	1206522850	26	1206522938	27	1206523023	23	1206523104	25	1206523183	24
70	1206522849	25	1206522938	27	1206523024	24	1206523104	25	1206523183	24
71	1206522850	26	1206522940	29	1206523024	24	1206523106	27	1206523184	25
72	1206522850	26	1206522939	28	1206523025	25	1206523106	27	1206523185	26
73	1206522851	27	1206522938	27	1206523026	26	1206523106	27	1206523184	25
74	1206522851	27	1206522938	27	1206523027	27	1206523106	27	1206523185	26
75		27	1206522938	27	1206523026	26	1206523107	28	1206523185	
76	1206522851 1206522851	27	1206522939	28	1206523025	25	1206523107	28	1206523186	26
77	1206522852	28	1206522940	29	1206523025	25	1206523107	28	1206523186	27
78	1206522854	30	1206522943	32	1206523027	27	1206523107	28	1206523185	26
79	1206522854	30	1206522940	29	1206523026	26	1206523108	29	1206523187	28
80	1206522853	29	1206522942	31	1206523027	27	1206523108	29	1206523187	28
81	1206522853	29	1206522941	30	1206523028	28	1206523108	29	1206523186	27
82	1206522853	29	1206522941	30	1206523027	27	1206523109	30	1206523187	28
83	1206522854	30	1206522943	32	1206523028	28	1206523110	31	1206523187	28
84	1206522854	30	1206522941	30	1206523029	29	1206523110	31	1206523187	28
85	1206522853	29	1206522941	30	1206523027	27	1206523108	29	1206523188	29
86	1206522856	32	1206522942	31	1206523029	29	1206523110	31	1206523188	29
87	1206522854	30	1206522942	31	1206523028	28	1206523111	32	1206523190	31
88	1206522855	31	1206522943	32	1206523030	30	1206523111	32	1206523188	29
89	1206522856	32	1206522944	33	1206523029	29	1206523111	32	1206523190	31
90	1206522856	32	1206522944	33	1206523031	31	1206523110	31	1206523189	30
91	1206522856	32	1206522945	34	1206523030	30	1206523115	36	1206523189	30
92	1206522856	32	1206522945	34	1206523030	30	1206523112	33	1206523189	30
93	1206522858	34	1206522946	35	1206523030	30	1206523111	32	1206523190	31
94	1206522856	32	1206522945	34	1206523032	32	1206523113	34	1206523191	32
95	1206522858	34	1206522945	34	1206523032	32	1206523114	35	1206523192	33
96	1206522858	34	1206522946	35	1206523031	31	1206523113	34	1206523191	32
97	1206522859	35	1206522946	35	1206523032	32	1206523115	36	1206523191	32
98	1206522858	34	1206522946	35	1206523033	33	1206523114	35	1206523192	33
99	1206522858	34	1206522947	36	1206523033	33	1206523114	35	1206523191	32

100 1206522860 36 1206522946 35 1206523033 33 1206523114 35 1206523192 33	100	1206522860	36	1206522946	35	1206523033	33	1206523114	35	1206523192	33
---	-----	------------	----	------------	----	------------	----	------------	----	------------	----

Count	Run 1	Run 1 A	Run 2	Run 2 A	Run 3	Run 3 A	Run 4	Run 4 A	Run 5	Run 5 A
1	1206523248	0	1206523333	0	1206524014	0	1206524170	0	1206524277	0
2	1206523249	1	1206523333	0	1206524012	-2	1206524171	1	1206524274	-3
3	1206523250	2	1206523333	0	1206524015	1	1206524175	5	1206524279	2
4	1206523252	4	1206523334	1	1206524014	0	1206524177	7	1206524275	-2
5	1206523251	3	1206523335	2	1206524014	0	1206524177	7	1206524275	-2
6	1206523252	4	1206523336	3	1206524017	3	1206524178	8	1206524277	0
7	1206523254	6	1206523338	5	1206524017	3	1206524181	11	1206524278	1
8	1206523253	5	1206523336	3	1206524019	5	1206524181	11	1206524278	1
9	1206523254	6	1206523338	5	1206524018	4	1206524181	11	1206524279	2
10	1206523256	8	1206523338	5	1206524019	5	1206524185	15	1206524279	2
11	1206523255	7	1206523340	7	1206524019	5	1206524180	10	1206524280	3
12	1206523256	8	1206523340	7	1206524021	7	1206524180	10	1206524280	3
13	1206523258	10	1206523342	9	1206524025	11	1206524189	19	1206524282	5
14	1206523256	8	1206523340	7	1206524020	6	1206524194	24	1206524282	5
		10		9		11		11		
16	1206523258	10	1206523342	7	1206524025 1206524021	7	1206524181 1206524182	12	1206524283	8
17	1206523258	10	1206523342	9	1206524022	8	1206524187	17	1206524283	6
18	1206523260	12	1206523342	9	1206524021	7	1206524183	13	1206524283	6
19	1206523261	13	1206523345	12	1206524022	8	1206524184	14	1206524284	7
20	1206523260	12	1206523343	10	1206524024	10	1206524189	19	1206524287	10
21	1206523260	12	1206523343	10	1206524023	9	1206524186	16	1206524309	32
22	1206523260	12	1206523345	12	1206524025	11	1206524184	14	1206524287	10
23	1206523262	14	1206523344	11	1206524025	11	1206524184	14	1206524290	13
24	1206523262	14	1206523345	12	1206524024	10	1206524186	16	1206524287	10
25	1206523264	16	1206523345	12	1206524025	11	1206524186	16	1206524287	10
26	1206523262	14	1206523345	12	1206524034	20	1206524186	16	1206524287	10
27	1206523266	18	1206523349	16	1206524024	10	1206524195	25	1206524288	11
28	1206523263	15	1206523346	13	1206524025	11	1206524189	19	1206524290	13
29	1206523262	14	1206523349	16	1206524026	12	1206524188	18	1206524287	10
30	1206523264	16	1206523348	15	1206524027	13	1206524187	17	1206524293	16
31	1206523264	16	1206523348	15	1206524026	12	1206524189	19	1206524288	11
32	1206523267	19	1206523347	14	1206524027	13	1206524190	20	1206524290	13
33	1206523265	17	1206523350	17	1206524028	14	1206524188	18	1206524305	28
34	1206523264	16	1206523348	15	1206524030	16	1206524189	19	1206524293	16
35	1206523269	21	1206523348	15	1206524029	15	1206524191	21	1206524290	13
36	1206523264	16	1206523352	19	1206524030	16	1206524192	22	1206524290	13
37	1206523266	18	1206523350	17	1206524032	18	1206524192	22	1206524292	15
38	1206523267	19	1206523354	21	1206524029	15	1206524191	21	1206524290	13
39	1206523265	17	1206523350	17	1206524029	15	1206524192	22	1206524293	16
40	1206523266	18	1206523352	19	1206524029	15	1206524196	26	1206524295	18
41	1206523266	18	1206523351	18	1206524029	15	1206524194	24	1206524296	19
42	1206523267	19	1206523352	19	1206524033	19	1206524192	22	1206524292	15
43	1206523267	19	1206523357	24	1206524030	16	1206524193	23	1206524293	16
44	1206523268	20	1206523359	26	1206524031	17	1206524194	24	1206524293	16
45	1206523268	20	1206523356	23	1206524031	17	1206524199	29	1206524295	18
46	1206523269	21	1206523354	21	1206524033	19	1206524193	23	1206524295	18
47	1206523271	23	1206523355	22	1206524033	19	1206524194	24	1206524298	21
48	1206523271	23	1206523357	24	1206524032	18	1206524194	24	1206524295	18
49	1206523271	24	1206523354	21	1206524032	19	1206524194	24	1206524295	20
50	1206523272	24	1206523356	23	1206524035	21	1206524195	25	1206524298	21
51	1206523271	23	1206523358	25	1206524034	20	1206524196	26	1206524299	22
52	1206523269	21	1206523356	23	1206524034	20	1206524197	27	1206524298	21
53	1206523272	24	1206523356	23	1206524033	19	1206524198	28	1206524298	21
54	1206523270	22	1206523358	25	1206524033	19	1206524197	27	1206524297	20

56	1206523271	23	1206523357	24	1206524038	24	1206524200	30	1206524298	21
57	1206523271	23	1206523358	25	1206524034	20	1206524202	32	1206524301	24
58	1206523277	29	1206523360	27	1206524035	21	1206524199	29	1206524300	23
59	1206523272	24	1206523359	26	1206524036	22	1206524203	33	1206524299	22
60	1206523274	26	1206523358	25	1206524034	20	1206524206	36	1206524302	25
61	1206523272	24	1206523359	26	1206524043	29	1206524201	31	1206524299	22
62	1206523274	26	1206523360	27	1206524039	25	1206524199	29	1206524300	23
63	1206523281	33	1206523360	27	1206524037	23	1206524200	30	1206524300	23
64	1206523275	27	1206523360	27	1206524039	25	1206524208	38	1206524300	23
65	1206523276	28	1206523360	27	1206524038	24	1206524200	30	1206524301	24
66	1206523275	27	1206523370	37	1206524039	25	1206524199	29	1206524301	24
67	1206523276	28	1206523360	27	1206524038	24	1206524200	30	1206524301	24
68	1206523275	27	1206523361	28	1206524038	24	1206524201	31	1206524302	25
69	1206523275	27	1206523361	28	1206524039	25	1206524201	31	1206524302	25
70	1206523274	26	1206523362	29	1206524040	26	1206524203	33	1206524306	29
71	1206523274	26	1206523362	29	1206524039	25	1206524203	33	1206524308	31
72	1206523275	27	1206523362	29	1206524041	27	1206524202	32	1206524309	32
73	1206523280	32	1206523362	29	1206524038	24	1206524205	35	1206524304	27
74	1206523277	29	1206523363	30	1206524039	25	1206524204	34	1206524304	27
75	1206523276	28	1206523363	30	1206524039	25	1206524203	33	1206524303	26
76	1206523282	34	1206523362	29	1206524040	26	1206524208	38	1206524306	29
77	1206523278	30	1206523367	34	1206524040	26	1206524206	36	1206524305	28
78	1206523284	36	1206523363	30	1206524044	30	1206524204	34	1206524305	28
79	1206523278	30	1206523363	30	1206524043	29	1206524207	37	1206524304	27
80	1206523279	31	1206523371	38	1206524041	27	1206524208	38	1206524308	31
81	1206523279	31	1206523366	33	1206524043	29	1206524210	40	1206524308	31
82	1206523284	36	1206523366	33	1206524047	33	1206524205	35	1206524306	29
83	1206523280	32	1206523367	34	1206524043	29	1206524205	35	1206524309	32
84	1206523281	33	1206523366	33	1206524043	29	1206524206	36	1206524308	31
85	1206523279	31	1206523368	35	1206524043	29	1206524210	40	1206524310	33
86	1206523278	30	1206523364	31	1206524044	30	1206524206	36	1206524309	32
87	1206523280	32	1206523366	33	1206524048	34	1206524208	38	1206524311	34
88	1206523279	31	1206523370	37	1206524048	34	1206524208	38	1206524311	34
89	1206523282	34	1206523367	34	1206524044	30	1206524212	42	1206524310	33
90	1206523280	32	1206523366	33	1206524043	29	1206524207	37	1206524311	34
91	1206523281	33	1206523366	33	1206524044	30	1206524208	38	1206524311	34
92	1206523284	36	1206523367	34	1206524044	30	1206524209	39	1206524311	34
93	1206523285	37	1206523367	34	1206524048	34	1206524208	38	1206524312	35
94	1206523285	37	1206523369	36	1206524047	33	1206524209	39	1206524315	38
95	1206523282	34	1206523372	39	1206524048	34	1206524211	41	1206524311	34
96	1206523284	36	1206523368	35	1206524044	30	1206524210	40	1206524313	36
97	1206523288	40	1206523370	37	1206524047	33	1206524211	41	1206524312	35
98	1206523284	36	1206523370	37	1206524048	34	1206524211	41	1206524312	35
99	1206523286	38	1206523371	38	1206524046	32	1206524210	40	1206524314	37
100	1206523283	35	1206523371	38	1206524046	32	1206524212	42	1206524314	37

One Second

Condor

1		1		1		1	1	
2	0.33	1		1		1	1	
3	0.43	2	0.18	1		1	1	
4	0.57	2	0.18	1		1	1	
5	0.63	3	0.27	2	0.1	1	1	
6	0.6	3	0.33	2	0.12	1	1	
7	0.7	4	0.44	2	0.13	1	1	

8	0.67	4	0.5	2	0.17	1		1	
9	0.75	5	0.63	3	0.3	2	0.07	1	
10	0.77	5	0.63	3	0.33	2	0.08	1	
11	0.73	6	0.55	3	0.38	2	0.09	1	
12	0.8	6	0.6	3	0.38	2	0.1	1	
13	0.76	7	0.7	4	0.44	2	0.12	1	
14	0.82	7	0.58	4	0.44	2	0.11	1	
15	0.79	8	0.67	4	0.4	2	0.13	1	
16	0.84	8	0.73	4	0.4	2	0.13	1	
17	0.85	9	0.69	5	0.56	3	0.17	2	0.04
18	0.82	9	0.69	5	0.5	3	0.17	2	0.05
19	0.86	10	0.77	5	0.56	3	0.17	2	0.05
20	0.83	10	0.71	5	0.56	3	0.19	2	0.06
21	0.88	11	0.73	6	0.67	3	0.19	2	0.06
22	0.88	11	0.79	6	0.6	3	0.18	2	0.06
23	0.85	12	0.75	6	0.6	3	0.17	2	0.06
24	0.89	12	0.75	6	0.6	3	0.18	2	0.06
25	0.86	13	0.76	7	0.58	4	0.25	2	0.06
26	0.9	13	0.81	7	0.64	4	0.24	2	0.06
27	0.9	14	0.82	7	0.7	4	0.24	2	0.06
28	0.88	14	0.78	7	0.7	4	0.22	2	0.06
29	0.91	15	0.79	8	0.73	4	0.22	2	0.06
30	0.88	15	0.79	8	0.67	4	0.24	2	0.06
31	0.91	16	0.84	8	0.67	4	0.24	2	0.06
32	0.89	16	0.84	8	0.67	4	0.24	2	0.06
33	0.92	17	0.81	9	0.69	5	0.28	3	0.09
34	0.92	17	0.85	9	0.75	5	0.29	3	0.09
35	0.9	18	0.86	9	0.64	5	0.29	3	0.09
36	0.92	18	0.80	9	0.69	5	0.29	3	0.09
37	0.92	19	0.83	10	0.67	5	0.29	3	0.09
38	0.9	19	0.86	10	0.71	5	0.29	3	0.09
39	0.93	20	0.83	10	0.71	5	0.29	3	0.09
	0.93		0.83	10		5		3	0.09
40	0.91	20		11	0.67	6	0.29	3	0.09
41			0.88		0.73		0.35		
42	0.91	21	0.88	11	0.73	6	0.35	3	0.09
43		22	0.85	11	0.79	6	0.38	3	
44	0.94	22	0.88	11	0.73	6	0.33	3	0.09
45	0.92	23	0.85	12	0.75	6	0.35	3	0.09
46	0.94	23	0.82	12	0.75	6	0.33	3	0.09
47	0.92	24	0.89	12	0.75	6	0.35	3	0.09
48	0.94	24	0.86	12	0.8	6	0.35	3	0.09
49	0.94	25	0.86	13	0.81	7	0.41	4	0.12
50	0.93	25	0.89	13	0.72	7	0.41	4	0.12
51	0.94	26	0.87	13	0.72	7	0.39	4	0.12
52	0.95	26	0.87	13	0.72	7	0.41	4	0.12
53	0.93	27	0.9	14	0.78	7	0.41	4	0.12
54	0.95	27	0.87	14	0.82	7	0.44	4	0.12
55	0.93	28	0.88	14	0.78	7	0.44	4	0.12
56	0.95	28	0.88	14	0.82	7	0.41	4	0.12
57	0.95	29	0.91	15	0.83	8	0.47	4	0.12
58	0.94	29	0.88	15	0.79	8	0.47	4	0.12
59	0.95	30	0.88	15	0.79	8	0.5	4	0.13
60	0.95	30	0.91	15	0.75	8	0.5	4	0.13
61	0.94	31	0.91	16	0.8	8	0.47	4	0.12
62	0.94	31	0.89	16	0.8	8	0.47	4	0.12
63	0.95	32	0.89	16	0.84	8	0.44	4	0.12
64	0.96	32	0.91	16	0.8	8	0.47	4	0.12
65	0.94	33	0.92	17	0.81	9	0.53	5	0.15

67	0.94	34	0.89	17	0.81	9	0.5	5	0.15
68	0.96	34	0.89	17	0.85	9	0.5	5	0.15
69	0.95	35	0.92	18	0.82	9	0.53	5	0.15
70	0.96	35	0.9	18	0.82	9	0.53	5	0.15
71	0.96	36	0.92	18	0.82	9	0.53	5	0.15
72	0.95	36	0.92	18	0.82	9	0.56	5	0.15
73	0.96	37	0.9	19	0.86	10	0.59	5	0.15
74	0.96	37	0.9	19	0.83	10	0.59	5	0.15
75	0.95	38	0.9	19	0.83	10	0.59	5	0.15
76	0.96	38	0.93	19	0.83	10	0.59	5	0.15
77	0.95	39	0.93	20	0.83	10	0.63	5	0.15
78	0.96	39	0.91	20	0.83	10	0.59	5	0.15
79	0.96	40	0.93	20	0.83	10	0.59	5	0.15
80	0.95	40	0.93	20	0.83	10	0.59	5	0.15
81	0.96	41	0.91	21	0.84	11	0.65	6	0.19
82	0.95	41	0.91	21	0.88	11	0.65	6	0.18
83	0.97	42	0.93	21	0.84	11	0.65	6	0.18
85	0.97	42	0.93	21 22	0.84	11	0.65	6	0.19
86	0.97	43	0.91	22	0.85	11	0.65	6	0.18
88	0.97	44	0.94	22	0.85	11	0.63	6	0.18
89	0.96	45	0.92	23	0.85	12	0.71	6	0.18
90	0.96	45	0.94	23	0.85	12	0.71	6	0.18
91	0.97	46	0.94	23	0.85	12	0.71	6	0.18
92	0.97	46	0.92	23	0.88	12	0.71	6	0.18
93	0.96	47	0.92	24	0.89	12	0.71	6	0.18
94	0.97	47	0.94	24	0.86	12	0.75	6	0.18
95	0.97	48	0.94	24	0.83	12	0.75	6	0.18
96	0.96	48	0.92	24	0.83	12	0.71	6	0.18
97	0.97	49	0.92	25	0.86	13	0.81	7	0.21
98	0.96	49	0.94	25	0.89	13	0.81	7	0.21
99	0.97	50	0.94	25	0.89	13	0.76	7	0.21
100	0.97	50	0.93	25	0.86	13	0.76	7	0.21
		51	0.93	26	0.87	13	0.68	7	0.21
		51	0.94	26	0.87	13	0.72	7	0.21
		52	0.95	26	0.87	13	0.76	7	0.21
		52	0.93	26	0.87	13	0.76	7	0.21
		53	0.95	27	0.9	14	0.78	7	0.21
		53	0.95	27	0.87	14	0.78	7	0.21
		54	0.93	27	0.87	14	0.78	7	0.21
		54	0.95	27	0.9	14	0.7	7	0.21
		55	0.95	28	0.88	14	0.82	7	0.21
		55	0.93	28	0.88	14	0.7	7	0.22
		56	0.93	28	0.88	14	0.7	7	0.21
		56	0.93	28	0.9	14	0.7	7	0.21
		57	0.95	29	0.88	15	0.75	8	0.24
		57	0.95	29	0.91	15	0.75	8	0.25
		58	0.94	29	0.91	15	0.79	8	0.24
		58	0.95	29	0.85	15	0.75	8	0.24
		59	0.95	30	0.88	15	0.68	8	0.24
		59	0.94	30	0.91	15	0.71	8	0.24
		60	0.94	30	0.88	15	0.68	8	0.24
		60	0.95	30	0.88	15	0.75	8	0.24
		61	0.95	31	0.91	16	0.76	8	0.24
		61	0.94	31	0.86	16	0.8	8	0.24
		62	0.94	31	0.89	16	0.8	8	0.24

	63	0.94	32	0.91	16	0.8	8	0.24
	64	0.94	32	0.89	16	0.8	8	0.24
	64	0.96	32	0.89	16	0.73	8	0.24
	65	0.96	33	0.89	17	0.85	9	0.26
	65	0.94	33	0.89	17	0.81	9	0.26
	66	0.94	33	0.92	17	0.81	9	0.27
	66	0.96	33	0.89	17	0.85	9	0.27
	67	0.96	34	0.89	17	0.77	9	0.27
	67	0.96	34	0.89	17	0.81	9	0.26
	68	0.94	34	0.89	17	0.81	9	0.28
	68	0.94	34	0.92	17	0.81	9	0.27
	69	0.96	35	0.92	18	0.78	9	0.27
	69	0.96	35	0.92	18	0.82	9	0.27
	70	0.95	35	0.9	18	0.82	9	0.27
	70	0.95	35	0.9	18	0.82	9	0.27
	71	0.96	36	0.9	18	0.82	9	0.27
	71	0.96	36	0.9	18	0.78	9	0.28
	72	0.95	36	0.92	18	0.78	9	0.27
	72	0.95	36	0.9	18	0.86	9	0.27
	73	0.96	37	0.9	19	0.83	10	0.27
	73	0.96	37	0.93	19	0.83	10	0.31
	74	0.96		0.93				
			37		19	0.79	10	0.29
	74	0.96	37	0.93	19	0.86	10	0.3
	75	0.95	38	0.93	19	0.83	10	0.3
	75	0.95	38	0.88	19	0.83	10	0.31
	76	0.96	38	0.9	19	0.79	10	0.3
	76	0.95	38	0.9	19	0.79	10	0.31
	77	0.95	39	0.91	20	0.83	10	0.3
	77	0.96	39	0.93	20	0.83	10	0.3
	78	0.96	39	0.93	20	0.83	10	0.3
	78	0.95	39	0.91	20	0.83	10	0.3
	79	0.95	40	0.91	20	0.87	10	0.3
	79	0.96	40	0.89	20	0.83	10	0.3
	80	0.96	40	0.91	20	0.87	10	0.31
	80	0.95	40	0.93	20	0.83	10	0.3
	81	0.95	41	0.91	21	0.88	11	0.33
	81	0.96	41	0.91	21	0.81	11	0.33
	82	0.96	41	0.91	21	0.84	11	0.34
	82	0.95	41	0.93	21	0.88	11	0.33
	83	0.95	42	0.93	21	0.84	11	0.33
	83	0.97	42	0.93	21	0.81	11	0.33
	84	0.97	42	0.91	21	0.84	11	0.33
	84	0.97	42	0.89	21	0.81	11	0.32
	85	0.97	43	0.91	22	0.85	11	0.34
	85	0.96	43	0.93	22	0.88	11	0.33
	86	0.96	43	0.91	22	0.85	11	0.33
	86	0.97	43	0.91	22	0.85	11	0.33
	87	0.97	44	0.92	22	0.85	11	0.33
	87	0.96	44	0.94	22	0.85	11	0.33
	88	0.96	44	0.94	22	0.85	11	0.32
	88	0.97	44	0.92	22	0.88	11	0.34
	89	0.97	45	0.94	23	0.88	12	0.36
	89	0.97	45	0.94	23	0.88	12	0.36
	90	0.97	45	0.92	23	0.85	12	0.36
	90	0.97	45	0.94	23	0.82	12	0.36
	91	0.96	46	0.92	23	0.85	12	0.36
	91	0.97	46	0.92	23	0.85	12	0.36
	92	0.97	46	0.94	23	0.85	12	0.36
	92	0.97	46	0.92	23	0.82	12	0.36

93	1	l 02	0.06	47	0.02	24	0.86	1,2	0.25
95									
96									
		96	0.96	48	0.94	24	0.86	12	0.35
98		97	0.97	49	0.94	25	0.86	13	0.39
		97	0.97	49	0.92		0.86	13	
99		98	0.96	49	0.94	25	0.86	13	0.39
99		98	0.97	49	0.92	25	0.86	13	0.39
100		99	0.97	50	0.93	25	0.89	13	0.38
100		99	0.96	50	0.94	25	0.89	13	0.41
51 0.93 26 0.87 13 0.39 1 51 0.03 26 0.87 13 0.41 1 51 0.04 26 0.84 13 0.29 1 51 0.04 26 0.87 13 0.38 1 52 0.95 26 0.87 13 0.38 1 52 0.93 26 0.9 13 0.38 1 52 0.93 26 0.9 13 0.38 1 53 0.93 26 0.84 13 0.39 1 53 0.93 27 0.87 14 0.42 1 53 0.93 27 0.87 14 0.41 1 53 0.93 27 0.87 14 0.41 1 54 0.93 27 0.87 14 0.41 1 54 0.95 27		100	0.96	50	0.93	25	0.86	13	0.39
51 0.93 26 0.87 13 0.41 1 51 0.94 26 0.84 13 0.99 2 0.93 26 0.87 13 0.38 3 22 0.93 26 0.97 13 0.38 4 1 22 0.93 26 0.99 13 0.88 5 2 0.93 26 0.99 13 0.38 0.38 1 2 0.93 26 0.94 13 0.39 0.39 1 3 0.93 27 0.87 14 0.42 0.41 0.41 0.42 0.41 0.44 0.41 0.42 0.41 0.41 0.44 0.41 0.44 0.41 0.44 0.41 0.44 0.41 0.44 0.41 0.44 0.44 0.41 0.44 0.44 0.44 0.44 0.44 0.44 0.44 0.44 0.44 0.44 0.44<		100	0.97	50	0.93	25	0.89	13	0.39
S1				51	0.93	26	0.87	13	0.39
51 0.94 26 0.87 13 0.38 1 22 0.95 26 0.87 13 0.38 1 52 0.93 26 0.99 13 0.38 1 52 0.93 26 0.99 13 0.98 1 52 0.93 26 0.94 13 0.99 1 53 0.93 27 0.87 14 0.42 1 53 0.93 27 0.87 14 0.41 1 53 0.93 27 0.87 14 0.41 1 54 0.95 27 0.87 14 0.41 1 54 0.95 27 0.87 14 0.41 1 54 0.95 27 0.87 14 0.41 1 54 0.95 27 0.87 14 0.41 1 54 0.95 27				51	0.93	26	0.87	13	0.41
				51	0.94	26	0.84	13	0.39
S2				51	0.94	26	0.87	13	0.38
52 0.93 26 0.9 13 0.38 52 0.93 26 0.84 13 0.99 53 0.93 27 0.87 14 0.41 53 0.93 27 0.87 14 0.41 53 0.93 27 0.87 14 0.42 54 0.93 27 0.87 14 0.41 54 0.93 27 0.87 14 0.41 54 0.93 27 0.87 14 0.41 54 0.95 27 0.87 14 0.41 54 0.95 27 0.87 14 0.41 54 0.95 27 0.87 14 0.41 55 0.93 28 0.9 14 0.42 55 0.93 28 0.9 14 0.42 55 0.93 28 0.88 14 0.42				52	0.95	26	0.87	13	0.38
52 0.93 26 0.84 13 0.39 53 0.93 27 0.87 14 0.42 53 0.95 27 0.87 14 0.42 53 0.93 27 0.87 14 0.42 53 0.93 27 0.87 14 0.41 54 0.95 27 0.87 14 0.41 54 0.95 27 0.87 14 0.41 54 0.95 27 0.87 14 0.41 54 0.95 27 0.87 14 0.41 54 0.95 27 0.87 14 0.41 55 0.03 28 0.9 14 0.42 55 0.03 28 0.9 14 0.42 55 0.03 28 0.9 14 0.42 55 0.03 28 0.9 14 0.42 55 0.03 28 0.88 14 0.42 56 0.95 </td <td></td> <td></td> <td></td> <td>52</td> <td>0.93</td> <td>26</td> <td>0.9</td> <td>13</td> <td>0.38</td>				52	0.93	26	0.9	13	0.38
53 0.93 27 0.87 14 0.42 1 53 0.95 27 0.87 14 0.41 1 53 0.93 27 0.87 14 0.41 1 53 0.93 27 0.87 14 0.41 1 54 0.93 27 0.87 14 0.41 1 54 0.95 27 0.87 14 0.41 1 54 0.95 27 0.87 14 0.41 1 54 0.95 27 0.87 14 0.41 1 55 0.93 28 0.9 14 0.42 1 55 0.93 28 0.9 14 0.42 1 55 0.93 28 0.9 14 0.42 1 55 0.93 28 0.9 14 0.42 1 56 0.95 28 <				52	0.93	26	0.9	13	0.38
53 0.95 27 0.87 14 0.41 53 0.93 27 0.87 14 0.42 53 0.93 27 0.87 14 0.41 54 0.93 27 0.87 14 0.41 54 0.95 27 0.87 14 0.41 54 0.95 27 0.87 14 0.41 54 0.95 27 0.87 14 0.41 54 0.95 27 0.87 14 0.41 54 0.95 27 0.87 14 0.41 55 0.93 28 0.9 14 0.42 55 0.93 28 0.9 14 0.42 55 0.93 28 0.9 14 0.42 56 0.93 28 0.9 14 0.42 56 0.95 28 0.88 14 0.42 <				52	0.93	26	0.84	13	0.39
53 0.93 27 0.87 14 0.42 53 0.93 27 0.87 14 0.41 54 0.93 27 0.87 14 0.44 54 0.95 27 0.87 14 0.41 54 0.95 27 0.87 14 0.41 54 0.95 27 0.87 14 0.41 54 0.95 27 0.87 14 0.41 64 0.95 27 0.87 14 0.41 65 0.93 28 0.9 14 0.42 65 0.93 28 0.9 14 0.42 65 0.93 28 0.88 14 0.41 66 0.95 28 0.88 14 0.42 66 0.95 28 0.9 14 0.42 66 0.95 28 0.88 14 0.42 66 0.95 28 0.88 14 0.42 77 0.93<				53	0.93	27	0.87	14	0.42
53 0.93 27 0.87 14 0.41 54 0.93 27 0.9 14 0.44 54 0.95 27 0.87 14 0.41 54 0.95 27 0.87 14 0.41 54 0.95 27 0.87 14 0.41 54 0.95 27 0.87 14 0.41 54 0.95 27 0.87 14 0.41 55 0.93 28 0.9 14 0.42 55 0.93 28 0.9 14 0.42 55 0.93 28 0.9 14 0.42 56 0.93 28 0.9 14 0.42 56 0.95 28 0.9 14 0.42 56 0.95 28 0.88 14 0.44 56 0.95 28 0.9 14 0.42 56 0.93 28 0.9 14 0.42 57 0.93				53	0.95	27	0.87	14	0.41
54 0.93 27 0.9 14 0.44 54 0.95 27 0.87 14 0.41 54 0.95 27 0.87 14 0.41 54 0.95 27 0.87 14 0.41 64 0.95 27 0.87 14 0.41 64 0.95 27 0.87 14 0.41 64 0.95 27 0.87 14 0.41 64 0.95 28 0.9 14 0.42 65 0.93 28 0.9 14 0.42 66 0.95 28 0.9 14 0.42 66 0.95 28 0.9 14 0.42 66 0.95 28 0.9 14 0.42 66 0.95 28 0.88 14 0.44 66 0.95 28 0.88 14 0.42 66 0.95 28 0.88 15 0.45 77 0.93 <td></td> <td></td> <td></td> <td>53</td> <td>0.93</td> <td>27</td> <td>0.87</td> <td>14</td> <td>0.42</td>				53	0.93	27	0.87	14	0.42
54 0.95 27 0.87 14 0.41 54 0.95 27 0.87 14 0.41 54 0.95 27 0.87 14 0.41 55 0.93 28 0.9 14 0.42 55 0.93 28 0.9 14 0.42 55 0.93 28 0.9 14 0.42 55 0.93 28 0.9 14 0.42 55 0.93 28 0.9 14 0.42 55 0.93 28 0.9 14 0.42 56 0.95 28 0.9 14 0.42 56 0.95 28 0.88 14 0.42 56 0.95 28 0.88 14 0.42 56 0.95 28 0.88 14 0.42 57 0.93 29 0.88 15 0.45 57 0.93 29 0.88 15 0.47 57 0.95				53	0.93	27	0.87	14	0.41
54 0.95 27 0.87 14 0.41 54 0.95 27 0.87 14 0.41 55 0.93 28 0.9 14 0.42 55 0.93 28 0.88 14 0.41 55 0.93 28 0.88 14 0.42 55 0.93 28 0.9 14 0.42 55 0.93 28 0.9 14 0.42 55 0.93 28 0.9 14 0.42 56 0.95 28 0.9 14 0.42 56 0.92 28 0.88 14 0.42 56 0.95 28 0.88 14 0.42 57 0.93 28 0.9 14 0.42 57 0.93 29 0.88 15 0.45 57 0.93 29 0.88 15 0.45 57 0.93 29 0.88 15 0.47 58 0.94 <td></td> <td></td> <td></td> <td>54</td> <td>0.93</td> <td>27</td> <td>0.9</td> <td>14</td> <td>0.44</td>				54	0.93	27	0.9	14	0.44
54 0.95 27 0.87 14 0.41 55 0.93 28 0.9 14 0.42 55 0.93 28 0.88 14 0.41 55 0.93 28 0.9 14 0.42 55 0.93 28 0.9 14 0.42 56 0.95 28 0.9 14 0.42 56 0.92 28 0.88 14 0.42 56 0.92 28 0.88 14 0.42 56 0.95 28 0.9 14 0.42 56 0.95 28 0.88 14 0.42 56 0.95 28 0.88 14 0.42 57 0.93 29 0.88 15 0.45 57 0.93 29 0.88 15 0.45 57 0.95 29 0.88 15 0.47 58 0.94 29 0.88 15 0.47 58 0.95 <td></td> <td></td> <td></td> <td>54</td> <td>0.95</td> <td>27</td> <td>0.87</td> <td>14</td> <td>0.41</td>				54	0.95	27	0.87	14	0.41
55 0.93 28 0.9 14 0.42 55 0.93 28 0.88 14 0.41 55 0.93 28 0.9 14 0.42 55 0.93 28 0.9 14 0.42 56 0.95 28 0.9 14 0.42 56 0.95 28 0.88 14 0.42 56 0.92 28 0.88 14 0.42 56 0.95 28 0.88 14 0.42 56 0.95 28 0.88 14 0.44 60 0.95 28 0.88 14 0.44 60 0.95 28 0.88 14 0.44 60 0.93 29 0.88 15 0.45 77 0.93 29 0.88 15 0.45 8 0.94 29 0.88 15 0.47 9 0.88 15 0.47 0.45 0.45 10 1.5 <td></td> <td></td> <td></td> <td>54</td> <td>0.95</td> <td>27</td> <td>0.87</td> <td>14</td> <td>0.41</td>				54	0.95	27	0.87	14	0.41
55 0.93 28 0.88 14 0.41 55 0.93 28 0.9 14 0.42 55 0.93 28 0.88 14 0.44 60 0.95 28 0.9 14 0.42 60 0.95 28 0.9 14 0.42 60 0.95 28 0.88 14 0.42 60 0.95 28 0.88 14 0.42 60 0.95 28 0.88 14 0.42 60 0.95 28 0.88 14 0.42 60 0.95 28 0.88 14 0.44 60 0.93 29 0.88 15 0.45 60 0.93 29 0.88 15 0.45 60 0.94 29 0.88 15 0.47 60 0.94 29 0.88 15 0.47 60 0.94 29 0.85 15 0.47 60 0.95<				54	0.95	27	0.87	14	0.41
55 0.93 28 0.9 14 0.42 55 0.93 28 0.88 14 0.44 56 0.95 28 0.9 14 0.42 56 0.92 28 0.88 14 0.42 56 0.95 28 0.88 14 0.42 56 0.93 28 0.9 14 0.42 57 0.93 29 0.88 15 0.45 57 0.93 29 0.88 15 0.45 57 0.93 29 0.88 15 0.47 57 0.93 29 0.88 15 0.47 57 0.93 29 0.88 15 0.47 57 0.93 29 0.88 15 0.47 57 0.95 29 0.88 15 0.47 58 0.94 29 0.88 15 0.45 58 0.94 29 0.85 15 0.47 59 0.95<				55	0.93	28	0.9	14	0.42
55 0.93 28 0.88 14 0.44 56 0.95 28 0.9 14 0.42 56 0.92 28 0.88 14 0.42 56 0.95 28 0.88 14 0.44 66 0.93 28 0.9 14 0.42 57 0.93 29 0.88 15 0.45 57 0.95 29 0.88 15 0.45 57 0.93 29 0.88 15 0.47 57 0.93 29 0.88 15 0.47 57 0.93 29 0.88 15 0.47 57 0.95 29 0.88 15 0.47 58 0.94 29 0.88 15 0.47 58 0.94 29 0.88 15 0.47 58 0.94 29 0.88 15 0.47 59 0.95 30 0.88 15 0.47 59 0.94				55	0.93	28	0.88	14	0.41
55 0.93 28 0.88 14 0.44 56 0.95 28 0.9 14 0.42 56 0.92 28 0.88 14 0.42 56 0.95 28 0.88 14 0.44 66 0.93 28 0.9 14 0.42 57 0.93 29 0.88 15 0.45 57 0.95 29 0.88 15 0.45 57 0.93 29 0.88 15 0.47 57 0.93 29 0.88 15 0.47 57 0.93 29 0.88 15 0.47 57 0.95 29 0.88 15 0.47 58 0.94 29 0.88 15 0.47 58 0.94 29 0.88 15 0.47 58 0.94 29 0.88 15 0.47 59 0.95 30 0.88 15 0.47 59 0.94				55	0.93	28	0.9	14	0.42
56 0.92 28 0.88 14 0.42 56 0.95 28 0.88 14 0.44 56 0.93 28 0.9 14 0.42 57 0.93 29 0.88 15 0.45 57 0.95 29 0.88 15 0.45 57 0.93 29 0.85 15 0.47 57 0.95 29 0.88 15 0.47 58 0.94 29 0.88 15 0.47 58 0.94 29 0.88 15 0.47 58 0.94 29 0.85 15 0.47 58 0.94 29 0.85 15 0.47 58 0.94 29 0.88 15 0.47 59 0.95 30 0.88 15 0.47 59 0.95 30 0.88 15 0.47 59 0.95 30 0.88 15 0.45 59 0.9					0.93			14	
56 0.92 28 0.88 14 0.42 56 0.95 28 0.88 14 0.44 56 0.93 28 0.9 14 0.42 57 0.93 29 0.88 15 0.45 57 0.95 29 0.88 15 0.45 57 0.93 29 0.85 15 0.47 57 0.95 29 0.88 15 0.47 58 0.94 29 0.88 15 0.47 58 0.94 29 0.88 15 0.47 58 0.94 29 0.85 15 0.47 58 0.94 29 0.85 15 0.47 58 0.94 29 0.88 15 0.47 59 0.95 30 0.88 15 0.47 59 0.95 30 0.88 15 0.47 59 0.95 30 0.88 15 0.45 59 0.9				56	0.95	28	0.9	14	0.42
56 0.93 28 0.9 14 0.42 57 0.93 29 0.88 15 0.45 57 0.95 29 0.88 15 0.45 57 0.93 29 0.85 15 0.47 57 0.95 29 0.88 15 0.47 58 0.94 29 0.88 15 0.45 58 0.94 29 0.91 15 0.44 58 0.94 29 0.85 15 0.47 58 0.95 29 0.85 15 0.47 58 0.94 29 0.85 15 0.47 58 0.95 29 0.85 15 0.47 59 0.95 30 0.88 15 0.47 59 0.95 30 0.88 15 0.47 59 0.95 30 0.88 15 0.45 59 0.95 30 0.88 15 0.45 60 0.9				56				14	0.42
56 0.93 28 0.9 14 0.42 57 0.93 29 0.88 15 0.45 57 0.95 29 0.88 15 0.45 57 0.93 29 0.85 15 0.47 57 0.95 29 0.88 15 0.47 58 0.94 29 0.88 15 0.45 58 0.94 29 0.91 15 0.44 58 0.94 29 0.85 15 0.47 58 0.95 29 0.85 15 0.47 58 0.94 29 0.85 15 0.47 58 0.95 29 0.85 15 0.47 59 0.95 30 0.88 15 0.47 59 0.95 30 0.88 15 0.47 59 0.95 30 0.88 15 0.45 59 0.95 30 0.88 15 0.45 60 0.9				56	0.95	28	0.88	14	0.44
57 0.93 29 0.88 15 0.45 57 0.95 29 0.88 15 0.45 57 0.93 29 0.85 15 0.47 57 0.95 29 0.88 15 0.47 58 0.94 29 0.88 15 0.45 58 0.94 29 0.91 15 0.44 58 0.94 29 0.85 15 0.47 58 0.94 29 0.85 15 0.47 58 0.94 29 0.85 15 0.47 58 0.94 29 0.88 15 0.47 58 0.94 29 0.88 15 0.47 59 0.95 30 0.88 15 0.45 59 0.94 30 0.88 15 0.47 59 0.95 30 0.91 15 0.45 59 0.95 30 0.88 15 0.45 60 0.									
57 0.95 29 0.88 15 0.45 57 0.93 29 0.85 15 0.47 57 0.95 29 0.88 15 0.47 58 0.94 29 0.88 15 0.45 58 0.94 29 0.91 15 0.44 58 0.95 29 0.85 15 0.47 58 0.94 29 0.88 15 0.47 59 0.95 30 0.88 15 0.47 59 0.95 30 0.88 15 0.47 59 0.95 30 0.88 15 0.45 59 0.95 30 0.88 15 0.45 59 0.95 30 0.88 15 0.45 59 0.95 30 0.88 15 0.45 60 0.94 30 0.86 15 0.47 60 0.95 30 0.88 15 0.45 60 0.									
57 0.93 29 0.85 15 0.47 57 0.95 29 0.88 15 0.47 58 0.94 29 0.88 15 0.45 58 0.94 29 0.91 15 0.44 58 0.95 29 0.85 15 0.47 58 0.94 29 0.88 15 0.47 59 0.95 30 0.88 15 0.45 59 0.94 30 0.88 15 0.47 59 0.95 30 0.88 15 0.45 59 0.95 30 0.88 15 0.45 59 0.95 30 0.88 15 0.45 60 0.94 30 0.88 15 0.45 60 0.95 30 0.88 15 0.45 60 0.92 30 0.88 15 0.45 60 0.92 30 0.88 15 0.45 60 0.									
57 0.95 29 0.88 15 0.47 58 0.94 29 0.88 15 0.45 58 0.94 29 0.91 15 0.44 58 0.95 29 0.85 15 0.47 58 0.94 29 0.88 15 0.47 59 0.95 30 0.88 15 0.45 59 0.94 30 0.88 15 0.47 59 0.95 30 0.91 15 0.45 59 0.95 30 0.88 15 0.45 59 0.95 30 0.88 15 0.45 60 0.94 30 0.88 15 0.45 60 0.95 30 0.88 15 0.45 60 0.92 30 0.88 15 0.45 60 0.92 30 0.88 15 0.45 60 0.94 30 0.88 15 0.45 60 0.									
58 0.94 29 0.88 15 0.45 58 0.94 29 0.91 15 0.44 58 0.95 29 0.85 15 0.47 58 0.94 29 0.88 15 0.47 59 0.95 30 0.88 15 0.45 59 0.94 30 0.88 15 0.47 59 0.95 30 0.91 15 0.45 59 0.95 30 0.88 15 0.45 60 0.94 30 0.88 15 0.45 60 0.94 30 0.86 15 0.47 60 0.95 30 0.88 15 0.45 60 0.92 30 0.88 15 0.45 60 0.92 30 0.88 15 0.45 60 0.94 30 0.88 15 0.45 60 0.94 30 0.88 15 0.45 60 0.									
58 0.94 29 0.91 15 0.44 58 0.95 29 0.85 15 0.47 58 0.94 29 0.88 15 0.47 59 0.95 30 0.88 15 0.45 59 0.94 30 0.88 15 0.47 59 0.95 30 0.91 15 0.45 59 0.95 30 0.88 15 0.45 60 0.94 30 0.88 15 0.45 60 0.94 30 0.88 15 0.47 60 0.95 30 0.88 15 0.45 60 0.92 30 0.88 15 0.45 60 0.94 30 0.88 15 0.45 60 0.94 30 0.88 15 0.45 60 0.94 30 0.88 15 0.45 60 0.94 30 0.88 15 0.45 60 0.									
58 0.95 29 0.85 15 0.47 58 0.94 29 0.88 15 0.47 59 0.95 30 0.88 15 0.45 59 0.94 30 0.88 15 0.47 59 0.95 30 0.91 15 0.45 59 0.95 30 0.88 15 0.45 60 0.94 30 0.86 15 0.47 60 0.95 30 0.88 15 0.45 60 0.92 30 0.88 15 0.45 60 0.92 30 0.88 15 0.45 60 0.94 30 0.88 15 0.45 60 0.94 30 0.88 15 0.45 60 0.94 30 0.88 15 0.45 60 0.94 30 0.88 15 0.45 60 0.94 30 0.89 16 0.48									
58 0.94 29 0.88 15 0.47 59 0.95 30 0.88 15 0.45 59 0.94 30 0.88 15 0.47 59 0.95 30 0.91 15 0.45 59 0.95 30 0.88 15 0.45 60 0.94 30 0.86 15 0.47 60 0.95 30 0.88 15 0.45 60 0.92 30 0.88 15 0.45 60 0.94 30 0.88 15 0.45 60 0.94 30 0.88 15 0.45 60 0.94 30 0.88 15 0.45 61 0.94 31 0.89 16 0.48 61 0.95 31 0.89 16 0.5									
59 0.95 30 0.88 15 0.45 59 0.94 30 0.88 15 0.47 59 0.95 30 0.91 15 0.45 59 0.95 30 0.88 15 0.45 60 0.94 30 0.86 15 0.47 60 0.95 30 0.88 15 0.45 60 0.92 30 0.88 15 0.45 60 0.94 30 0.88 15 0.47 61 0.94 31 0.89 16 0.48 61 0.95 31 0.89 16 0.5									
59 0.94 30 0.88 15 0.47 59 0.95 30 0.91 15 0.45 59 0.95 30 0.88 15 0.45 60 0.94 30 0.86 15 0.47 60 0.95 30 0.88 15 0.45 60 0.92 30 0.88 15 0.45 60 0.94 30 0.88 15 0.47 61 0.94 31 0.89 16 0.48 61 0.95 31 0.89 16 0.5									
59 0.95 30 0.91 15 0.45 59 0.95 30 0.88 15 0.45 60 0.94 30 0.86 15 0.47 60 0.95 30 0.88 15 0.45 60 0.92 30 0.88 15 0.45 60 0.94 30 0.88 15 0.47 61 0.94 31 0.89 16 0.48 61 0.95 31 0.89 16 0.5									
59 0.95 30 0.88 15 0.45 60 0.94 30 0.86 15 0.47 60 0.95 30 0.88 15 0.45 60 0.92 30 0.88 15 0.45 60 0.94 30 0.88 15 0.47 61 0.94 31 0.89 16 0.48 61 0.95 31 0.89 16 0.5									
60 0.94 30 0.86 15 0.47 60 0.95 30 0.88 15 0.45 60 0.92 30 0.88 15 0.45 60 0.94 30 0.88 15 0.47 61 0.94 31 0.89 16 0.48 61 0.95 31 0.89 16 0.5									
60 0.95 30 0.88 15 0.45 60 0.92 30 0.88 15 0.45 60 0.94 30 0.88 15 0.47 61 0.94 31 0.89 16 0.48 61 0.95 31 0.89 16 0.5									
60 0.92 30 0.88 15 0.45 60 0.94 30 0.88 15 0.47 61 0.94 31 0.89 16 0.48 61 0.95 31 0.89 16 0.5									
60 0.94 30 0.88 15 0.47 61 0.94 31 0.89 16 0.48 61 0.95 31 0.89 16 0.5									
61 0.94 31 0.89 16 0.48 61 0.95 31 0.89 16 0.5									
61 0.95 31 0.89 16 0.5									
61 0.95 31 0.89 16 0.41									
				61	0.95	31	0.89	16	0.41

1	1		l	1	1	1	l	l
			61	0.95	31	0.89	16	0.44
			62	0.95	31	0.89	16	0.46
			62	0.93	31	0.89	16	0.48
			62	0.94	31	0.89	16	0.5
			62	0.94	31	0.91	16	0.47
			63	0.94	32	0.89	16	0.5
			63	0.95	32	0.89	16	0.47
			63	0.94	32	0.86	16	0.41
			63	0.94	32	0.89	16	0.42
			64	0.94	32	0.89	16	0.43
			64	0.94	32	0.89	16	0.44
			64	0.96	32	0.89	16	0.43
			64	0.96	32	0.89	16	0.43
			65	0.96	33	0.89	17	0.47
			65	0.94	33	0.89	17	0.45
			65	0.94	33	0.92	17	0.55
			65	0.94	33	0.92	17	0.52
			66	0.94	33	0.92	17	0.49
			66	0.96	33	0.89	17	0.47
			66	0.96	33	0.92	17	0.46
			66	0.94	33	0.89	17	0.47
			67	0.94	34	0.92	17	0.45
			67	0.96	34	0.89	17	0.46
			67	0.96	34	0.89	17	0.46
			67	0.96	34	0.92	17	0.53
			68	0.96	34	0.92	17	0.53
			68	0.96	34	0.89	17	0.5
			68	0.94	34	0.89	17	0.53
			68	0.94	34	0.92	17	0.52
			69	0.95	35	0.9	18	0.53
			69	0.93	35	0.9	18	0.55
			69	0.93	35	0.9	18	0.53
			69	0.96	35	0.88	18	0.53
			70	0.95	35	0.9	18	0.55
			70	0.96	35	0.92	18	0.55
			70	0.95	35	0.9	18	0.51
			70	0.95	35	0.9	18	0.55
			71	0.95	36	0.9	18	0.55
			71	0.95	36	0.92	18	0.53
			71	0.96	36	0.9	18	0.55
			71	0.95	36	0.92	18	0.53
			72	0.96	36	0.9	18	0.5
			72	0.96	36	0.9	18	0.53
			72	0.95	36	0.9	18	0.51
			72	0.95	36	0.88	18	0.5
			73	0.95	37	0.93	19	0.56
			73	0.95	37	0.9	19	0.54
			73	0.96	37	0.9	19	0.54
			73	0.95	37	0.88	19	0.56
			74	0.96	37	0.93	19	0.54
			74	0.96	37	0.9	19	0.56
			74	0.95	37	0.9	19	0.56
			74	0.96	37	0.9	19	0.56
			75	0.95	38	0.9	19	0.56
						· .		
			75	0.95	38	0.9	19	0.56
1			75 75	0.95	38	0.9	19	0.56
			75	0.96	38	0.9	19	0.56
			75 75	0.96 0.95	38 38	0.9	19 19	0.56 0.58
			75	0.96	38	0.9	19	0.56

I	I	I	I	1	1	ı	I	I
			76	0.95	38	0.93	19	0.56
			76	0.96	38	0.93	19	0.59
			77	0.95	39	0.91	20	0.59
			77	0.96	39	0.93	20	0.61
			77	0.96	39	0.91	20	0.63
			77	0.96	39	0.91	20	0.59
			78	0.96	39	0.91	20	0.63
			78	0.96	39	0.91	20	0.61
			78	0.95	39	0.93	20	0.61
			78	0.95	39	0.89	20	0.59
			79	0.95	40	0.93	20	0.59
			79	0.96	40	0.91	20	0.61
			79	0.96	40	0.91	20	0.59
			79	0.95	40	0.91	20	0.59
			80	0.96	40	0.91	20	0.61
			80	0.94	40	0.91	20	0.63
			80	0.95	40	0.91	20	0.61
			80	0.96	40	0.93	20	0.61
			81	0.95	41	0.91	21	0.66
			81	0.96	41	0.93	21	0.64
			81	0.96	41	0.93	21	0.62
			81	0.95	41	0.91	21	0.64
			82	0.95	41	0.91	21	0.62
			82	0.95	41	0.91	21	0.62
			82	0.96	41	0.93	21	0.64
			82	0.96	41	0.91	21	0.66
			83	0.97	42	0.91	21	0.66
			83	0.97	42	0.91	21	0.66
			83	0.94	42	0.91	21	0.66
			83	0.95	42	0.91	21	0.64
			84	0.95	42	0.91	21	0.64
			84	0.95	42	0.91	21	0.62
			84	0.95	42	0.91	21	0.66
			84	0.97	42	0.93	21	0.64
			85	0.97	43	0.91	22	0.65
			85	0.97	43	0.91	22	0.67
				0.96		0.91		0.67
			85		43		22	
			85	0.96	43	0.93	22	0.69
			86	0.96	43	0.91	22	0.67
			86	0.96	43	0.91	22	0.65
			86	0.97	43	0.9	22	0.67
			86	0.97	43	0.91	22	0.65
			87	0.97	44	0.92	22	0.65
			87	0.97	44	0.92	22	0.65
			87	0.96	44	0.92	22	0.65
			87	0.97	44	0.92	22	0.67
			88	0.96	44	0.92	22	0.65
			88	0.96	44	0.94	22	0.63
			88	0.96	44	0.94	22	0.61
			88	0.95	44	0.92	22	0.63
			89	0.97	45	0.94	23	0.64
			89	0.97	45	0.92	23	0.64
			89	0.97	45	0.94	23	0.64
						0.94		
			89	0.97	45		23	0.64
			90	0.96	45	0.94	23	0.64
			90	0.96	45	0.92	23	0.66
			90	0.97	45	0.92	23	0.66
İ		1	90	0.96	45	0.94	23	0.64
			91	0.97	46	0.92	23	0.64

1	l	I		l		١.,	l.,,	l	
				91	0.97	46	0.94	23	0.64
				91	0.96	46	0.92	23	0.64
				91	0.97	46	0.94	23	0.64
				92	0.97	46	0.94	23	0.64
				92	0.96	46	0.92	23	0.68
				92	0.97	46	0.92	23	0.66
				92	0.97	46	0.92	23	0.64
				93	0.96	47	0.92	24	0.73
				93	0.97	47	0.92	24	0.71
				93	0.96	47	0.92	24	0.73
				93	0.97	47	0.94	24	0.69
				94	0.97	47	0.94	24	0.71
				94	0.97	47	0.94	24	0.71
				94	0.97	47	0.92	24	0.71
				94	0.95	47	0.92	24	0.71
				95	0.96	48	0.94	24	0.71
				95	0.97	48	0.94	24	0.71
				95	0.96	48	0.92	24	0.73
				95	0.97	48	0.92	24	0.71
				96	0.97	48	0.91	24	0.71
				96	0.95	48	0.92	24	0.73
				96	0.97	48	0.92	24	0.73
				96	0.97	48	0.92	24	0.75
				97	0.96	49	0.92	25	0.74
				97	0.97	49	0.92	25	0.74
				97	0.96	49	0.92	25	0.74
				97	0.96	49	0.91	25	0.74
				98	0.97	49	0.91	25	0.78
				98	0.96	49	0.91	25	0.76
				98	0.96		0.92		0.76
						49		25	
				98	0.97	49	0.94	25	0.76
				99	0.96	50	0.93	25	0.76
				99	0.97	50	0.93	25	0.78
				99	0.97	50	0.93	25	0.76
				99	0.96	50	0.91	25	0.76
				100	0.97	50	0.93	25	0.78
				100	0.96	50	0.91	25	0.71
				100	0.96	50	0.91	25	0.76
				100	0.97	50	0.93	25	0.74
						51	0.94	26	0.81
						51	0.94	26	0.81
						51	0.93	26	0.79
						51	0.94	26	0.79
						51	0.93	26	0.79
						51	0.93	26	0.79
						51	0.93	26	0.79
						51	0.93	26	0.81
						52	0.93	26	0.79
						52	0.93	26	0.79
						52	0.95	26	0.76
						52	0.93	26	0.67
						52	0.93	26	0.68
						52	0.95	26	0.72
						52	0.93	26	0.74
						52	0.91	26	0.76
						53	0.93	27	0.79
						53	0.93	27	0.79
						53	0.93	27	0.82
						53	0.95	27	0.82
	I	<u> </u>		1	<u> </u>	<u> </u>	<u> </u>	<u> </u>	1

1	Ī			1			
				53	0.93	27	0.79
				53	0.93	27	0.84
				53	0.95	27	0.82
				53	0.93	27	0.82
				54	0.95	27	0.82
				54	0.93	27	0.79
				54	0.95	27	0.69
				54	0.92	27	0.84
				54	0.93	27	0.84
				54	0.93	27	0.79
				54	0.93	27	0.77
				54	0.95	27	0.73
				55	0.93	28	0.74
				55	0.95	28	0.74
				55	0.93	28	0.74
				55	0.95	28	0.74
				55	0.93	28	0.74
				55	0.93	28	0.72
				55	0.93	28	0.74
				55	0.93	28	0.74
				56	0.93	28	0.76
				56	0.93	28	0.7
				56	0.93	28	0.88
				56	0.93	28	0.82
				56	0.93	28	0.82
				56	0.93	28	0.82
				56	0.93	28	0.82
				56	0.93	28	0.82
				57	0.95	29	0.85
				57	0.95	29	0.88
				57	0.93	29	0.88
				57	0.93	29	0.83
				57	0.95	29	0.85
				57	0.95	29	0.85
				57	0.95	29	0.88
				57	0.93	29	0.85
				58	0.95	29	0.81
				58	0.93	29	0.88
				58	0.95	29	0.85
				58	0.94	29	0.85
				58	0.94	29	0.85
				58	0.95	29	0.85
				58	0.95	29	0.83
				58	0.94	29	0.85
				59	0.94	30	0.88
				59	0.95	30	0.83
				59	0.95	30	0.83
				59	0.95	30	0.86
				59	0.94	30	0.88
				59	0.94	30	0.88
				59	0.94	30	0.83
				59	0.94	30	0.83
				60	0.95	30	0.88
				60	0.94	30	0.91
				60	0.95	30	0.86
				60	0.94	30	0.91
				60	0.94	30	0.86
L				i .	1		I .
				60	0.95	30	0.88
				60	0.95	30	0.88

1	1	1	1		ı			1
					60	0.95	30	0.86
					61	0.94	31	0.86
					61	0.95	31	0.91
					61	0.94	31	0.89
					61	0.94	31	0.91
					61	0.95	31	0.89
					61	0.92	31	0.86
					61	0.95	31	0.89
					61	0.94	31	0.89
					62	0.94	31	0.91
					62	0.94	31	0.86
					62	0.94	31	0.89
					62	0.95	31	0.84
					62	0.94	31	0.86
					62	0.94	31	0.89
					62	0.94	31	0.86
					62	0.94	31	0.91
					63	0.95	32	0.91
					63	0.94	32	0.86
					63	0.95	32	0.91
					63	0.94	32	0.89
					63	0.94	32	0.86
					63	0.94	32	0.86
					63	0.93	32	0.86
					63	0.95	32	0.89
					64	0.94	32	0.84
					64	0.96	32	0.89
					64	0.94	32	0.86
					64	0.96	32	0.91
					64	0.96	32	0.89
					64	0.96	32	0.84
					64	0.94	32	0.89
					64	0.94	32	0.84
					65	0.94	33	0.92
					65	0.96	33	0.92
					65	0.96	33	0.89
					65	0.94	33	0.87
					65	0.93	33	0.89
					65	0.93	33	0.89
					65	0.94	33	0.87
					65	0.94	33	0.87
					66	0.96	33	0.87
					66	0.94	33	0.85
					66	0.94	33	0.89
					66	0.94	33	0.87
					66	0.96	33	0.87
					66	0.96	33	0.87
					66	0.96	33	0.85
					66	0.94	33	0.85
					67	0.94	34	0.89
					67	0.96	34	0.89
					67	0.94	34	0.87
					67	0.94	34	0.92
					67	0.94	34	0.89
					67	0.94	34	0.87
					67	0.94	34	0.92
					67	0.96	34	0.89
					68	0.96	34	0.87
					68	0.94	34	0.87

1	1			ı			
				68	0.96	34	0.85
				68	0.94	34	0.85
				68	0.96	34	0.85
				68	0.94	34	0.89
				68	0.96	34	0.87
				68	0.94	34	0.87
				69	0.95	35	0.88
				69	0.95	35	0.88
				69	0.95	35	0.92
				69	0.96	35	0.88
				69	0.95	35	0.88
				69	0.95	35	0.9
				69	0.93	35	0.88
				69	0.95	35	0.85
				70	0.95	35	0.92
				70	0.96	35	0.9
				70	0.96	35	0.92
				70	0.95	35	0.88
				70	0.96	35	0.9
				70	0.96	35	0.85
				70	0.96	35	0.9
				70	0.95	35	0.9
				71	0.96	36	0.88
				71	0.95	36	0.92
				71	0.95	36	0.88
				71	0.96	36	0.88
				71	0.95	36	0.92
				71	0.95	36	0.92
				71	0.95	36	0.9
				71	0.96	36	0.9
				72 72	0.95	36 36	0.88
							0.86
				72	0.96	36	
				72	0.95	36	0.88
				72	0.96	36	0.88
				72	0.95	36	0.92
				72	0.95	36	0.88
				72	0.95	36	0.9
				73	0.95	37	0.93
				73	0.96	37	0.88
				73	0.95	37	0.93
				73	0.96	37	0.9
				73	0.95	37	0.88
				73	0.96	37	0.88
				73	0.96	37	0.88
				73	0.95	37	0.9
				74	0.96	37	0.93
				74	0.95	37	0.88
				74	0.96	37	0.88
				74	0.95	37	0.9
				74	0.96	37	0.9
				 74	0.95	37	0.86
				74	0.94	37	0.88
				74	0.96	37	0.86
				75	0.95	38	0.88
				75	0.95	38	0.93
				 75	0.95	38	0.88
				75	0.96	38	0.93
				75	0.95	38	0.88
		•		•		-	

75	
75	
76	
76	
76	
76	
76 0.96 38 0.88 76 0.95 38 0.9 76 0.95 38 0.88 76 0.96 38 0.9 77 0.96 39 0.91 77 0.95 39 0.91 77 0.95 39 0.91 77 0.95 39 0.89 77 0.95 39 0.91 77 0.96 39 0.89 8 0.77 0.95 39 0.91 77 0.96 39 0.89 9 0.91 77 0.96 39 0.89 10 0.77 0.95 39 0.91 0.89 10 0.77 0.95 39 0.91 0.89 10 0.77 0.94 39 0.89 0.91 78 0.96 39 0.91 0.89 0.91 78 0.96 39 0.91 0.91 78 0.96 39 0.91 0.91 0.89 0.91	
76 0.95 38 0.98 76 0.95 38 0.88 76 0.96 38 0.9 77 0.96 39 0.91 77 0.95 39 0.89 77 0.95 39 0.89 77 0.96 39 0.89 77 0.96 39 0.89 77 0.96 39 0.91 77 0.96 39 0.91 77 0.96 39 0.91 77 0.95 39 0.91 77 0.95 39 0.91 77 0.95 39 0.91 77 0.95 39 0.91 78 0.95 39 0.91 78 0.96 39 0.91 78 0.95 39 0.91 78 0.95 39 0.91 78 0.95 39 0.91 78 0.95 39 0.91 78 0.9	
76 0.95 38 0.88 76 0.96 38 0.9 77 0.96 39 0.91 77 0.95 39 0.87 77 0.95 39 0.91 77 0.96 39 0.89 77 0.95 39 0.91 77 0.95 39 0.89 77 0.95 39 0.89 77 0.95 39 0.89 77 0.95 39 0.89 80 39 0.89 0.89 90 39 0.89 0.89 10 77 0.95 39 0.91 77 0.95 39 0.91 78 0.96 39 0.91 78 0.96 39 0.91 78 0.96 39 0.91 78 0.96 39 0.91 78 0.96 39 0.91 79 0.95 40 0.89 79<	
76 0.95 38 0.88 76 0.96 38 0.9 77 0.96 39 0.91 77 0.95 39 0.87 77 0.95 39 0.91 77 0.96 39 0.89 77 0.95 39 0.91 77 0.95 39 0.91 77 0.95 39 0.91 77 0.95 39 0.91 77 0.95 39 0.89 90 0.91 0.95 39 0.91 77 0.95 39 0.91 77 0.95 39 0.91 78 0.96 39 0.91 78 0.96 39 0.91 78 0.95 39 0.91 78 0.96 39 0.91 78 0.95 39 0.91 78 0.96 39 0.91 78 0.95 39 0.91 7	
76 0.96 38 0.9 77 0.96 39 0.91 77 0.95 39 0.87 77 0.95 39 0.91 77 0.96 39 0.89 77 0.96 39 0.89 77 0.96 39 0.89 77 0.96 39 0.89 77 0.96 39 0.89 77 0.95 39 0.91 77 0.95 39 0.91 77 0.95 39 0.91 77 0.95 39 0.91 78 0.95 39 0.91 78 0.96 39 0.91 78 0.96 39 0.91 78 0.95 39 0.91 78 0.96 39 0.91 78 0.96 39 0.91 78 0.95 39 0.91 78 0.96 39 0.91 79 0.9	
77 0.96 39 0.91 77 0.95 39 0.87 77 0.95 39 0.91 77 0.96 39 0.89 77 0.95 39 0.91 77 0.96 39 0.89 77 0.96 39 0.89 77 0.94 39 0.91 77 0.94 39 0.89 8 0.95 39 0.91 78 0.96 39 0.91 78 0.96 39 0.91 78 0.95 39 0.91 78 0.95 39 0.91 78 0.95 39 0.91 78 0.95 39 0.91 79 0.95 40 0.89 79 0.95 40 0.89 79 0.95 40 0.91 79 0.96 40 0.91 79 0.96 40 0.91 79 0.9	
77 0.95 39 0.87 77 0.95 39 0.91 77 0.96 39 0.89 77 0.95 39 0.91 77 0.96 39 0.89 77 0.95 39 0.91 77 0.94 39 0.89 8 0.95 39 0.91 78 0.96 39 0.91 78 0.96 39 0.91 78 0.96 39 0.91 78 0.95 39 0.91 78 0.95 39 0.91 78 0.95 39 0.91 78 0.95 39 0.91 78 0.95 39 0.91 78 0.95 39 0.91 78 0.95 39 0.91 79 0.95 40 0.89 79 0.95 40 0.89 79 0.95 40 0.89 79 0.9	
77 0.95 39 0.91 77 0.96 39 0.89 77 0.95 39 0.91 77 0.96 39 0.89 77 0.95 39 0.91 77 0.94 39 0.89 78 0.95 39 0.91 78 0.96 39 0.91 78 0.96 39 0.91 78 0.96 39 0.91 78 0.96 39 0.91 78 0.95 39 0.91 78 0.95 39 0.91 78 0.95 39 0.91 78 0.95 39 0.91 78 0.95 39 0.91 78 0.95 39 0.91 78 0.95 39 0.91 78 0.95 39 0.93 79 0.95 40 0.89 79 0.95 40 0.89 79 0.	
77 0.96 39 0.89 77 0.95 39 0.91 77 0.96 39 0.89 77 0.94 39 0.91 78 0.95 39 0.91 78 0.96 39 0.91 78 0.96 39 0.91 78 0.96 39 0.91 78 0.95 39 0.91 78 0.95 39 0.91 78 0.95 39 0.91 78 0.95 39 0.91 78 0.95 39 0.91 78 0.95 39 0.91 79 0.95 40 0.89 79 0.95 40 0.89 79 0.95 40 0.89 79 0.95 40 0.91 79 0.96 40 0.91 79 0.96 40 0.91 79 0.96 40 0.91 79 0.	
77 0.95 39 0.91 77 0.96 39 0.89 77 0.95 39 0.91 77 0.94 39 0.89 78 0.95 39 0.91 78 0.96 39 0.91 78 0.96 39 0.91 78 0.96 39 0.91 78 0.96 39 0.91 78 0.95 39 0.91 78 0.95 39 0.91 78 0.95 39 0.91 78 0.96 39 0.91 78 0.95 39 0.91 78 0.96 39 0.91 79 0.95 40 0.89 79 0.95 40 0.89 79 0.96 40 0.91 79 0.96 40 0.91 79 0.96 40 0.91 79 0.96 40 0.91 79 0.	
77 0.96 39 0.89 77 0.95 39 0.91 77 0.94 39 0.89 78 0.95 39 0.91 78 0.96 39 0.91 78 0.96 39 0.91 78 0.96 39 0.91 78 0.95 39 0.91 78 0.95 39 0.91 78 0.95 39 0.91 78 0.95 39 0.91 78 0.95 39 0.91 78 0.95 39 0.91 78 0.95 39 0.91 78 0.95 39 0.91 79 0.95 40 0.89 79 0.95 40 0.89 79 0.95 40 0.89 79 0.96 40 0.89 79 0.96 40 0.89 79 0.96 40 0.91 79 0.	
77 0.95 39 0.91 77 0.94 39 0.89 78 0.95 39 0.91 78 0.96 39 0.91 78 0.96 39 0.91 78 0.96 39 0.91 78 0.95 39 0.91 78 0.95 39 0.91 78 0.95 39 0.91 78 0.95 39 0.91 78 0.95 39 0.91 79 0.95 40 0.89 79 0.95 40 0.89 79 0.95 40 0.89 79 0.96 40 0.91 79 0.96 40 0.91 79 0.96 40 0.91 79 0.96 40 0.91 79 0.96 40 0.91 79 0.96 40 0.91	
77 0.94 39 0.89 78 0.95 39 0.91 78 0.96 39 0.91 78 0.96 39 0.91 78 0.96 39 0.91 78 0.96 39 0.91 78 0.96 39 0.91 78 0.95 39 0.91 78 0.95 39 0.89 78 0.95 39 0.91 78 0.95 39 0.91 78 0.95 39 0.91 78 0.95 39 0.91 78 0.95 39 0.91 78 0.95 40 0.89 79 0.95 40 0.89 79 0.95 40 0.89 79 0.95 40 0.89 79 0.96 40 0.89	
78 0.95 39 0.91 78 0.96 39 0.91 78 0.96 39 0.91 78 0.96 39 0.91 78 0.95 39 0.91 78 0.95 39 0.91 78 0.95 39 0.91 78 0.95 39 0.91 78 0.95 39 0.91 79 0.95 40 0.89 79 0.95 40 0.89 79 0.95 40 0.89 79 0.95 40 0.91 79 0.96 40 0.89 79 0.96 40 0.89 79 0.96 40 0.89 79 0.96 40 0.89 79 0.96 40 0.89 79 0.96 40 0.91 79 0.96 40 0.91 79 0.95 40 0.91 79	
78 0.96 39 0.91 78 0.96 39 0.91 78 0.96 39 0.91 78 0.95 39 0.91 78 0.95 39 0.91 78 0.95 39 0.91 78 0.95 39 0.91 79 0.95 40 0.89 79 0.95 40 0.89 79 0.95 40 0.89 79 0.95 40 0.89 79 0.95 40 0.89 79 0.95 40 0.89 79 0.96 40 0.89 79 0.96 40 0.89 79 0.96 40 0.89 79 0.96 40 0.89 79 0.96 40 0.91 79 0.95 40 0.91 79 0.95 40 0.91	
78 0.96 39 0.91 78 0.96 39 0.91 78 0.95 39 0.89 78 0.95 39 0.91 78 0.96 39 0.91 78 0.95 39 0.93 79 0.95 40 0.89 79 0.95 40 0.89 79 0.95 40 0.89 79 0.95 40 0.91 79 0.96 40 0.89 79 0.96 40 0.89 79 0.96 40 0.89 79 0.96 40 0.91 79 0.95 40 0.91 79 0.95 40 0.91 79 0.95 40 0.91	
78 0.96 39 0.91 78 0.95 39 0.89 78 0.95 39 0.91 78 0.96 39 0.91 78 0.95 39 0.93 79 0.95 40 0.89 79 0.95 40 0.89 79 0.95 40 0.89 79 0.95 40 0.91 79 0.96 40 0.89 79 0.96 40 0.89 79 0.96 40 0.91 79 0.95 40 0.91 79 0.95 40 0.91	
78 0.95 39 0.89 78 0.95 39 0.91 78 0.96 39 0.91 78 0.95 39 0.93 79 0.95 40 0.89 79 0.95 40 0.89 79 0.95 40 0.89 79 0.95 40 0.91 79 0.96 40 0.89 79 0.96 40 0.91 79 0.95 40 0.91 79 0.95 40 0.91 79 0.95 40 0.91	
78 0.95 39 0.91 78 0.96 39 0.91 78 0.95 39 0.93 79 0.95 40 0.89 79 0.95 40 0.89 79 0.95 40 0.89 79 0.95 40 0.91 79 0.96 40 0.89 79 0.96 40 0.91 79 0.95 40 0.91 79 0.95 40 0.91 79 0.95 40 0.91	
78 0.96 39 0.91 78 0.95 39 0.93 79 0.95 40 0.89 79 0.95 40 0.89 79 0.95 40 0.89 79 0.95 40 0.91 79 0.96 40 0.89 79 0.96 40 0.89 79 0.96 40 0.91 79 0.95 40 0.91 79 0.95 40 0.91 79 0.95 40 0.91	
78 0.95 39 0.93 79 0.95 40 0.89 79 0.95 40 0.89 79 0.95 40 0.89 79 0.95 40 0.91 79 0.96 40 0.89 79 0.96 40 0.89 79 0.96 40 0.91 79 0.95 40 0.91 79 0.95 40 0.91	
79 0.95 40 0.89 79 0.95 40 0.89 79 0.95 40 0.89 79 0.95 40 0.91 79 0.96 40 0.89 79 0.96 40 0.91 79 0.96 40 0.91 79 0.95 40 0.91	
79 0.95 40 0.89 79 0.95 40 0.89 79 0.95 40 0.91 79 0.96 40 0.89 79 0.96 40 0.91 79 0.95 40 0.91 79 0.95 40 0.91	
79 0.95 40 0.89 79 0.95 40 0.91 79 0.96 40 0.89 79 0.96 40 0.91 79 0.95 40 0.91 79 0.95 40 0.91	
79 0.95 40 0.91 79 0.96 40 0.89 79 0.96 40 0.91 79 0.95 40 0.91 79 0.95 40 0.91	
79 0.96 40 0.89 79 0.96 40 0.91 79 0.95 40 0.91	
79 0.96 40 0.89 79 0.96 40 0.91 79 0.95 40 0.91	
79 0.96 40 0.91 79 0.95 40 0.91	
79 0.95 40 0.91	
80 0.96 40 0.89	
80 0.96 40 0.91	
80 0.96 40 0.89	
80 0.96 40 0.93	
80 0.95 40 0.89	
80 0.95 40 0.87	
80 0.95 40 0.91	
80 0.96 40 0.87	
81 0.95 41 0.89	
81 0.95 41 0.91	
81 0.95 41 0.89	
81 0.95 41 0.91	
81 0.96 41 0.91	
81 0.95 41 0.91	
81 0.95 41 0.89	
81 0.95 41 0.89	
82 0.96 41 0.91	
82 0.95 41 0.89	
82 0.96 41 0.91	
82 0.96 41 0.89	
82 0.96 41 0.91	
82 0.96 41 0.91	
82 0.94 41 0.89	

Signature Sign		1	1		ı			
					83	0.95	42	0.91
					83	0.97	42	0.91
SS					83	0.95	42	0.93
					83	0.95	42	0.89
S					83	0.95	42	0.91
					83	0.95	42	0.89
					83	0.95	42	0.91
S4					83	0.97	42	0.91
Selection Sele					84	0.97	42	0.91
S								
SS								
S								
S								
S								
SS								
SS								
SS					85	0.96	43	
86 0.97 43 0.91 86 0.97 43 0.91 86 0.96 43 0.9 86 0.97 43 0.91 86 0.97 43 0.91 86 0.97 43 0.91 86 0.96 43 0.91 86 0.96 43 0.91 87 0.96 44 0.94 87 0.96 44 0.94 87 0.96 44 0.92 87 0.96 44 0.92 87 0.97 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.97 44 0.92 88 0.9					85	0.97	43	0.9
86 0.97 43 0.91 86 0.96 43 0.9 86 0.97 43 0.91 86 0.97 43 0.91 86 0.97 43 0.91 86 0.96 43 0.91 86 0.96 43 0.91 87 0.96 44 0.94 87 0.96 44 0.94 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 88 0.97 44 0.92 88 0.97 44 0.92 88 0.97 44 0.92 88 0.9					85	0.96	43	0.91
Section Sect					86	0.97	43	0.91
86 0.97 43 0.91 86 0.97 43 0.91 86 0.97 43 0.91 86 0.97 43 0.91 86 0.96 43 0.91 86 0.96 43 0.91 87 0.96 44 0.94 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 88 0.97 44 0.92 88 0.97 44 0.94 88 0.97 44 0.94 88 0.97 44 0.92 88 0.					86	0.97	43	0.91
86 0.97 43 0.91 86 0.97 43 0.91 86 0.96 43 0.91 86 0.96 43 0.91 87 0.96 44 0.94 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 88 0.97 44 0.92 88 0.97 44 0.92 88 0.97 44 0.94 88 0.97 44 0.94 88 0.97 44 0.94 88 0.97 44 0.92 88 0.					86	0.96	43	0.9
86 0.97 43 0.91 86 0.96 43 0.91 86 0.96 43 0.91 87 0.96 44 0.94 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.97 44 0.92 88 0.97 44 0.94 88 0.97 44 0.92 88 0.97 44 0.92 88 0.97 44 0.92 88 0.97 44 0.92 88 0.97 44 0.92 88 0.					86	0.97	43	0.9
86 0.96 43 0.91 86 0.96 43 0.9 87 0.96 44 0.94 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 88 0.97 44 0.92 88 0.97 44 0.94 88 0.97 44 0.94 88 0.97 44 0.94 88 0.97 44 0.94 88 0.97 44 0.94 88 0.97 44 0.94 88 0.97 44 0.92 88 0.97 44 0.92 88 0.9					86	0.97	43	0.91
86 0.96 43 0.94 87 0.96 44 0.94 87 0.97 44 0.92 87 0.97 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 88 0.97 44 0.92 88 0.97 44 0.92 88 0.97 44 0.92 88 0.97 44 0.92 88 0.97 44 0.92 88 0.97 44 0.92 88 0.97 44 0.92 88 0.97 44 0.92 88 0.97 44 0.92 88 0.97 44 0.92 88 0.97 44 0.92 89 0.					86	0.97	43	0.91
86 0.96 43 0.94 87 0.96 44 0.94 87 0.97 44 0.92 87 0.97 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 88 0.97 44 0.92 88 0.97 44 0.92 88 0.97 44 0.92 88 0.97 44 0.92 88 0.97 44 0.92 88 0.97 44 0.92 88 0.97 44 0.92 88 0.97 44 0.92 88 0.97 44 0.92 88 0.97 44 0.92 88 0.97 44 0.92 89 0.					86	0.96	43	0.91
87 0.96 44 0.94 87 0.96 44 0.9 87 0.97 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.97 44 0.92 87 0.97 44 0.92 88 0.97 44 0.94 88 0.97 44 0.94 88 0.97 44 0.92 88 0.97 44 0.94 88 0.97 44 0.94 88 0.97 44 0.92 88 0.97 44 0.92 88 0.97 44 0.92 88 0.97 44 0.92 88 0.97 44 0.92 88 0.97 44 0.92 88 0.97 44 0.92 89 0.9								
87 0.96 44 0.92 87 0.97 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 88 0.97 44 0.92 88 0.97 44 0.94 88 0.97 44 0.92 88 0.97 44 0.92 88 0.97 44 0.94 88 0.97 44 0.92 88 0.97 44 0.92 88 0.97 44 0.92 88 0.97 44 0.92 88 0.97 44 0.92 88 0.97 44 0.92 88 0.97 44 0.92 88 0.97 44 0.92 88 0.97 44 0.92 88 0.								
87 0.97 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.9 87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.92 88 0.97 44 0.94 88 0.97 44 0.94 88 0.97 44 0.92 88 0.97 44 0.99 88 0.97 44 0.99 88 0.97 44 0.99 88 0.97 44 0.99 88 0.97 44 0.99 88 0.97 44 0.99 88 0.97 44 0.99 88 0.97 44 0.99 88 0.97 44 0.99 88 0.96 45 0.92 89 0.96 45 0.92 89 0.9								
87 0.96 44 0.92 87 0.96 44 0.92 87 0.96 44 0.9 87 0.96 44 0.92 87 0.96 44 0.92 88 0.97 44 0.94 88 0.97 44 0.92 88 0.97 44 0.92 88 0.97 44 0.92 88 0.97 44 0.94 88 0.97 44 0.92 88 0.97 44 0.94 88 0.97 44 0.92 88 0.97 44 0.92 88 0.97 44 0.92 88 0.97 44 0.92 88 0.97 44 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.9								
87 0.96 44 0.92 87 0.96 44 0.9 87 0.97 44 0.92 87 0.96 44 0.92 88 0.97 44 0.94 88 0.97 44 0.9 88 0.97 44 0.92 88 0.97 44 0.94 88 0.97 44 0.94 88 0.97 44 0.94 88 0.97 44 0.94 88 0.97 44 0.92 88 0.97 44 0.92 88 0.97 44 0.92 88 0.97 44 0.92 88 0.97 44 0.92 88 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96								
87 0.96 44 0.9 87 0.97 44 0.92 87 0.96 44 0.92 88 0.97 44 0.94 88 0.97 44 0.9 88 0.96 44 0.92 88 0.97 44 0.94 88 0.97 44 0.9 88 0.97 44 0.9 88 0.97 44 0.9 88 0.97 44 0.9 88 0.97 44 0.9 88 0.97 44 0.9 88 0.97 44 0.9 88 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96								
87 0.97 44 0.92 87 0.96 44 0.92 88 0.97 44 0.94 88 0.97 44 0.92 88 0.96 44 0.92 88 0.97 44 0.94 88 0.97 44 0.9 88 0.97 44 0.9 88 0.97 44 0.9 88 0.97 44 0.9 88 0.97 44 0.9 88 0.96 44 0.9 88 0.97 44 0.9 88 0.97 44 0.9 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96								
87 0.96 44 0.92 88 0.97 44 0.94 88 0.97 44 0.92 88 0.96 44 0.92 88 0.97 44 0.94 88 0.97 44 0.94 88 0.97 44 0.92 88 0.97 44 0.92 88 0.97 44 0.92 88 0.96 44 0.9 88 0.97 44 0.9 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96								
88 0.97 44 0.94 88 0.97 44 0.9 88 0.96 44 0.92 88 0.97 44 0.94 88 0.97 44 0.92 88 0.97 44 0.92 88 0.96 44 0.9 88 0.97 44 0.9 88 0.97 44 0.9 88 0.97 44 0.9 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>								
88 0.97 44 0.9 88 0.96 44 0.92 88 0.97 44 0.94 88 0.97 44 0.9 88 0.97 44 0.92 88 0.96 44 0.9 88 0.97 44 0.9 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.97 45 0.92 89 0.96 45 0.92 90 0.97 </td <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>								
88 0.96 44 0.92 88 0.97 44 0.94 88 0.97 44 0.9 88 0.97 44 0.92 88 0.96 44 0.9 88 0.97 44 0.9 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.97 45 0.92 90 0.97 45 0.92								
88 0.97 44 0.94 88 0.97 44 0.9 88 0.97 44 0.92 88 0.96 44 0.9 88 0.97 44 0.9 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 90 0.97 45 0.92 90 0.97 45 0.92								
88 0.97 44 0.9 88 0.97 44 0.92 88 0.96 44 0.9 88 0.97 44 0.9 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 90 0.97 45 0.92 90 0.97 45 0.92								
88 0.97 44 0.92 88 0.96 44 0.9 88 0.97 44 0.9 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.97 45 0.92 89 0.97 45 0.92 90 0.97 45 0.92 90 0.97 45 0.92								
88 0.96 44 0.9 88 0.97 44 0.9 89 0.96 45 0.92 89 0.96 45 0.92 89 0.97 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 90 0.97 45 0.92 90 0.97 45 0.92						0.97		0.9
88 0.97 44 0.9 89 0.96 45 0.92 89 0.96 45 0.92 89 0.97 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.97 45 0.92 89 0.96 45 0.92 90 0.97 45 0.92 90 0.97 45 0.92						0.97	44	
89 0.96 45 0.92 89 0.96 45 0.92 89 0.97 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.97 45 0.92 89 0.96 45 0.92 90 0.97 45 0.92 90 0.97 45 0.92					88	0.96	44	0.9
89 0.96 45 0.92 89 0.97 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.97 45 0.92 89 0.96 45 0.92 90 0.97 45 0.92 90 0.97 45 0.92					88	0.97	44	0.9
89 0.97 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.97 45 0.92 89 0.96 45 0.92 90 0.97 45 0.92 90 0.97 45 0.92	-				89	0.96	45	0.92
89 0.96 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 89 0.97 45 0.92 89 0.96 45 0.92 90 0.97 45 0.92 90 0.97 45 0.92					89	0.96	45	0.92
89 0.96 45 0.92 89 0.96 45 0.92 89 0.97 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 90 0.97 45 0.92 90 0.97 45 0.92					89	0.97	45	0.92
89 0.96 45 0.92 89 0.96 45 0.92 89 0.97 45 0.92 89 0.96 45 0.92 89 0.96 45 0.92 90 0.97 45 0.92 90 0.97 45 0.92					89	0.96	45	0.92
89 0.96 45 0.92 89 0.97 45 0.92 89 0.96 45 0.92 90 0.97 45 0.92 90 0.97 45 0.92 90 0.97 45 0.92								
89 0.97 45 0.92 89 0.96 45 0.92 90 0.97 45 0.92 90 0.97 45 0.92 90 0.97 45 0.92								
89 0.96 45 0.92 90 0.97 45 0.92 90 0.97 45 0.92								
90 0.97 45 0.92 90 0.97 45 0.92								
90 0.97 45 0.92								
1 1 190 1097 145 1092								
1000					90	0.97	45	0.92

I.	I.	I	 1	I.	ı	ı	I
				90	0.97	45	0.9
				90	0.97	45	0.92
				90	0.97	45	0.92
				90	0.96	45	0.92
				90	0.97	45	0.92
				91	0.96	46	0.9
				91	0.97	46	0.9
				91	0.96	46	0.92
				91	0.96	46	0.9
				91	0.96	46	0.9
				91	0.96	46	0.94
				91	0.96	46	0.92
				91	0.96	46	0.92
				92	0.97	46	0.88
				92	0.96	46	0.92
				92	0.97	46	0.92
							0.92
				92	0.97	46	
				92	0.97	46	0.92
				92	0.97	46	0.92
				92	0.97	46	0.92
				92	0.97	46	0.9
				93	0.97	47	0.92
				93	0.97	47	0.94
				93	0.96	47	0.9
				93	0.96	47	0.9
				93	0.96	47	0.9
				93	0.96	47	0.9
				93	0.96	47	0.94
				93	0.97	47	0.94
				94	0.96	47	0.92
				94	0.96	47	0.92
				94	0.97	47	0.9
				94	0.97	47	0.92
				94	0.97	47	0.92
				94	0.97	47	0.9
				94	0.96	47	0.92
				94	0.96	47	0.94
				95	0.97	48	0.92
				95	0.97	48	0.91
				95	0.96	48	0.92
				95	0.96	48	0.92
				95	0.96	48	0.92
				95	0.97	48	0.94
				95	0.96	48	0.92
				 95	0.94	48	0.91
				96	0.96	48	0.94
				96	0.96	48	0.92
				96	0.97	48	0.92
				96	0.97	48	0.92
				96	0.97	48	0.92
				96	0.96	48	0.92
				96	0.97	48	0.92
				96	0.97	48	0.92
				97	0.97	49	0.92
				97	0.97	49	0.92
				97	0.97	49	0.91
				97	0.97	49	0.91
				97	0.96	49	0.92
				97	0.97	49	0.89

1 1			l 1		ı	ı	I
				97	0.96	49	0.91
				97	0.97	49	0.91
				98	0.97	49	0.92
				98	0.97	49	0.91
				98	0.96	49	0.94
				98	0.96	49	0.92
				98	0.97	49	0.92
				98	0.96	49	0.92
				98	0.97	49	0.94
				98	0.96	49	0.92
				99	0.96	50	0.93
				99	0.96	50	0.93
				99	0.97	50	0.93
				99	0.97	50	0.94
				99	0.97	50	0.93
				99	0.97	50	0.94
				99	0.96	50	0.93
				99			
					0.97	50	0.93
				100	0.97	50	0.91
				100	0.97	50	0.94
				100	0.96	50	0.93
				100	0.96	50	0.91
				100	0.96	50	0.93
				100	0.96	50	0.94
				100	0.97	50	0.93
				100	0.96	50	0.91
						51	0.91
						51	0.93
						51	0.93
						51	0.93
						51	0.93
						51	0.94
						51	0.91
						51	0.93
						51	0.91
						51	0.94
						51	0.93
						51	0.93
						51	0.94
						51	0.93
						51	0.91
						51	0.94
						52	0.93
						52	0.93
						52	0.93
						52	0.93
						52	0.95
						52	0.93
						52	0.93
						52	0.91
						52	0.95
						52	0.91
						52	0.91
						52	0.93
						52	0.93
						52	0.91
ı							
						52	0.02
						52	0.93
						52 52 53	0.93 0.95 0.93

1 1			l I		l	l	l
						53	0.91
						53	0.93
						53	0.91
						53	0.9
						53	0.9
						53	0.91
						53	0.95
						53	0.91
						53	0.93
						53	0.93
						53	0.95
						53	0.93
						53	0.95
						53	0.95
						53	0.93
						54	0.93
						54	0.93
						54	0.92
						54	
							0.95
						54	0.93
						54	0.95
						54	0.95
						54	0.93
						54	0.93
						54	0.95
						54	0.93
						54	0.92
						54	0.93
						54	0.92
						54	0.93
						54	0.93
						55	0.93
						55	0.92
						55	0.95
						55	0.93
						55	0.93
						55	0.93
						55	0.95
						55	0.92
						55	0.95
						55	0.93
						55	0.93
						55	0.93
						55	0.93
						55	0.92
						55	0.93
						55	0.92
						56	0.93
						56	0.95
						56	0.93
						56	0.92
						56	0.93
						56	0.93
						56	0.92
						56	0.93
						56	0.92
						56	0.92
						56	0.02
						56 56	0.93

i i	ı	ı	I	1 1	ĺ	ı	ı	i
							56	0.92
							56	0.93
							56	0.9
							56	0.92
							57	0.93
							57	0.93
							57	0.93
							57	0.93
							57	0.93
							57	0.93
							57	0.93
							57	0.93
							57	0.93
							57	0.93
							57	0.93
							57	0.95
							57	0.93
							57	0.93
							57	0.93
							57	0.93
							58	0.93
							58	0.94
								0.94
							58	
							58	0.94
							58	0.94
							58	0.91
							58	0.92
							58	0.95
							58	0.92
							58	0.94
							58	0.94
							58	0.94
							58	0.94
							58	0.94
							58	0.94
							58	0.94
							59	0.94
							59	0.94
							59	0.94
							59	0.95
							59	0.94
							59	0.94
							59	0.95
							59	0.94
							59	0.94
							59	0.94
							59	0.94
							59	0.92
							59	0.95
							59	0.92
							59	0.94
							59	0.94
							60	0.94
							60	0.94
							60	0.94
İ			I .	I			60	0.94
							60	0.02
							60	0.92
							60 60	0.92 0.95 0.94

	I.	l	I	ı	1	l	I	I	I
								60	0.94
Section									
								60	
								60	0.94
								60	0.95
								60	0.94
								60	0.95
								60	0.94
								61	0.94
								61	0.94
								61	0.94
								61	0.94
Method Method<								61	0.94
Method Method<								61	0.94
								61	
Company									
Company									
Company									
Company									
62 0.95 62 0.94 62 0.94 62 0.95 62 0.94 62 0.94 62 0.94 62 0.94 62 0.95 63 0.93 63 0.93 63 0.94 63 0.									
62 0.94 62 0.95 62 0.94 62 0.94 62 0.94 62 0.94 62 0.94 62 0.95 63 0.93 63 0.94 63 0.									
62 0.95 62 0.94 62 0.94 62 0.94 62 0.94 62 0.95 63 0.93 63 0.93 63 0.94									
62 0.94 62 0.94 62 0.94 62 0.94 62 0.95 63 0.93 63 0.94 63 0.									
62 0.94 62 0.94 62 0.95 63 0.93 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.93 63 0.94 63 0.								62	
62 0.94 62 0.95 63 0.93 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.93 63 0.94 63 0.									
62 0.95 63 0.93 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.93 63 0.94 63 0.94 63 0.93 63 0.91 63 0.94 64 0.								62	
63 0.93 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.93 63 0.94 63 0.94 63 0.93 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.95 64 0.96								62	0.94
63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.93 63 0.94 63 0.94 63 0.93 63 0.94 63 0.91 63 0.91 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94								62	0.95
63 0.94 63 0.94 63 0.94 63 0.94 63 0.93 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.91 63 0.91 63 0.94								63	0.93
63 0.94 63 0.94 63 0.93 63 0.94 63 0.94 63 0.94 63 0.93 63 0.91 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.95 64 0.96								63	0.94
63 0.94 63 0.93 63 0.94 63 0.94 63 0.93 63 0.93 63 0.91 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.95 64 0.96								63	0.94
63 0.93 63 0.94 63 0.94 63 0.93 63 0.91 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.95 64 0.96								63	0.94
63 0.94 63 0.94 63 0.93 63 0.91 63 0.91 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.95 64 0.96								63	0.94
63 0.94 63 0.93 63 0.91 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.95 64 0.96								63	0.93
63 0.93 63 0.91 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.95 63 0.95								63	0.94
63 0.91 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.95 63 0.95								63	0.94
63 0.91 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.95 63 0.95								63	0.93
63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.95 64 0.96									0.91
63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.94 63 0.95 64 0.96									
63 0.94 63 0.94 63 0.94 63 0.94 63 0.95 63 0.95									
63 0.94 63 0.94 63 0.95 63 0.95 64 0.96									
63 0.94 63 0.95 64 0.96									
63 0.95 64 0.96									
64 0.96									
0.70									
						<u> </u>		ال	3.70

l I		l	l I	l I		l.	I	I	I
								64	0.96
								64	0.94
								64	0.94
								64	0.96
								64	0.94
								64	0.94
								64	0.94
								64	0.94
								64	0.94
								64	0.94
								64	0.94
								64	0.94
								64	0.93
								64	0.91
								65	0.93
								65	0.94
								65	0.93
								65	0.93
								65	0.93
								65	0.93
								65	0.94
								65	0.96
								65	0.94
								65	0.94
								65	0.96
								65	0.94
								65	0.94
								65	0.94
								65	0.94
								65	0.94
								66	0.94
								66	0.94
								66	0.94
								66	0.94
								66	0.94
								66	0.94
								66	0.94
								66	0.93
								66	0.93
								66	0.94
								66	0.94
								66	0.93
								66	0.94
								66	0.94
								66	0.94
								66	0.94
								67	0.96
								67	0.94
					-			67	0.94
								67	0.94
								67	0.96
								67	0.94
								67	0.96
								67	0.96
								67	0.94
								67	0.94
								67	0.94
								67	0.94

l I		I	l 1		l	l	l
						67	0.94
						67	0.96
						67	0.94
						68	0.94
						68	0.94
						68	0.94
						68	0.94
						68	0.94
						68	0.94
						68	0.94
						68	0.93
						68	0.94
						68	0.96
						68	0.94
						68	0.96
						68	0.94
						68	0.94
						68	0.93
						68	0.94
						69	0.95
						69	0.93
						69	0.95
						69	0.95
						69	0.95
						69	0.96
						69	0.95
						69	0.95
						69	0.96
						69	0.93
						69	0.93
						69	0.92
						69	0.93
						69	0.95
						69	0.95
						69	0.95
						70	0.95
						70	0.96
						70	0.95
						70	0.95
						70	0.95
						70	0.95
						70	0.95
						70	0.95
						70	0.95
						70	0.95
						70	0.93
						70	0.96
						70	0.95
						70	0.95
						70	0.95
						70	0.95
						71	0.96
						71	0.95
						71	0.95
						71	0.95
						71	0.95
					I	* *	J.,,,
						71	0.95
						71	0.95
						71 71 71	0.95 0.95 0.95

ĺ	1	1	I	l I	1	I	l	1
							71	0.95
							71	0.96
							71	0.96
							71	0.95
							71	0.95
							71	0.95
							71	0.96
							71	0.96
							72	0.95
							72	0.95
							72	0.95
							72	0.95
							72	0.94
							72	0.96
							72	0.94
							72	0.92
							72	0.95
							72	0.94
							72	0.95
							72	0.95
							72	0.95
							72	0.95
							72	0.96
							72	0.94
							73	0.95
							73	0.94
							73	0.96
							73	0.95
							73	0.95
							73	0.95
							73	0.92
							73	0.95
							73	0.94
							73	0.94
							73	0.95
							73	0.94
							73	0.95
							73	0.94
							73	0.92
							73	0.95
							74	0.95
							74	0.96
							74	0.95
							74	0.95
							74	0.95
							74	0.95
							74	0.95
							74	0.96
							74	0.94
							74	0.95
							74	0.95
							74	0.95
							74	0.96
							74	0.95
							74	0.96
							74	0.95
							75	0.95
1							1 75	0.95
							75 75	0.95

1 1		l	I	l 1		l	l	l
							75	0.95
							75	0.95
							75	0.96
							75	0.95
							75	0.95
							75	0.95
							75	0.94
							75	0.93
							75	0.93
							75	0.91
							75	0.95
							75	0.95
							75	0.95
							76	0.96
							76	0.95
							76	0.95
							76	0.96
							76	0.96
							76	0.96
							76	0.96
							76	0.95
							76	0.95
							76	0.95
							76	0.95
							76	0.95
							76	0.95
							76	0.94
							76	0.94
							76	0.94
							77	0.95
							77	0.95
							77	0.96
							77	0.94
							77	0.95
							77	0.95
							77	0.95
							77	0.94
							77	0.96
								0.95
							77	
							77	0.96
							77	0.95
							77	0.95
							77	0.96
							77	0.95
							77	0.95
							78	0.95
							78	0.95
							78	0.95
							78	0.95
							78	0.95
							78	0.94
							78	0.95
							78	0.95
							78	0.95
							78	0.95
							78	0.95
						l	, 0	V.75
							78	0.06
							78	0.96
							78 78 78	0.96 0.95 0.94

1 1	l		l 1			l	l
						78	0.94
						78	0.95
						79	0.95
						79	0.96
						79	0.95
						79	0.95
						79	0.95
						79	0.96
						79	0.96
						79	0.95
						79	0.95
						79	0.95
						79	0.94
						79	0.94
						79	0.95
						79	0.95
						79	0.96
-							
						79	0.95
						80	0.96
						80	0.95
						80	0.96
						80	0.96
						80	0.95
						80	0.95
						80	0.94
						80	0.95
						80	0.95
						80	0.95
						80	0.95
						80	0.96
						80	0.95
						80	0.96
						80	0.96
						80	0.96
						81	0.95
						81	0.95
						81	0.95
						81	0.95
						81	0.95
						81	0.95
						81	0.96
						81	0.95
						81	0.95
						81	0.95
						81	0.95
						81	0.95
						81	0.95
						81	0.94
						81	0.94
						81	0.95
						82	0.95
						82	0.96
						82	0.94
						82	0.94
						82	0.96
						82	0.95
						0.4	1 0 95
						82	0.95

l I		l	l I	l I	1	l	l	l
							82	0.95
							82	0.96
							82	0.95
							82	0.95
							82	0.96
							82	0.96
							82	0.95
							83	0.97
							83	0.95
							83	0.95
							83	0.97
							83	0.97
							83	0.97
							83	0.97
							83	0.97
							83	0.95
							83	0.95
							83	0.95
							83	0.95
							83	0.95
							83	0.94
							83	0.94
							83	0.95
							84	0.95
							84	0.95
							84	0.97
							84	0.95
							84	0.95
							84	0.95
							84	0.95
							84	0.95
							84	0.95
							84	0.95
							84	0.95
							84	0.97
							84	0.97
							84	0.95
							84	0.93
							84	0.95
							85	0.97
							85	0.96
							85	0.96
							85	0.97
							85	0.96
							85	0.97
							85	0.97
							85	0.96
							85	0.97
							85	0.96
							85	0.94
							85	0.93
							85	0.94
							85	0.94
							85	0.94
							85	0.94
							86	0.96
							86	0.96
							86 86	0.96

				86 86 86 86	0.96 0.95 0.96 0.96 0.97
				86 86	0.96 0.96
				86	0.96
				86	0.97
				86	0.97
				86	0.96
				86	0.96
				86	0.97
		_		86	0.96
	1			86	0.96
				86	0.96
				87	0.97
				87	0.97
				87	0.96
				87	0.97
				87	0.97
				87	0.97
				87	0.97
				87	0.96
				87	0.96
				87	0.96
				87	0.97
				87	0.96
				87	0.96
				87	0.97
				87	0.95
				87	0.96
				88	0.96
				88	0.96
				88	0.96
				88	0.95
				88	0.96
				88	0.97
					0.96
				88	
				88	0.97
				88	0.95
				88	0.96
				88	0.96
				88	0.96
				88	0.96
				88	0.95
			 	88	0.96
				88	0.95
				89	0.96
				89	0.96
				89	0.97
				89	0.97
				89	0.96
				89	0.96
				89	0.96
				89	0.96
				89	0.96
				89	0.96
				89	0.96
				89	0.97
				89	0.95
				89	0.97
				89	0.97

l I			I	l 1	İ	I	I	l
							89	0.96
							90	0.97
							90	0.96
							90	0.96
							90	0.97
							90	0.96
							90	0.95
							90	0.97
							90	0.96
							90	0.96
							90	0.97
							90	0.96
							90	0.96
							90	0.97
							90	0.97
							90	0.96
							90	0.97
							91	0.96
							91	0.96
							91	0.96
							91	0.96
							91	0.96
							91	0.97
							91	0.96
							91	0.97
							91	0.96
							91	0.96
							91	0.97
							91	0.96
							91	0.96
							91	0.96
							91	0.97
							91	0.96
							92	0.97
							92	0.96
							92	0.96
							92	0.95
							92	0.96
								0.95
							92	
							92	0.96
							92	0.96
							92	0.96
							92	0.97
							92	0.96
							92	0.97
							92	0.96
							92	0.96
							92	0.97
							92	0.97
							93	0.96
							93	0.97
							93	0.97
							93	0.96
							93	0.96
							93	0.96
							93	0.97
					i .	I	177	J V. 2 /
							03	0.96
							93	0.96
							93 93 93	0.96 0.96 0.96

l I			I	l I		İ	I	I	I
								93	0.96
								93	0.97
								93	0.97
								93	0.96
								93	0.96
								93	0.97
								94	0.97
								94	0.96
								94	0.97
								94	0.97
								94	0.97
								94	0.96
								94	0.96
								94	0.97
								94	0.97
								94	0.96
								94	0.96
								94	0.96
								94	0.96
								94	0.96
								94	0.95
								94	0.95
								95	0.96
								95	0.96
								95	0.96
								95	0.96
								95	0.97
								95	0.97
								95	0.97
								95	0.96
								95	0.96
								95	0.97
								95	0.97
								95	0.96
								95	0.96
								95	0.97
								95	0.96
								95	0.96
								96	0.97
								96	0.97
								96	0.97
								96	0.96
								96	0.96
								96	0.96
								96	0.96
								96	0.97
								96	0.96
								96	0.97
					-			96	0.96
								96	0.97
								96	0.96
								96	0.96
								96	0.97
								96	0.97
								97	0.96
								97	0.96
i									
								97	0.96

I	I	I	I	1	1	I	ı	I
							97	0.96
							97	0.97
							97	0.96
							97	0.97
							97	0.95
							97	0.96
							97	0.96
							97	0.97
							97	0.97
							97	0.96
							97	0.96
							98	0.97
							98	0.97
							98	0.97
							98	0.96
							98	0.97
							98	0.97
							98	0.95
							98	0.97
							98	0.96
							98	0.97
							98	0.97
							98	0.96
							98	0.96
								0.90
							98	
							98	0.97
							98	0.96
							99	0.96
							99	0.96
							99	0.97
							99	0.97
							99	0.96
							99	0.96
							99	0.97
							99	0.96
							99	0.97
							99	0.96
							99	0.96
							99	0.97
							99	0.96
							99	0.95
							99	0.96
							99	0.97
							100	0.97
							100	0.96
							100	0.95
							100	0.96
							100	0.97
							100	0.96
							100	0.96
							100	0.97
							100	0.96
							100	0.96
							100	0.97
							100	0.96
							100	0.97
							100	0.97
							100	0.97
							100	0.96

BOINC

						-			
1	0.03	1	0.01	1	0.01	1	0.03	1	0.03
2	0.06	1	0.01	1	0.01	1	0.03	1	0.03
3	0.09	2	0.03	1	0.01	1	0.02	1	0.03
4	0.11	2	0.03	1	0.01	1	0.03	1	0.02
5	0.17	3	0.04	2	0.03	1	0.03	1	0.01
6	0.17	3	0.04	2	0.03	1	0.03	1	0.03
7	0.23	4	0.05	2	0.03	1	0.03	1	0.03
8	0.24	4	0.05	2	0.02	1	0.03	1	0.03
9	0.26	5	0.06	3	0.04	2	0.06	1	0.03
10	0.33	5	0.06	3	0.04	2	0.06	1	0.03
11	0.37	6	0.07	3	0.04	2	0.06	1	0.03
12	0.38	6	0.07	3	0.04	2	0.05	1	0.03
13	0.42	7	0.08	4	0.05	2	0.07	1	0.02
14	0.47	7	0.09	4	0.05	2	0.06	1	0.02
15	0.43	8	0.1	4	0.05	2	0.05	1	0.03
16	0.46	8	0.1	4	0.05	2	0.06	1	0.03
17	0.04	9	0.11	5	0.06	3	0.1	2	0.06
		9		5		3		2	
18	0.51		0.11	5	0.06	3	0.01	2	0.05
19	0.53	10	0.12		0.06		0.07		0.05
20	0.63	10	0.13	5	0.06	3	0.06	2	0.06
21	0.24	11	0.14	6	0.07	3	0.06	2	0.06
22	0.71	11	0.14	6	0.08	3	0.08	2	0.06
23	0.66	12	0.14	6	0.07	3	0.08	2	0.06
24	0.67	12	0.15	6	0.07	3	0.1	2	0.06
25	0.78	13	0.15	7	0.09	4	0.11	2	0.05
26	0.81	13	0.16	7	0.09	4	0.13	2	0.05
27	0.79	14	0.18	7	0.09	4	0.07	2	0
28	0.93	14	0.17	7	0.08	4	0.1	2	0.06
29	0.73	15	0.18	8	0.1	4	0.08	2	0.05
30	0.83	15	0.19	8	0.09	4	0.08	2	0.06
31	0.43	16	0.2	8	0.1	4	0.1	2	0.04
32	0.91	16	0.2	8	0.1	4	0.09	2	0.04
33	1.1	17	0.21	9	0.11	5	0.14	3	0.07
34	0.83	17	0.21	9	0.11	5	0.14	3	0.08
35	0.88	18	0.23	9	0.11	5	0.13	3	0.07
36	1.06	18	0.23	9	0.11	5	0.14	3	0.07
37	1.06	19	0.24	10	0.12	5	0.14	3	0.07
38	1.12	19	0.23	10	0.13	5	0.09	3	0.08
39	1.08	20	0.23	10	0.13	5	0.09	3	0.08
						5		3	
40	1.25	20	0.24	10	0.13		0.12		0.07
41	1	21	0.26	11	0.13	6	0.17	3	0.09
42	1.2	21	0.24	11	0.13	6	0.13	3	0.07
43	1.05	22	0.26	11	0.14	6	0.11	3	0.09
44	1.29	22	0.28	11	0.13	6	0.13	3	0.07
45	1.5	23	0.29	12	0.15	6	0.17	3	0.08
46	1.59	23	0.27	12	0.15	6	0.15	3	0.07
47	1.57	24	0.28	12	0.14	6	0.14	3	0.06
48	1.41	24	0.3	12	0.15	6	0.15	3	0.09
49	1.4	25	0.3	13	0.16	7	0.17	4	0.07
50	1.22	25	0.29	13	0.15	7	0.17	4	0.1
51	1.31	26	0.31	13	0.15	7	0.15	4	0.1
52	1.27	26	0.3	13	0.15	7	0.15	4	0.08
53	1.33	27	0.32	14	0.16	7	0.16	4	0.1
54	1.54	27	0.31	14	0.16	7	0.14	4	0.09
55	1.38	28	0.33	14	0.18	7	0.14	4	0.1
56	1.51	28	0.35	14	0.16	7	0.18	4	0.09
57	1.43	29	0.34	15	0.17	8	0.21	4	0.06
58	1.93	29	0.34	15	0.17	8	0.21	4	0.00
20	1./3	-/	U.JT	ر ،	V.1 /	٧	V.2	1	V.1

59	1.44	30	0.37	15	0.18	8	0.16	4	0.1
60	1.46	30	0.35	15	0.18	8	0.16	4	0.09
61	1.69	31	0.36	16	0.2	8	0.15	4	0.09
62	1.59	31	0.36	16	0.19	8	0.17	4	0.08
63	0.74	32	0.38	16	0.19	8	0.22	4	0.07
64	1.56	32	0.38	16	0.18	8	0.17	4	0.11
65	1.86	33	0.38	17	0.2	9	0.2	5	0.07
66	1.89	33	0.39	17	0.2	9	0.23	5	0.1
67	1.91	34	0.42	17	0.2	9	0.23	5	0.11
68	1.66	34	0.4	17	0.2	9	0.19	5	0.13
69	1.73	35	0.41	18	0.21	9	0.28	5	0.12
70	1.67	35	0.41	18	0.21	9	0.2	5	0.13
71	1.73	36	0.45	18	0.21	9	0.2	5	0.12
72	1.89	36	0.42	18	0.21	9	0.23	5	0.1
73	1.97	37	0.44	19	0.22	10	0.25	5	0.16
74	2.06	37	0.43	19	0.22	10	0.21	5	0.11
75	2.14	38	0.45	19	0.22	10	0.23	5	0.13
76	2.17	38	0.45	19	0.22	10	0.25	5	0.11
77	1.88	39	0.45	20	0.25	10	0.17		0.1
78 79	2.23	39 40	0.48	20	0.24	10	0.24	5	0.11
80	2.19	40	0.47	20	0.23	10	0.23	5	0.11
81	2.61	41	0.3	21	0.24	11	0.02	6	0.12
82	2.28	41	0.48	21	0.26	11	0.26	6	0.14
83	2.08	42	0.49	21	0.25	11	0.28	6	0.15
84	2.21	42	0.49	21	0.25	11	0.27	6	0.15
85	2.13	43	0.51	22	0.26	11	0.28	6	0.14
86	2.21	43	0.51	22	0.26	11	0.31	6	0.15
87	2.64	44	0.52	22	0.27	11	0.27	6	0.13
88	2.44	44	0.49	22	0.26	11	0.28	6	0.11
89	2.23	45	0.55	23	0.27	12	0.3	6	0.11
90	2.57	45	0.53	23	0.27	12	0.32	6	0.14
91	2.33	46	0.54	23	0.27	12	0.33	6	0.11
92	2.24	46	0.54	23	0.27	12	0.33	6	0.13
93	2.38	47	0.55	24	0.28	12	0.27	6	0.11
94	2.69	47	0.55	24	0.28	12	0.26	6	0.13
95	2.38	48	0.56	24	0.3	12	0.33	6	0.15
96	2.4	48	0.56	24	0.3	12	0.3	6	0.13
97	2.43	49	0.57	25	0.29	13	0.35	7	0.13
98	2.65	49	0.57	25	0.29	13	0.31	7	0.13
99	2.91	50	0.59	25	0.29	13	0.24	7	0.15
100	2.5	50	0.59	25	0.29	13	0.32	7	0.15
		51	0.07	26	0.31	13	0.27	7	0.17
		51	0.6	26	0.31	13	0.03	7	0.15
		52	0.61	26	0.31	13	0.33	7	0.2
		52	0.6	26	0.29	13	0.32	7	0.17
		53	0.62	27	0.32	14	0.03	7	0.17
		53	0.62	27	0.3	14	0.3	7	0.13
		54	0.64	27	0.31	14	0.32	7	0.18
		54	0.64	27	0.32	14	0.32	7	0.16
		55	0.64	28	0.06	14	0.36	7	0.15
		55	0.65	28	0.33	14	0.3	7	0.16
		56	0.66	28	0.35	14	0.34	7	0.15
		56	0.09	28	0.33	14	0.35	7	0.17
		57	0.67	29	0.34	15	0.24	8	0.17
		57	0.67	29	0.06	15	0.35	8	0.17
		58	0.67	29	0.34	15	0.38	8	0.16
	1	58	0.64	29	0.32	15	0.3	8	0.02

1	l 50	0.60	120	0.25	15	L 0.21	١.	0.16
	59	0.69	30	0.35	15	0.31	8	0.16
	60	0.7	30	0.35	15	0.38	8	0.12
	60	0.67	30	0.33	15	0.43	8	0.17
	61	0.09	31	0.36	16	0.4	8	0.22
	61	0.68	31	0.36	16	0.31	8	0.18
	62	0.73	31	0.36	16	0.4	8	0.17
	62	0.73	31	0.36	16	0.39	8	0.2
	63	0.73	32	0.37	16	0.43	8	0.2
	63	0.72	32	0.36	16	0.41	8	0.23
	64	0.74	32	0.38	16	0.39	8	0.19
	64	0.75	32	0.38	16	0.31	8	0.16
	65	0.75	33	0.39	17	0.43	9	0.26
	65	0.68	33	0.38	17	0.44	9	0.2
	66	0.72	33	0.37	17	0.38	9	0.18
	66	0.78	33	0.38	17	0.37	9	0.2
	67	0.78	34	0.4	17	0.3	9	0.2
	67	0.74	34	0.38	17	0.41	9	0.18
	68	0.8	34	0.4	17	0.37	9	0.2
	68	0.79	34	0.39	17	0.3	9	0.15
	69	0.8	35	0.41	18	0.04	9	0.23
	69	0.8	35	0.38	18	0.44	9	0.16
	70	0.77	35	0.39	18	0.39	9	0.18
	70	0.78	35	0.41	18	0.39	9	0.14
	71	0.83	36	0.42	18	0.41	9	0.17
	71	0.83	36	0.42	18	0.32	9	0.16
	72	0.8	36	0.42	18	0.38	9	0.18
	72	0.85	36	0.4	18	0.39	9	0.19
	73	0.81	37	0.43	19	0.4	10	0.24
	73	0.85	37	0.41	19	0.37	10	0.2
	74	0.81	37	0.39	19	0.37	10	0.2
	74	0.81	37	0.43	19	0.37	10	0.22
	75	0.82	38	0.41	19	0.22	10	0.21
	75	0.82	38	0.44	19	0.4	10	0.19
	76	0.84	38	0.42	19	0.4	10	0.2
	76	0.84	38	0.42	19	0.33	10	0.19
	77	0.85	39	0.45	20	0.43	10	0.19
	77	0.2	39	0.43	20	0.33	10	0.22
	78	0.91	39	0.43	20	0.41	10	0.2
	78	0.87	39	0.43	20	0.36	10	0.2
	79	0.88	40	0.47	20	0.43	10	0.22
	79	0.87	40	0.44	20	0.32	10	0.2
	80	0.92	40	0.44	20	0.43	10	0.15
	80	0.88	40	0.44	20	0.43	10	0.13
	81	0.88	41	0.44	21	0.43	11	0.28
	81	0.89	41	0.48	21	0.37	11	0.28
	82	0.89	41	0.46	21	0.37	11	0.21
	82	0.91	41	0.46	21	0.34	11	0.25
	83	0.94	42	0.46	21	0.46	11	0.23
	83		42	0.49	21		11	0.23
		0.86				0.31		
	84	0.92	42	0.47	21	0.46	11	0.23
	84	0.92	42	0.47	21	0.45	11	0.23
	85	0.99	43	0.5	22	0.43	11	0.25
	85	0.99	43	0.48	22	0.39	11	0.24
	86	0.95	43	0.48	22	0.43	11	0.23
	86	0.96	43	0.5	22	0.43	11	0.21
	87	0.97	44	0.49	22	0.43	11	0.22
	87	0.96	44	0.51	22	0.48	11	0.27
	88	0.97	44	0.49	22	0.32	11	0.22
	88	0.96	44	0.49	22	0.39	11	0.25

	89	0.99	45	0.5	23	0.34	12	0.24
	89	0.98	45	0.49	23	0.4	12	0.26
	90	0.99	45	0.49	23	0.56	12	0.26
	90	0.99	45	0.49	23	0.5	12	0.24
	91	1	46	0.51	23	0.43	12	0.29
	91	1	46	0.51	23	0.47	12	0.19
	92	1.01	46	0.51	23	0.33	12	0.29
	92	1.01	46	0.51	23	0.51	12	0.21
	93	1.03	47	0.52	24	0.39	12	0.27
	93	1.01	47	0.52	24	0.39	12	0.21
	94	1.04	47	0.52	24	0.44	12	0.19
	94	1.03	47	0.51	24	0.47	12	0.15
	95	1.04	48	0.53	24	0.47	12	0.29
	95	0.99	48	0.53	24	0.47	12	0.26
	96	1.04	48	0.53	24	0.4	12	0.25
	96	1.05	48	0.53	24	0.48	12	0.29
	97	1.07	49	0.53	25	0.56	13	0.21
	97	1.07	49	0.53	25	0.49	13	0.25
	98	1.08	49	0.54	25	0.44	13	0.25
	98	1.01	49	0.54	25	0.47	13	0.23
	99	1.09	50	0.54	25	0.44	13	0.18
	99	1.09	50	0.55	25	0.49	13	0.25
	100	1.1	50	0.56	25	0.54	13	0.26
	100	1.1	50	0.55	25	0.42	13	0.26
			51	0.55	26	0.42	13	0.28
			51	0.55	26	0.46	13	0.23
			51	0.56	26	0.49	13	0.29
			51	0.55	26	0.63	13	0.25
			52	0.58	26	0.43	13	0.25
			52	0.57	26	0.57	13	0.28
			52	0.57	26	0.46	13	0.29
			52	0.57	26	0.2	13	0.25
			53	0.58	27	0.49	14	0.21
			53	0.58	27	0.56	14	0.23
			53	0.55	27	0.4	14	0.24
			53	0.55	27	0.47	14	0.23
			54	0.59	27	0.4	14	0.25
			54	0.59	27	0.59	14	0.24
			54	0.59	27	0.48	14	0.25
			54	0.55	27	0.37	14	0.19
			55	0.6	28	0.53	14	0.24
			55	0.6	28	0.44	14	0.27
			55	0.6	28	0.45	14	0.27
			55	0.57	28	0.41	14	0.27
			56	0.58	28	0.48	14	0.2
			56	0.62	28	0.49	14	0.25
			56	0.58	28	0.47	14	0.21
			56	0.62	28	0.45	14	0.2
			57	0.63	29	0.29	15	0.23
			57	0.59	29	0.39	15	0.23
			57	0.62	29	0.47	15	0.23
			57	0.59	29	0.58	15	0.24
			58	0.6	29	0.58	15	0.24
			58	0.63	29	0.51	15	0.26
			58	0.63	29	0.57		0.24
							15	
			58	0.6	29	0.46	15	0.27
			59	0.64	30	0.48	15	0.22

1	I	I	I	I	l.	l.	ı	ı	I
				59	0.65	30	0.53	15	0.26
				60	0.63	30	0.48	15	0.27
				60	0.65	30	0.52	15	0.21
				60	0.66	30	0.48	15	0.26
				60	0.63	30	0.48	15	0.24
				61	0.63	31	0.65	16	0.25
				61	0.63	31	0.42	16	0.25
				61	0.64	31	0.62	16	0.28
				61	0.66	31	0.39	16	0.24
				62	0.65	31	0.35	16	0.28
				62	0.65	31	0.6	16	0.23
				62	0.65	31	0.46	16	0.25
				62	0.64	31	0.46	16	0.13
				63	0.66	32	0.52	16	0.26
				63	0.65	32	0.43	16	0.27
				63	0.68	32	0.47	16	0.26
				63	0.66	32	0.62	16	0.26
				64	0.67	32	0.62	16	0.28
				64	0.67	32	0.48	16	0.28
				64	0.67	32	0.51	16	0.28
				64	0.67	32	0.44	16	0.28
				65	0.68	33	0.48	17	0.35
				65	0.68	33	0.65	17	0.31
				65	0.68	33	0.49	17	0.25
				65	0.68	33	0.58	17	0.34
				66	0.72	33	0.61	17	0.35
				66	0.69	33	0.58	17	0.27
				66	0.68	33	0.48	17	0.25
				66	0.69	33	0.49	17	0.25
				67	0.7	34	0.53	17	0.27
				67	0.69	34	0.57	17	0.27
				67	0.73	34	0.51	17	0.33
				67	0.69	34	0.59	17	0.29
				68	0.7	34	0.52	17	0.29
				68	0.71	34	0.51	17	0.22
				68	0.7	34	0.6	17	0.24
				68	0.71	34	0.67	17	0.22
				69	0.68	35	0.56	18	0.27
				69	0.71	35	0.55	18	0.23
				69	0.71	35	0.57	18	0.29
				69	0.72	35	0.45	18	0.19
				70	0.71	35	0.57	18	0.32
				70	0.73	35	0.65	18	0.26
				70	0.73	35	0.52	18	0.3
				70	0.74	35	0.35	18	0.32
				71	0.74	36	0.51	18	0.34
				71	0.74	36	0.54	18	0.24
				71	0.74	36	0.64	18	0.3
				71	0.74	36	0.67	18	0.27
				72	0.75	36	0.49	18	0.26
				72	0.75	36	0.62	18	0.27
				72	0.75	36	0.38	18	0.26
				72	0.74	36	0.58	18	0.29
				73	0.75	37	0.6	19	0.28
				73	0.87	37	0.5	19	0.26
				73	0.75	37	0.54	19	0.26
				73	0.72	37	0.62	19	0.28
				74	0.76	37	0.5	19	0.25
				74	0.77	37	0.73	19	0.33

1 1	I	I	I.	I.	I	I	I
		74	0.76	37	0.64	19	0.28
		74	0.77	37	0.45	19	0.33
		75	0.74	38	0.55	19	0.26
		75	0.78	38	0.61	19	0.29
		75	0.77	38	0.66	19	0.27
		75	0.78	38	0.6	19	0.3
		76	0.79	38	0.57	19	0.24
		76	0.79	38	0.61	19	0.28
		76	0.79	38	0.6	19	0.31
		76	0.79	38	0.52	19	0.27
		77	0.79	39	0.62	20	0.27
		77	0.77	39	0.58	20	0.3
		77	0.79	39	0.57	20	0.3
		77	0.79	39	0.53	20	0.26
		78	0.8	39	0.62	20	0.29
		78	0.8	39	0.57	20	0.29
		78	0.75	39	0.53	20	0.2
		78	0.8	39	0.76	20	0.27
		79	0.77	40	0.59	20	0.26
	+	79	0.82	40	0.5	20	0.24
		79	0.82	40	0.5	20	0.29
		79	0.81	40	0.51	20	0.27
		80	0.79	40	0.48	20	0.33
	+	80	0.79	40	0.48	20	0.33
		80	0.95	40	0.54	20	0.32
		80	0.83	40	0.54	20	0.29
				41		21	
		81	0.84		0.49		0.25
		81	0.79	41	0.53	21	0.29
		81	0.84	41	0.56	21	0.31
		81	0.84	41	0.79	21	0.28
		82	0.81	41	0.56	21	0.24
		82	0.84	41	0.56	21	0.27
		82	0.8	41	0.56	21	0.28
		82	0.99	41	0.62	21	0.29
		83	0.99	42	0.53	21	0.31
		83	0.81	42	0.54	21	0.28
		83	0.86	42	0.58	21	0.25
		83	0.85	42	0.61	21	0.22
		84	0.87	42	0.61	21	0.31
		84	0.82	42	0.61	21	0.27
		84	0.87	42	0.58	21	0.31
		84	0.82	42	0.69	21	0.28
		85	0.84	43	0.52	22	0.32
		85	0.83	43	0.64	22	0.32
		85	0.84	43	0.57	22	0.15
		85	0.83	43	0.61	22	0.35
		86	0.83	43	0.53	22	0.26
		86	0.86	43	0.58	22	0.31
		86	0.83	43	0.45	22	0.3
		86	0.84	43	0.62	22	0.28
		87	0.85	44	0.56	22	0.26
		87	0.86	44	0.52	22	0.3
			0.9	44	0.6	22	0.23
		87	0.9				
		87	0.89	44	0.59	22	0.24
		87	0.89	44		22	
		87 88	0.89	44	0.47		0.3
		87 88 88	0.89 0.91 0.91	44 44 44	0.47 0.52	22	0.3
		87 88 88 88	0.89 0.91 0.91 0.87	44 44 44 44	0.47 0.52 0.53	22 22 22	0.3 0.28 0.3
		87 88 88	0.89 0.91 0.91	44 44 44	0.47 0.52	22	0.3

1	1	I	l	0.00	1.5	L	l 22	0.20
			89	0.88	45	0.62	23	0.29
			89	0.87	45	0.62	23	0.29
			89	0.87	45	0.54	23	0.22
			90	0.88	45	0.66	23	0.28
			90	0.88	45	0.74	23	0.31
			90	0.89	45	0.58	23	0.32
			90	0.89	45	0.61	23	0.29
			91	0.89	46	0.79	23	0.34
			91	0.9	46	0.55	23	0.4
			91	0.89	46	0.57	23	0.17
			91	0.9	46	0.64	23	0.31
			92	0.89	46	0.88	23	0.29
			92	0.92	46	0.61	23	0.32
			92	0.91	46	0.63	23	0.4
			92	0.9	46	0.73	23	0.31
			93	0.91	47	0.64	24	0.31
			93	0.92	47	0.6	24	0.36
			93	0.91	47	0.76	24	0.35
			93	0.92	47	0.7	24	0.3
			94	0.92	47	0.59	24	0.28
			94	0.92	47	0.64	24	0.27
			94	0.92	47	0.64	24	0.31
			94	0.92	47	0.59	24	0.24
			95	0.92	48	0.61	24	0.24
			95	0.94	48	0.61	24	0.39
			95	0.94	48	0.72	24	0.31
			95	0.93	48	0.51	24	0.27
			96	0.95	48	0.66	24	0.32
			96	0.94	48	0.62	24	0.31
			96	0.95	48	0.77	24	0.35
			96	0.94	48	0.65	24	0.3
			97	0.94	49	0.69	25	0.3
			97	0.95	49	0.58	25	0.25
			97	0.95	49	0.61	25	0.29
			97	0.95	49	0.64	25	0.25
			98	0.95	49	0.56	25	0.25
			98	0.92	49	0.58	25	0.32
			98	0.96	49	0.63	25	0.31
			98	0.96	49	0.62	25	0.3
			99	0.97	50	0.56	25	0.3
			99	0.97	50	0.63	25	0.32
			99	0.97	50	0.64	25	0.3
			99	0.92	50	0.58	25	0.32
			100	0.98	50	0.56	25	0.34
			100	0.98	50	0.74	25	0.31
			100	0.98	50	0.75	25	0.28
			100	0.93	50	0.6	25	0.25
					51	0.6	26	0.31
					51	0.61	26	0.31
					51	0.61	26	0.29
					51	0.61	26	0.35
					51	0.57	26	0.33
					51	0.57	26	0.3
					51	0.61	26	0.26
					51	0.61	26	0.33
					52	0.63	26	0.33
					52	0.67	26	0.29
					52	0.66	26	0.27
					52	0.76	26	0.32
		<u> </u>		<u> </u>			1	J.J.

I.	I.	I	 	I.	ı	ı	I
				52	0.62	26	0.31
				52	0.8	26	0.31
				52	0.62	26	0.27
				52	0.63	26	0.31
				53	0.67	27	0.32
				53	0.72	27	0.41
				53	0.68	27	0.33
				53	0.5	27	0.3
				53	0.62	27	0.34
				53	0.6	27	0.33
				53	0.68	27	0.32
				53	0.63	27	0.35
				54	0.68	27	0.28
				54	0.81	27	0.34
				54	0.57	27	0.34
				54	0.64	27	0.32
				54	0.69	27	0.32
				54	0.61	27	0.34
				54	0.59	27	0.37
				54	0.64	27	0.33
				55	0.7	28	0.3
				55	0.65	28	0.33
				55	0.65	28	0.36
				55	0.73	28	0.33
				55	0.62	28	0.35
				55	0.65	28	0.37
				55	0.68	28	0.32
				55	0.65	28	0.31
				56	0.63	28	0.41
				56	0.62	28	0.39
				56	0.73	28	0.36
				56	0.65	28	0.33
				56	0.6	28	0.29
				56	0.64	28	0.29
				56	0.59	28	0.33
				56	0.63	28	0.34
				57	0.83	29	0.33
				57	0.67	29	0.23
				57	0.63	29	0.33
				57	0.54	29	0.37
				57	0.54	29	0.35
				57	0.69	29	0.39
				 57	0.63	29	0.35
				57	0.6	29	0.32
				58	0.64	29	0.4
				58	0.84	29	0.35
				58	0.6	29	0.34
				58	0.64	29	0.33
				58	0.64	29	0.32
				58	0.64	29	0.32
				58	0.66	29	0.35
				58	0.57	29	0.37
				59	0.81	30	0.37
				59	0.7	30	0.33
				59	0.66	30	0.38
				59	0.79	30	0.35
				59	0.76	30	0.35
				59	0.71	30	0.32
				59	0.66	30	0.34

I.	1	I	 	I	ı	ı	I
				59	0.69	30	0.4
				60	0.71	30	0.36
				60	0.64	30	0.32
				60	0.67	30	0.38
				60	0.67	30	0.35
				60	0.65	30	0.36
				60	0.76	30	0.32
				60	0.71	30	0.32
				60	0.65	30	0.28
				61	0.73	31	0.37
				61	0.68	31	0.36
				61	0.58	31	0.34
				61	0.54	31	0.3
				61	0.64	31	0.33
				61	0.67	31	0.36
				61			0.30
					0.64	31	
				61	0.63	31	0.36
				62	0.7	31	0.34
				62	0.75	31	0.31
				62	0.69	31	0.35
				62	0.69	31	0.33
				62	0.59	31	0.31
				62	0.7	31	0.34
				62	0.65	31	0.39
				62	0.66	31	0.33
				63	0.6	32	0.38
				63	0.63	32	0.31
				63	0.66	32	0.35
				63	0.66	32	0.4
				63	0.66	32	0.32
				63	0.6	32	0.32
				63	0.65	32	0.33
				63	0.83	32	0.33
				64	0.67	32	0.34
				64	0.73	32	0.32
				64	0.74	32	0.34
				64	0.68	32	0.38
				64	0.7	32	0.48
				64	0.68	32	0.34
				64	0.67	32	0.36
				64	0.63	32	0.3
				65	0.82	33	0.34
				 65	0.71	33	0.39
				65	0.83	33	0.33
				65	0.77	33	0.34
				65	0.72	33	0.35
				65	0.68	33	0.34
				65	0.72	33	0.37
				65	0.68	33	0.33
				66	0.73	33	0.37
				66	0.71	33	0.31
				66	0.7	33	0.33
				66	0.7	33	0.33
				66	0.67	33	0.32
				66	0.69	33	0.34
				66	0.69	33	0.37
				66	0.69	33	0.34
				67	0.63	34	0.34
				67	0.71	34	0.33

				67	0.66	34	0.35
				67	0.7	34	0.34
				67	0.71	34	0.35
				67	0.7	34	0.36
				67	0.67	34	0.37
				67	0.67	34	0.32
				68	0.72	34	0.38
				68	0.71	34	0.34
				68	0.93	34	0.34
				68	0.72	34	0.32
				68	0.75	34	0.36
				68	0.6	34	0.31
				68	0.68	34	0.34
				68	0.99	34	0.43
				69	0.73	35	0.36
				69	0.69	35	0.3
				69	0.77	35	0.34
				69	0.78	35	0.33
				69	0.63	35	0.35
				69	0.76	35	0.35
				69	0.7	35	0.32
				69	0.64	35	0.3
				70	0.78	35	0.36
				70	0.83	35	0.35
				70	0.74	35	0.36
				70	0.73	35	0.29
				70	0.72	35	0.51
				70	0.7	35	0.34
				70	0.7	35	0.35
				70	0.63	35	0.33
							-
				71	0.61	36	0.32
				71	0.82	36	0.32
				71	0.72	36	0.35
				71	0.66	36	0.4
				71	0.68	36	0.29
				71	0.89	36	0.38
				71	0.68	36	0.35
				71	0.79	36	0.32
				72	0.71	36	0.31
				72	0.76	36	0.36
				72	0.67	36	0.34
				72	0.64	36	0.36
				72	0.71	36	0.34
				72	0.64	36	0.34
				72	0.85	36	0.33
				72	0.79	36	0.36
				73	0.72	37	0.37
				73	0.69	37	0.37
				73	0.76	37	0.41
				73	0.66	37	0.36
				73	0.72	37	0.37
				73	0.72	37	0.35
				73	0.72	37	0.33
				73	0.7	37	0.37
				74	0.62	37	0.35
				74	0.62	37	0.33
				74	0.66	37	0.37
		1	ii	/ =	U.13	-/	0.57
				74	0.67	37	0.32

I.	I.	I	1		I.	ı	ı	I
					74	0.7	37	0.37
					74	0.63	37	0.41
					74	0.69	37	0.36
					75	0.75	38	0.39
					75	0.7	38	0.38
					75	0.64	38	0.34
					75	0.72	38	0.34
					75	0.7	38	0.4
					75	0.71	38	0.36
					75	0.71	38	0.31
					75	0.71	38	0.36
					76	0.75	38	0.48
					76	0.71	38	0.38
					76	0.8	38	0.36
					76	0.96	38	0.38
					76	0.68	38	0.34
					76	0.7	38	0.35
					76	0.67	38	0.38
					76	0.75	38	0.36
					77	0.72	39	0.36
					77	0.69	39	0.38
					77	0.73	39	0.35
					77	0.7	39	0.41
					77	0.72	39	0.36
					77	0.72	39	0.36
					77	0.69	39	0.39
					77	0.72	39	0.32
					78	0.73	39	0.35
					78	0.73	39	0.41
					78	0.74	39	0.35
					78	0.7	39	0.35
					78	0.77	39	0.34
					78	0.69	39	0.37
					78	0.67	39	0.38
					78	0.78	39	0.33
					79	0.7	40	0.34
					79	0.79	40	0.37
					79	0.71	40	0.42
					79	0.75	40	0.32
					79	0.74	40	0.38
					79	0.71	40	0.36
					79	0.71	40	0.37
					79	0.71	40	0.4
					80	0.88	40	0.36
					80	0.72	40	0.34
					80	0.89	40	0.38
					80	0.71	40	0.38
					80	0.74	40	0.38
					80	0.75	40	0.35
					80	0.72	40	0.39
					80	0.72	40	0.39
					81	0.84	41	0.36
					81	0.7	41	0.38
					81	0.76	41	0.38
					81			
						0.76	41	0.37
					81	0.76	41	0.34
					81	0.73	41	0.42
					81	0.81	41	0.38
					81	0.71	41	0.38

1	1			I.			1
				82	0.73	41	0.35
				82	0.73	41	0.35
				82	0.74	41	0.32
				82	0.69	41	0.35
				82	0.67	41	0.38
				82	0.73	41	0.36
				82	0.73	41	0.35
				82	0.67	41	0.29
				83	0.69	42	0.4
				83	0.7	42	0.38
				83	0.71	42	0.38
				83	0.74	42	0.35
				83	0.71	42	0.42
				83	0.82	42	0.37
				83	0.75	42	0.33
				83	0.72	42	0.34
				84	0.72	42	0.39
				84	0.75	42	0.46
				84	0.76	42	0.4
				84	0.79	42	0.35
				84	0.74	42	0.37
				84	0.84	42	0.34
				84	0.71	42	0.45
				84	0.88	42	0.32
				85	0.77	43	0.38
				85	0.73	43	0.36
				85	0.73	43	0.37
				85	0.72	43	0.45
				85	0.76	43	0.41
				85	0.79	43	0.39
				85	0.69	43	0.34
				85	0.77	43	0.36
				86	0.77	43	0.37
				86	0.7	43	0.37
				86	0.82	43	0.39
				86	0.73	43	0.37
				86	0.7	43	0.37
				86	0.7	43	0.37
				86	0.74	43	0.37
				86	0.81	43	0.38
				87	0.77	44	0.38
				87	0.74	44	0.37
				87	0.71	44	0.38
				87	0.74	44	0.38
				87	0.71	44	0.38
				87	0.78	44	0.41
				87	0.75	44	0.37
				87	0.78	44	0.31
				88	0.78	44	0.36
				88	0.72	44	0.36
				88	0.76	44	0.37
					0.72	44	0.36
				88			
				88	0.74	44	0.38
				88	0.75	44	0.37
				88	0.76	44	0.37
				88	0.72	44	0.37
				89	0.72	45	0.37
				89	0.73	45	0.38
				89	0.79	45	0.45
	-		-				

			92	0.72	46	0.36
			92	0.74	46	0.37
			92	0.8	46	0.41
			92	0.75	46	0.36
			92	0.77	46	0.4
			92	1.02	46	0.39
			93	0.73	47	0.38
			93	0.76	47	0.35
			93	0.73	47	0.38
			93	0.82	47	0.4
			93	0.82	47	0.42
			93	0.72	47	0.37
			93	0.72	47	0.37
					47	0.37
			0.4	0.63		0.57
			94	0.83		0.38
			94	0.94	47	0.38
						0.38 0.44 0.4
			94 94	0.94 0.76	47	0.44
			94 94 94	0.94 0.76 0.79	47 47 47	0.44
			94 94 94 94	0.94 0.76 0.79 0.76 0.73	47 47 47 47	0.44 0.4 0.3
			94 94 94 94 94	0.94 0.76 0.79 0.76	47 47 47 47 47	0.44 0.4 0.3 0.34
			94 94 94 94 94 94	0.94 0.76 0.79 0.76 0.73	47 47 47 47 47 47	0.44 0.4 0.3 0.34 0.4
			94 94 94 94 94 94 94	0.94 0.76 0.79 0.76 0.73 0.73	47 47 47 47 47 47 47	0.44 0.4 0.3 0.34 0.4 0.35
			94 94 94 94 94 94 94 95	0.94 0.76 0.79 0.76 0.73 0.73 0.73	47 47 47 47 47 47 47 47 48	0.44 0.4 0.3 0.34 0.4 0.35 0.41
			94 94 94 94 94 94 94 95 95	0.94 0.76 0.79 0.76 0.73 0.73 0.73 0.73 0.73	47 47 47 47 47 47 47 47 48 48	0.44 0.4 0.3 0.34 0.4 0.35 0.41 0.38
			94 94 94 94 94 94 94 95 95	0.94 0.76 0.79 0.76 0.73 0.73 0.73 0.73 0.74 0.78	47 47 47 47 47 47 47 47 48 48	0.44 0.4 0.3 0.34 0.4 0.35 0.41 0.38 0.38
			94 94 94 94 94 94 95 95 95	0.94 0.76 0.79 0.76 0.73 0.73 0.73 0.73 0.74 0.78 0.77	47 47 47 47 47 47 47 47 48 48 48	0.44 0.4 0.3 0.34 0.4 0.35 0.41 0.38 0.38
			94 94 94 94 94 94 95 95 95 95	0.94 0.76 0.79 0.76 0.73 0.73 0.73 0.73 0.74 0.78 0.77	47 47 47 47 47 47 47 47 48 48 48 48	0.44 0.4 0.3 0.34 0.4 0.35 0.41 0.38 0.38 0.38
			94 94 94 94 94 94 95 95 95 95 95 95 95	0.94 0.76 0.79 0.76 0.73 0.73 0.73 0.73 0.74 0.78 0.77 0.78	47 47 47 47 47 47 47 48 48 48 48 48	0.44 0.4 0.3 0.34 0.4 0.35 0.41 0.38 0.38 0.38 0.45 0.38
			94 94 94 94 94 94 95 95 95 95 95 95	0.94 0.76 0.79 0.76 0.73 0.73 0.73 0.73 0.74 0.78 0.77 0.78 0.77	47 47 47 47 47 47 47 48 48 48 48 48 48	0.44 0.4 0.3 0.34 0.4 0.35 0.41 0.38 0.38 0.38 0.45 0.38 0.37
			94 94 94 94 94 94 95 95 95 95 95 95 95	0.94 0.76 0.79 0.76 0.73 0.73 0.73 0.73 0.74 0.78 0.77 0.78 0.75 0.74	47 47 47 47 47 47 47 48 48 48 48 48 48 48	0.44 0.4 0.3 0.34 0.4 0.35 0.41 0.38 0.38 0.38 0.45 0.38 0.37 0.41
			94 94 94 94 94 94 95 95 95 95 95 95 95 96	0.94 0.76 0.79 0.76 0.73 0.73 0.73 0.74 0.78 0.77 0.78 0.75 0.74 0.81	47 47 47 47 47 47 47 48 48 48 48 48 48 48 48	0.44 0.4 0.3 0.34 0.4 0.35 0.41 0.38 0.38 0.38 0.45 0.38 0.37 0.41 0.39
			94 94 94 94 94 94 95 95 95 95 95 95 96 96	0.94 0.76 0.79 0.76 0.73 0.73 0.73 0.73 0.74 0.78 0.77 0.78 0.75 0.74 0.81 0.86 0.76 0.75 0.78	47 47 47 47 47 47 48 48 48 48 48 48 48 48 48 48	0.44 0.4 0.3 0.34 0.4 0.35 0.41 0.38 0.38 0.38 0.38 0.45 0.38 0.37 0.41 0.39 0.37 0.32 0.47
			94 94 94 94 94 94 95 95 95 95 95 95 96 96	0.94 0.76 0.79 0.76 0.73 0.73 0.73 0.73 0.74 0.78 0.77 0.78 0.75 0.74 0.86 0.76 0.75	47 47 47 47 47 47 48 48 48 48 48 48 48 48 48 48	0.44 0.4 0.3 0.34 0.4 0.35 0.41 0.38 0.38 0.38 0.38 0.45 0.38 0.37 0.41 0.39 0.37

I.	I.	I	1	1	I.	ı	ı	I
					96	0.85	48	0.45
					96	0.81	48	0.41
					97	0.8	49	0.32
					97	0.76	49	0.35
					97	0.75	49	0.38
					97	0.75	49	0.35
					97	0.77	49	0.38
					97	0.77	49	0.4
					97	0.75	49	0.39
					97	0.87	49	0.38
					98	0.76	49	0.34
					98	0.75	49	0.42
					98	0.84	49	0.36
					98	0.77	49	0.38
					98	0.77	49	0.38
					98			0.32
						0.77	49	
					98	0.83	49	0.49
					98	0.76	49	0.38
					99	0.88	50	0.38
					99	0.8	50	0.41
					99	0.77	50	0.38
					99	0.84	50	0.39
					99	0.77	50	0.37
					99	0.8	50	0.39
					99	0.85	50	0.45
					99	0.85	50	0.38
					100	0.85	50	0.45
					100	0.82	50	0.41
					100	0.85	50	0.39
					100	0.83	50	0.39
					100	0.82	50	0.38
					100	0.81	50	0.43
					100	0.81	50	0.38
					100	0.77	50	0.37
					100	0.77		
							51	0.36
							51	0.38
							51	0.37
							51	0.43
							51	0.35
							51	0.42
							51	0.39
							51	0.39
							51	0.35
							51	0.38
							51	0.36
							51	0.38
							51	0.37
							51	0.39
							51	0.36
							51	0.37
							52	0.37
							52	0.37
							52	0.37
							52	0.36
							52	0.35
							52	0.39
							52	0.36
							52	0.42
							52	0.42

ST 0.45 0.	I.	İ	I	ı		1	I	I	I
								52	0.39
								52	
								52	0.43
								52	0.46
								52	0.35
								52	0.41
								53	0.42
								53	0.38
S3								53	0.36
S3								53	0.38
								53	0.37
S								53	0.38
								53	0.37
								53	0.38
S									
S									
Signature Sign									
SS 0.39 SS 0.41 SS 0.41 SS 0.41 SS 0.41 SS 0.45 SS									
Signature Sign									
Signature Sign									
S3									
S4 0.39									
S4									
S4									
54 0.39 54 0.39 54 0.39 54 0.42 54 0.36 54 0.4 54 0.39 54 0.39 54 0.39 54 0.39 54 0.39 54 0.4 54 0.4 54 0.4 54 0.4 54 0.37 55 0.39 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 <t< td=""><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></t<>									
54 0.39 54 0.39 54 0.36 54 0.36 54 0.36 54 0.39 54 0.39 54 0.43 54 0.43 54 0.43 54 0.43 54 0.44 54 0.4 55 0.39 55 0.4 55 0.4 55 0.4 55 0.39 55 0.39 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.3 55 0.3									
54 0.39 54 0.42 54 0.42 54 0.36 54 0.4 54 0.39 54 0.43 54 0.43 54 0.39 54 0.39 54 0.37 55 0.39 55 0.4 55 0.4 55 0.4 55 0.39 55 0.39 55 0.39 55 0.39 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4									
54 0.42 54 0.36 54 0.4 54 0.39 54 0.39 54 0.33 54 0.39 54 0.39 54 0.39 54 0.37 55 0.37 55 0.39 55 0.4 55 0.4 55 0.38 55 0.38 55 0.39 55 0.39 55 0.39 55 0.39 55 0.39 55 0.39 55 0.39 55 0.39 55 0.39 55 0.39 55 0.39 55 0.39 55 0.30 55 0.31 55 0.32 55 0.33 55 0.33 55 0.33 55 0.33 55 0.35<									
54 0.36 54 0.4 54 0.39 54 0.38 54 0.38 54 0.39 54 0.39 54 0.39 54 0.37 55 0.39 55 0.39 55 0.4 55 0.4 55 0.38 55 0.39 55 0.39 55 0.39 55 0.39 55 0.39 55 0.39 55 0.41 55 0.41 55 0.43 55 0.43 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>									
54 0.4 54 0.39 54 0.39 54 0.38 54 0.39 54 0.39 54 0.37 55 0.39 55 0.41 55 0.41 55 0.38 55 0.39 55 0.39 55 0.41 55 0.39 55 0.39 55 0.39 55 0.39 55 0.39 55 0.39 55 0.39 55 0.39 55 0.39 55 0.39 55 0.39 55 0.39 55 0.39 55 0.39 55 0.39 55 0.39 55 0.39 55 0.39 55 0.41 55 0.43 55 0.43 55 0.4									
54 0.39 54 0.38 54 0.43 54 0.43 54 0.39 54 0.39 54 0.4 54 0.37 55 0.39 55 0.4 55 0.4 55 0.4 55 0.38 55 0.39 55 0.39 55 0.39 55 0.41 55 0.41 55 0.41 55 0.43 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4									
54 0.38 54 0.43 54 0.43 54 0.39 54 0.39 54 0.4 55 0.37 55 0.4 55 0.4 55 0.4 55 0.38 55 0.39 55 0.39 55 0.39 55 0.39 55 0.39 55 0.39 55 0.39 55 0.39 55 0.41 55 0.41 55 0.43 55 0.43 55 0.37 55 0.38 55 0.38 55 0.38 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.37 55 0.38 55 0.38 55 0.38									
54 0.43 54 0.39 54 0.39 54 0.4 54 0.4 55 0.37 55 0.39 55 0.4 55 0.4 55 0.4 55 0.39 55 0.39 55 0.39 55 0.39 55 0.41 55 0.41 55 0.43 55 0.43 55 0.4 55 0.4 55 0.4 55 0.37 55 0.37 55 0.35 55 0.35 55 0.4 55 0.35 55 0.41 56 0.4 56 0.4 56 0.36									
54 0.39 54 0.39 54 0.4 54 0.4 54 0.37 55 0.39 55 0.39 55 0.4 55 0.41 55 0.38 55 0.39 55 0.39 55 0.39 55 0.41 55 0.4 55 0.4 55 0.4 55 0.4 55 0.37 55 0.4 55 0.37 55 0.38 55 0.38 55 0.38 55 0.38 55 0.38 55 0.35 55 0.35 55 0.41 56 0.4 56 0.41 56 0.36									
54 0.39 54 0.4 54 0.37 55 0.39 55 0.4 55 0.4 55 0.4 55 0.38 55 0.38 55 0.39 55 0.39 55 0.39 55 0.41 55 0.41 55 0.4 55 0.4 55 0.4 55 0.4 55 0.36 55 0.36									
54 0.4 54 0.37 55 0.39 55 0.4 55 0.41 55 0.4 55 0.4 55 0.38 55 0.39 55 0.39 55 0.39 55 0.41 55 0.4 55 0.4 55 0.4 55 0.3 55 0.4 55 0.3 55 0.3 55 0.3 55 0.3 55 0.3 55 0.3 55 0.3 55 0.3 55 0.4 55 0.3 55 0.4 55 0.3 55 0.4 55 0.3 55 0.3 55 0.4 56 0.4 56 0.4 56 0.36 <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>									
54 0.37 55 0.39 55 0.4 55 0.41 55 0.41 55 0.4 55 0.38 55 0.39 55 0.39 55 0.41 55 0.41 55 0.4 55 0.4 55 0.4 55 0.4 55 0.4 55 0.37 55 0.37 55 0.38 55 0.35 55 0.35 55 0.41 56 0.4 56 0.41 56 0.36									
55 0.39 55 0.4 55 0.41 55 0.4 55 0.38 55 0.39 55 0.39 55 0.39 55 0.41 55 0.41 55 0.4 55 0.4 55 0.4 55 0.37 55 0.38 55 0.38 55 0.35 55 0.41 56 0.41 56 0.41 56 0.36									
55 0.4 55 0.41 55 0.4 55 0.38 55 0.39 55 0.39 55 0.39 55 0.41 55 0.41 55 0.43 55 0.4 55 0.37 55 0.37 55 0.38 55 0.38 55 0.35 55 0.41 56 0.41 56 0.41 56 0.36									
55 0.41 55 0.4 55 0.38 55 0.39 55 0.39 55 0.39 55 0.41 55 0.4 55 0.43 55 0.4 55 0.37 55 0.37 55 0.38 55 0.35 55 0.41 56 0.41 56 0.41 56 0.36								55	0.39
55 0.4 55 0.38 55 0.39 55 0.39 55 0.41 55 0.41 55 0.43 55 0.43 55 0.4 55 0.37 55 0.4 55 0.38 55 0.38 55 0.35 55 0.41 56 0.41 56 0.41 56 0.36								55	0.4
55 0.38 55 0.39 55 0.39 55 0.39 55 0.41 55 0.4 55 0.43 55 0.4 55 0.4 55 0.37 55 0.4 55 0.38 55 0.38 55 0.35 55 0.41 56 0.41 56 0.41 56 0.36								55	0.41
55 0.39 55 0.39 55 0.41 55 0.4 55 0.43 55 0.4 55 0.4 55 0.37 55 0.4 55 0.38 55 0.35 55 0.35 55 0.41 56 0.41 56 0.36								55	0.4
55 0.39 55 0.41 55 0.4 55 0.43 55 0.4 55 0.37 55 0.37 55 0.38 55 0.38 55 0.35 55 0.41 56 0.41 56 0.36								55	0.38
55 0.41 55 0.4 55 0.43 55 0.4 55 0.37 55 0.37 55 0.38 55 0.38 55 0.35 55 0.41 56 0.41 56 0.36								55	0.39
55 0.4 55 0.43 55 0.4 55 0.37 55 0.4 55 0.38 55 0.35 55 0.35 55 0.41 56 0.41 56 0.36								55	0.39
55 0.43 55 0.4 55 0.37 55 0.4 55 0.4 55 0.38 55 0.35 55 0.41 56 0.4 56 0.36								55	0.41
55 0.4 55 0.37 55 0.4 55 0.38 55 0.35 55 0.35 55 0.41 56 0.41 56 0.36								55	0.4
55 0.37 55 0.4 55 0.38 55 0.35 55 0.35 55 0.41 56 0.4 56 0.36								55	0.43
55 0.4 55 0.38 55 0.35 55 0.41 56 0.4 56 0.41 56 0.36								55	0.4
55 0.38 55 0.35 55 0.41 56 0.4 56 0.41 56 0.36								55	0.37
55 0.38 55 0.35 55 0.41 56 0.4 56 0.41 56 0.36								55	0.4
55 0.35 55 0.41 56 0.4 56 0.41 56 0.36									
55 0.41 56 0.4 56 0.41 56 0.36									
56 0.4 56 0.41 56 0.36									
56 0.41 56 0.36									
56 0.36									
								50	V.4

I.	I.	I	I	1	1	I	ı	I
							56	0.4
							56	0.39
							56	0.39
							56	0.37
							56	0.38
							56	0.4
							56	0.38
							56	0.4
							56	0.44
							56	0.42
							56	0.4
							56	0.4
							57	0.41
							57	0.4
							57	0.39
							57	0.41
							57	0.41
							57	0.48
							57	0.4
							57	0.38
							57	0.39
							57	0.46
							57	0.42
							57	0.39
							57	0.4
							57	0.36
							57	0.36
							57	0.39
							58	0.45
							58	0.4
							58	0.39
							58	0.39
							58	0.37
							58	0.41
							58	0.38
							58	0.39
							58	0.39
							58	0.4
							58	0.35
							58	0.43
							58	0.34
							58	0.38
					 		58	0.4
							58	0.39
							59	0.42
							59	0.42
							59	0.4
							59	0.36
							59	0.39
							59	0.39
							59	0.39
							59	0.38
							59	0.44
							59	0.38
							59	
								0.36
							59	0.42
							59	0.38
							59	0.39
							59	0.38

I	I	I	I	1	1	I	ı	I
							59	0.39
							60	0.36
							60	0.4
							60	0.38
							60	0.39
							60	0.4
							60	0.42
							60	0.38
							60	0.39
							60	0.4
							60	0.43
							60	0.4
							60	0.4
							60	0.41
							60	0.38
							60	0.39
							60	0.4
							61	0.41
							61	0.4
							61	0.38
							61	0.39
							61	0.38
							61	0.36
							61	0.4
								0.44
							61	
							61	0.37
							61	0.39
							61	0.39
							61	0.39
							61	0.41
							61	0.38
							61	0.39
							61	0.38
							62	0.35
							62	0.46
							62	0.35
							62	0.37
							62	0.41
							62	0.44
							62	0.4
							62	0.44
							62	0.4
							62	0.39
							62	0.45
							62	0.41
							62	0.42
							62	0.43
							62	0.43
							62	0.41
							63	0.44
							63	0.39
							63	0.4
							63	0.45
							63	0.43
							63	0.41
							63	0.41
							63	0.39
							63	0.44
							63	0.37

I.	I	I	I	1		1	I	ı	I
								63	0.42
								63	0.38
								63	0.5
								63	0.35
								63	0.39
								63	0.42
								64	0.4
								64	0.41
								64	0.37
								64	0.39
								64	0.41
								64	0.4
								64	0.38
								64	0.4
								64	0.4
								64	0.4
								64	0.38
								64	0.46
								64	0.41
								64	0.37
								64	0.43
								64	0.4
								65	0.43
								65	0.4
								65	0.43
								65	0.42
								65	0.42
								65	0.44
								65	0.47
								65	0.43
								65	0.39
								65	0.4
								65	0.41
								65	0.41
								65	0.42
								65	0.41
								65	0.39
								65	0.41
								66	0.38
								66	0.38
								66	0.44
								66	0.41
								66	0.41
								66	0.44
								66	0.4
								66	0.4
								66	0.42
								66	0.41
								66	0.4
								66	0.43
								66	0.43
								66	0.41
									0.41
								66	
								66	0.4
								67	0.41
								67	0.42
								67	0.4
								67	0.4
								67	0.41
					-				-

1	I	I	ı		1	I	ı	I
							67	0.41
							67	0.42
							67	0.4
							67	0.41
							67	0.42
							67	0.39
							67	0.43
							67	0.4
							67	0.47
							67	0.41
							67	0.42
							68	0.42
							68	0.41
							68	0.43
							68	0.41
							68	0.43
							68	0.41
							68	0.41
							68	0.41
							68	0.4
							68	0.42
							68	0.43
							68	0.39
							68	0.43
							68	0.42
							68	0.42
							68	0.36
							69	0.41
							69	0.45
							69	0.41
							69	0.4
							69	0.38
							69	0.43
							69	0.35
							69	0.44
							69	0.56
							69	0.41
							69	0.41
							69	0.41
							69	0.4
							69	0.41
							69	0.44
							69	0.39
							70	0.4
							70	0.42
							70	0.38
							70	0.43
							70	0.43
							70	0.48
							70	0.47
							70	0.43
							70	0.4
							70	0.43
							70	0.4
							70	0.43
							70	0.43
							70	0.44
							70	0.4
							70	0.4

1 1		I	l 1	Ì	l	l	l
						71	0.4
						71	0.45
						71	0.43
						71	0.43
						71	0.44
						71	0.43
						71	0.4
						71	0.41
						71	0.4
						71	0.42
						71	0.41
						71	0.4
						71	0.47
						71	0.39
						71	0.41
						71	0.41
						72	0.4
						72	0.42
						72	0.42
						72	0.38
						72	0.43
						72	0.42
						72	0.42
						72	0.4
						72	0.43
						72	0.42
						72	0.43
						72	0.38
						72	0.37
						72	0.43
						72	0.52
						72	0.41
						73	0.41
						73	0.39
						73	0.45
						73	0.41
						73	0.41
						73	0.39
						73	0.4
						73	0.41
						73	0.4
						73	0.49
						73	0.47
						73	0.41
						73	0.4
						73	0.42
						73	0.49
						73	0.42
						74	0.43
						74	0.38
						74	0.43
						74	0.42
						74	0.4
						74	0.43
						74	0.41
						74	0.49
						74	0.49
						74 74	0.43

	I	l	I	I		1	I	ı	I
								74	0.46
								74	0.39
								75	
								75	0.41
Section Sect								75	0.4
								75	0.39
								75	0.39
								75	0.47
								75	0.39
								75	0.42
								75	0.42
								75	0.42
75 0.39 75 0.37 75 0.37 75 0.37 75 0.37 75 0.37 75 0.42 75 0.44 76 0.43 76 0.44 77 0.44 77 0.44 77 0.44 77 0.44 77 0.44 77 0.44 77 0.44 77 0.44 77 0.44 77 0.44 77 0.44 77 0.44 77 0.42 77 0.44 77 0.45 77									
1									
1									
76									
10									
1									
76									
1									
76 0.42 0.44 0.									
76									
76 0.38 76 0.42 76 0.42 76 0.44 76 0.39 76 0.41 76 0.41 76 0.41 76 0.42 76 0.42 77 0.41 77 0.41 77 0.44 77 0.42 77 0.42 77 0.42 77 0.42 77 0.43 77 0.44 77 0.42 77 0.42 77 0.43 77 0.44 77 0.44 77 0.45 77 0.45 77 0.41 77 0.42 77 0.42 77 0.43 77 0.44 77 0.42 77 0.43 77 0.44 77 0.42 77 0.43 77 0.44 77 0.42 77 0.43 77 0.44 77 0.42 77 0.43 77 0.45 77 0.42 77 0.42 77 0.43 77 0.44 77 0.42 77 0.43 77 0.45 77 0.42 77 0.43 77 0.42 77 0.43 77 0.44 77 0.42 77 0.42 77 0.43 77 0.44 77 0.42 77 0.43 77 0.44 77 0.42 77 0.45 77 0.45 77 0.46 78 0.4									
10									
10									
1									
76 0.41 76 0.41 76 0.41 76 0.42 76 0.42 77 0.41 77 0.41 77 0.44 77 0.44 77 0.44 77 0.42 77 0.43 77 0.42 77 0.42 77 0.42 77 0.42 77 0.43 77 0.43 77 0.43 77 0.43 77 0.43 77 0.42 77 0.43 77 0.42 77 0.43 77 0.42 77 0.43 77 0.42 77 0.43 77 0.42 77 0.43 77 0.42 77 0.43 77 0.42 77 0.43 77 0.									
76 0.41 76 0.46 76 0.46 76 0.42 77 0.41 77 0.41 77 0.44 77 0.44 77 0.44 77 0.42 77 0.41 77 0.41 77 0.42 77 0.42 77 0.42 77 0.42 77 0.41 77 0.42 77 0.43 77 0.44 77 0.42 77 0.42 77 0.43 77 0.42 77 0.42 77 0.43 77 0.42 77 0.42 77 0.43 77 0.42 77 0.43 77 0.42 77 0.43 77 0.42 77 0.43 77 0.									
76 0.46 76 0.39 76 0.42 77 0.41 77 0.41 77 0.44 77 0.44 77 0.44 77 0.41 77 0.43 77 0.42 77 0.41 77 0.42 77 0.42 77 0.43 77 0.42 77 0.43 77 0.43 77 0.42 77 0.42 77 0.42 77 0.42 77 0.42 77 0.42 77 0.42 77 0.42 77 0.42 77 0.42 77 0.42 77 0.42 77 0.42 77 0.42 77 0.42 77 0.42 77 0.42 77 0.									
76 0.39 76 0.42 77 0.41 77 0.41 77 0.44 77 0.44 77 0.44 77 0.41 77 0.42 77 0.43 77 0.42 77 0.42 77 0.42 77 0.43 77 0.42 77 0.43 77 0.43 77 0.43 77 0.43 77 0.43 77 0.42 77 0.42 77 0.42 77 0.42 77 0.42 77 0.42 77 0.42 77 0.42 77 0.43 77 0.42 77 0.42 77 0.43 77 0.42 77 0.43 77 0.42 77 0.									
76 0.42 77 0.41 77 0.41 77 0.44 77 0.44 77 0.44 77 0.42 77 0.41 77 0.42 77 0.42 77 0.42 77 0.42 77 0.42 77 0.42 77 0.42 77 0.42 77 0.42 77 0.42 77 0.42 77 0.42 77 0.42 77 0.42 77 0.42 77 0.42 77 0.42 78 0.4									
77 0.41 77 0.41 77 0.44 77 0.44 77 0.42 77 0.41 77 0.41 77 0.43 77 0.42 77 0.41 77 0.42 77 0.42 77 0.43 77 0.43 77 0.43 77 0.42 77 0.42 77 0.38 77 0.38 77 0.38 77 0.42 78 0.4 78 0.4 78 0.4 78 0.4 78 0.4 78 0.4 78 0.4 78 0.4 78 0.4 78 0.4 78 0.4 78 0.43									
77 0.41 77 0.44 77 0.44 77 0.42 77 0.41 77 0.42 77 0.41 77 0.42 77 0.42 77 0.41 77 0.42 77 0.42 77 0.43 77 0.42 77 0.43 77 0.43 77 0.42 77 0.43 77 0.42 77 0.42 77 0.42 77 0.42 77 0.38 77 0.38 77 0.42 78 0.4 78 0.4									
77 0.44 77 0.44 77 0.42 77 0.41 77 0.42 77 0.43 77 0.42 77 0.41 77 0.42 77 0.41 77 0.42 77 0.42 77 0.43 77 0.43 77 0.43 77 0.43 77 0.43 77 0.43 77 0.45 77 0.40 77 0.42 77 0.42 77 0.42 77 0.38 77 0.38 77 0.38 77 0.38 77 0.42 78 0.4 78 0.4 78 0.4								77	
77 0.44 77 0.42 77 0.41 77 0.43 77 0.42 77 0.41 77 0.42 77 0.43 77 0.43 77 0.42 77 0.42 77 0.38 77 0.38 77 0.42 77 0.42 77 0.42 77 0.42 77 0.42 77 0.42 78 0.4 78 0.4 78 0.4 78 0.4 78 0.4 78 0.4 78 0.4 78 0.4 78 0.4 78 0.4 78 0.4 78 0.4 79 0.43								77	
77 0.42 77 0.41 77 0.43 77 0.42 77 0.42 77 0.42 77 0.42 77 0.42 77 0.43 77 0.42 77 0.43 77 0.42 77 0.42 77 0.42 77 0.38 77 0.38 77 0.38 77 0.42 78 0.4 78 0.4 78 0.4								77	0.44
77 0.41 77 0.43 77 0.42 77 0.41 77 0.42 77 0.42 77 0.43 77 0.43 77 0.43 77 0.42 77 0.38 77 0.38 77 0.38 77 0.42 78 0.4 78 0.4								77	0.44
77 0.43 77 0.42 77 0.41 77 0.42 77 0.43 77 0.43 77 0.42 77 0.38 77 0.38 77 0.42 78 0.4 78 0.4 78 0.4 78 0.4 78 0.4 78 0.4 78 0.4 78 0.4 78 0.4 78 0.4 78 0.4 78 0.4 78 0.4 78 0.4 78 0.4 78 0.43								77	0.42
77 0.42 77 0.41 77 0.42 77 0.43 77 0.43 77 0.44 77 0.42 77 0.42 77 0.38 77 0.38 77 0.38 77 0.42 78 0.4 78 0.4 78 0.4 78 0.4								77	0.41
77 0.41 77 0.42 77 0.43 77 0.44 77 0.4 77 0.4 77 0.4 77 0.4 77 0.42 77 0.38 77 0.38 77 0.38 77 0.42 78 0.4 78 0.4 78 0.4 78 0.4								77	0.43
77 0.42 77 0.43 77 0.44 77 0.42 77 0.42 77 0.42 77 0.38 77 0.38 77 0.38 77 0.42 78 0.4 78 0.4 78 0.4 78 0.4 78 0.4 78 0.4								77	0.42
77 0.43 77 0.4 77 0.4 77 0.42 77 0.38 77 0.38 77 0.38 77 0.42 78 0.4 78 0.4 78 0.4 78 0.4 78 0.4 78 0.4 78 0.4								77	0.41
77 0.4 77 0.42 77 0.38 77 0.38 77 0.38 77 0.42 78 0.4 78 0.4 78 0.4 78 0.4 78 0.4 78 0.4 78 0.4 78 0.4								77	0.42
77 0.42 77 0.38 77 0.38 77 0.42 77 0.42 78 0.4 78 0.4 78 0.4 78 0.4 78 0.4 78 0.4 78 0.4 78 0.4								77	0.43
77 0.38 77 0.38 77 0.38 77 0.42 78 0.4 78 0.4 78 0.4 78 0.41 78 0.4 78 0.4 78 0.4								77	0.4
77 0.38 77 0.42 78 0.4 78 0.4 78 0.4 78 0.4 78 0.41 78 0.41 78 0.4 78 0.4								77	0.42
77 0.38 77 0.42 78 0.4 78 0.4 78 0.4 78 0.4 78 0.41 78 0.41 78 0.4 78 0.4									
77 0.42 78 0.4 78 0.4 78 0.4 78 0.4 78 0.4 78 0.41 78 0.4 78 0.4 78 0.4									
78 0.4 78 0.4 78 0.4 78 0.41 78 0.41 78 0.41 78 0.4 78 0.4									
78 0.4 78 0.41 78 0.41 78 0.4 78 0.43									
78 0.41 78 0.4 78 0.4 78 0.43									
78 0.4 78 0.43									
78 0.43									
/8 0.39									
								/0	0.39

1 1		l		l I		İ	I	l	l
								78	0.43
								78	0.42
								78	0.41
								78	0.44
								78	0.42
								78	0.4
								78	0.44
								78	0.46
								78	0.44
								78	0.45
								79	0.42
								79	0.43
								79	0.43
								79	0.42
								79	0.42
								79	0.44
								79	0.42
								79	0.42
								79	0.41
								79	0.42
								79	0.44
								79	0.42
								79	0.43
								79	0.41
								79	0.46
								79	0.42
								80	0.39
								80	0.4
								80	0.45
								80	0.43
								80	0.39
								80	0.39
								80	0.42
								80	0.41
								80	0.41
								80	0.41
								80	0.39
								80	0.43
									0.43
								80	
								80	0.43
								80	0.41
								80	0.41
								81	0.42
								81	0.38
								81	0.42
								81	0.43
								81	0.44
								81	0.43
					-			81	0.43
								81	0.43
								81	0.42
								81	0.41
								81	0.42
								81	0.44
								81	0.45
								81	0.43
									0.43
								81	
								81 81 82	0.42

l I		l I	l I	ĺ		l	l
						82	0.44
						82	0.42
						82	0.4
						82	0.43
						82	0.45
						82	0.43
						82	0.42
						82	0.43
						82	0.41
						82	0.41
						82	0.39
						82	0.42
						82	0.43
						82	0.41
						82	0.42
						83	0.42
						83	0.42
						83	0.43
						83	0.4
						83	0.4
						83	0.39
						83	0.39
						83	0.47
						83	0.42
						83	0.43
						83	0.5
						83	0.44
						83	0.43
						83	0.44
						83	0.43
						83	0.45
						84	0.45
						84	0.45
						84	0.45
						84	0.4
						84	0.41
						84	0.45
						84	0.43
						84	0.43
						84	0.41
						84	0.42
						84	0.41
						84	0.43
						84	0.44
						84	0.45
						84	0.42
						84	0.42
						85	0.42
						85	0.43
						85	0.43
						85	0.43
						85	0.41
						85	0.43
						85	0.42
						85	0.42
						85	0.44
l l						85	0.43
						85 85	0.41

1	l I	ı		1		ı	l	l
							85	0.44
							85	0.49
							85	0.39
							85	0.42
							86	0.42
							86	0.42
							86	0.42
							86	0.41
							86	0.42
							86	0.42
							86	0.42
							86	0.44
							86	0.43
							86	0.4
							86	0.43
							86	0.43
							86	0.48
							86	0.41
					 		86	0.41
							86	0.41
							87	0.42
							87	0.43
							87	0.42
							87	0.4
							87	0.46
							87	0.4
							87	0.42
							87	0.43
							87	0.45
							87	0.44
							87	0.41
							87	0.4
							87	0.41
							87	0.45
							87	0.45
							87	0.4
							88	0.43
							88	0.43
							88	0.42
							88	0.44
							88	0.41
							88	0.44
							88	0.45
							88	0.43
							88	0.42
							88	0.47
							88	0.47
							88	0.43
							88	0.45
							88	0.42
							88	0.53
							88	0.45
							89	0.43
							89	0.41
							89	0.45
							89	0.45
							89	0.42
							89	0.45
								0.45 0.53

l I		l	I	l 1	1	I	I	l
							89	0.67
							89	0.53
							89	0.44
							89	0.46
							89	0.41
							89	0.45
							89	0.41
							89	0.42
							89	0.43
							90	0.44
							90	0.44
							90	0.44
							90	0.42
							90	0.43
							90	0.46
							90	0.44
							90	0.51
							90	0.43
							90	
								0.45
							90	0.44
							90	0.42
							90	0.43
							90	0.42
							90	0.43
							90	0.44
							91	0.43
							91	0.43
							91	0.42
							91	0.43
							91	0.52
							91	0.41
							91	0.44
							91	0.44
							91	0.44
							91	0.42
							91	0.43
							91	0.42
							91	0.42
							91	0.42
							91	0.52
							91	0.41
							92	0.42
							92	0.42
							92	0.43
							92	0.43
							92	0.43
							92	0.42
							92	0.42
							92	0.43
							92	0.44
							92	0.42
							92	0.41
							92	0.44
							92	0.42
							92	0.42
							92	0.42
							92	0.42
							1 47	1.11.47
							93	0.42

1	ı		l I	l	I	I	I
						93	0.43
						93	0.45
						93	0.43
						93	0.44
						93	0.44
						93	0.42
						93	0.43
						93	0.44
						93	0.42
						93	0.43
						93	0.44
						93	0.42
						93	0.44
						93	0.43
						94	0.46
						94	0.46
						94	0.45
						94	0.42
						94	0.44
						94	0.43
						94	0.44
						94	0.44
						94	0.44
						94	0.44
						94	0.44
						94	0.47
						94	0.43
						94	0.43
						94	0.42
						94	0.43
						95	0.55
						95	0.43
						95	0.46
						95	0.48
							0.45
						95 95	
							0.43
						95	0.43
						95	0.48
						95	0.44
						95	0.43
						95	0.44
						95	0.49
						95	0.44
						95	0.46
						95	0.44
						95	0.42
						96	0.42
						96	0.45
						96	0.44
						96	0.43
						96	0.44
						96	0.47
						96	0.47
						96	
							0.44
						96	0.45
1						96	0.43
	-				1	1.00	1.0.42
						96	0.42
						96 96 96	0.43

1 1		l	l I	l 1	1	l	1	1
							96	0.45
							96	0.54
							96	0.43
							97	0.43
							97	0.43
							97	0.43
							97	0.43
							97	0.46
							97	0.44
							97	0.44
							97	0.46
							97	0.44
							97	0.42
							97	0.43
							97	0.54
							97	0.54
							97	0.54
							97	0.44
							97	0.44
							98	0.43
							98	0.45
							98	0.44
							98	0.45
							98	0.45
							98	0.43
							98	0.46
							98	0.44
							98	0.55
							98	0.43
							98	0.44
							98	0.43
							98	0.43
							98	0.55
							98	0.44
							98	0.43
							99	0.44
							99	0.55
							99	0.47
							99	0.56
							99	0.44
							99	0.43
							99	0.44
							99	0.51
							99	0.45
							99	0.45
							99	0.46
							99	0.44
							99	0.46
							99	0.43
							99	0.46
							99	0.45
							100	0.44
							100	0.45
							100	0.44
							100	0.44
							100	0.44
							100	0.56
1								0.50
							100	0.44

1	1 '	1		1	l	1	
						100	0.46

CompTorrent

	ı								
1	0.03	1	0.5	1	0.04	1	1	1	0.33
2	0.06	1	1	1	0.04	1	0.33	1	1
3	0.09	2	1	1	0.04	1	0.5	1	0.33
4	0.11	2	0.67	1	0.03	1	0.33	1	0.33
5	0.17	3	1	2	0.15	1	0.5	1	0.33
6	0.17	3	0.75	2	0.13	1	0.33	1	0.5
7	0.23	4	1	2	0.13	1	0.33	1	0.14
8	0.24	4	1	2	0.18	1	0.25	1	0.25
9	0.26	5	0.83	3	0.5	2	0.5	1	0.25
10	0.33	5	1	3	0.5	2	0.5	1	0.33
11	0.37	6	1	3	0.43	2	0.4	1	0.25
12	0.38	6	1	3	0.21	2	0.4	1	0.25
13	0.42	7	1	4	0.44	2	0.5	1	0.14
14	0.47	7	1	4	0.67	2	0.33	1	0.25
15	0.43	8	0.3	4	0.67	2	0.33	1	0.33
16	0.46	8	1	4	0.57	2	0.5	1	0.14
17	0.04	9	1	5	0.63	3	0.6	2	0.67
18	0.51	9	1	5	0.56	3	0.5	2	0.25
19	0.53	10	0.67	5	0.71	3	0.6	2	0.25
20	0.63	10	0.91	5	0.63	3	0.5	2	0.25
21	0.24	11	0.79	6	0.67	3	0.5	2	0.29
22	0.71	11	1	6	0.75	3	0.38	2	0.5
23	0.66	12	1	6	0.67	3	0.43	2	0.25
24	0.67	12	0.63	6	0.67	3	0.6	2	0.22
25	0.78	13	1	7	0.7	4	0.57	2	0.67
26	0.81	13	0.81	7	0.7	4	0.57	2	0.33
27	0.79	14	1	7	0.58	4	0.44	2	0.29
28	0.93	14	0.64	7	0.7	4	0.57	2	0.33
29	0.73	15	0.94	8	0.73	4	0.57	2	0.33
30	0.83	15	1	8	0.8	4	0.5	2	0.33
31	0.43	16	0.8	8	0.73	4	0.57	2	0.33
32	0.91	16	1	8	0.8	4	0.57	2	0.5
33	1.1	17	0.81	9	0.75	5	0.63	3	0.6
34	0.83	17	1	9	0.69	5	0.63	3	0.5
35	0.88	18	0.82	9	0.69	5	0.63	3	0.75
36	1.06	18	1	9	0.69	5	0.71	3	0.5
37	1.06	19	0.68	10	0.83	5	0.63	3	0.75
38	1.12	19	0.95	10	0.83	5	0.71	3	0.5
39	1.08	20	1	10	0.83	5	0.56	3	0.6
40	1.25	20	0.8	10	0.83	5	0.71	3	0.6
41	1	21	1	11	0.79	6	0.75	3	0.33
42	1.2	21	0.84	11	0.79	6	0.75	3	0.43
43	1.05	22	0.96	11	0.79	6	0.75	3	0.27
44	1.29	22	0.81	11	0.79	6	0.75	3	0.33
45	1.5	23	1	12	0.8	6	0.67	3	0.3
46	1.59	23	0.77	12	0.86	6	0.67	3	0.38
47	1.57	24	1	12	0.86	6	0.67	3	0.25
48	1.41	24	1	12	0.8	6	0.75	3	0.38
49	1.4	25	0.86	13	0.81	7	0.78	4	0.67
50	1.22	25	0.96	13	0.81	7	0.78	4	0.5
51	1.31	26	0.87	13	0.81	7	0.78	4	0.36
52	1.27	26	1	13	0.81	7	0.7	4	0.44
53	1.33	27	0.84	14	0.88	7	0.7	4	0.36
54	1.54	27	1	14	0.88	7	0.7	4	0.57
		l							

55	1.38	28	0.88	14	0.88	7	0.78	4	0.4
56	1.51		0.88	14	0.88	7	0.78	4	0.4
		28							
57	1.43	29	0.88	15	0.79	8	0.89	4	0.67
58	1.93	29	1	15	0.83	8	0.89	4	0.67
59	1.44	30	0.81	15	0.75	8	0.89	4	0.8
60	1.46	30	1	15	0.88	8	0.73	4	0.57
61	1.69	31	0.97	16	0.89	8	0.89	4	0.67
62	1.59	31	0.86	16	0.89	8	0.73	4	0.57
63	0.74	32	1	16	0.84	8	0.73	4	0.57
64	1.56	32	0.89	16	0.84	8	0.8	4	0.5
65	1.86	33	0.97	17	0.85	9	0.82	5	0.71
66	1.89	33	0.87	17	0.85	9	0.82	5	0.56
67	1.91	34	1	17	0.89	9	0.82	5	0.63
68	1.66	34	0.92	17	0.85	9	0.9	5	0.5
69	1.73	35	1	18	0.82	9	0.82	5	0.5
70	1.67	35	0.9	18	0.9	9	0.82	5	0.63
71	1.73	36	0.97	18	0.9	9	0.75	5	0.45
72	1.89	36	0.84	18	0.86	9	0.82	5	0.45
73	1.97	37	1	19	0.83	10	0.91	5	0.45
74	2.06	37	0.9	19	0.9	10	0.91	5	0.45
75	2.14	38	1	19	0.86	10	0.83	5	0.71
76		38	0.83	19	0.86	10	0.83	5	0.45
77	2.17	39						5	
	1.88		1	20	0.91	10	0.83		0.71
78	2.23	39	1	20	0.91	10	0.83	5	0.45
79	2.19	40	0.89	20	0.91	10	0.77	5	0.63
80	2	40	1	20	0.87	10	0.83	5	0.71
81	2.61	41	0.93	21	0.88	11	0.85	6	0.67
82	2.28	41	1	21	0.88	11	0.92	6	0.67
83	2.08	42	0.91	21	0.84	11	0.92	6	0.86
84	2.21	42	0.98	21	0.88	11	0.85	6	1
85	2.13	43	0.93	22	0.92	11	0.85	6	0.86
86	2.21	43	1	22	0.88	11	0.85	6	0.86
87	2.64	44	0.92	22	0.92	11	0.85	6	0.67
88	2.44	44	1	22	0.88	11	0.79	6	0.86
89	2.23	45	0.9	23	0.88	12	0.86	6	0.86
90	2.57	45	0.98	23	0.88	12	0.86	6	0.86
91	2.33	46	0.9	23	0.88	12	0.86	6	0.75
92	2.24	46	1	23	0.88	12	0.92	6	0.75
93	2.38	47	0.94	24	0.92	12	0.8	6	0.67
94	2.69	47	0.98	24	0.89	12	0.8	6	0.75
95	2.38	48	0.91	24	0.89	12	0.86	6	0.75
			1	24	0.89	12		6	0.73
96	2.4	48					0.86		
97	2.43	49	0.84	25	0.89	13	0.93	7	0.78
98	2.65	49	0.98	25	0.89	13	0.87	7	0.88
99	2.91	50	1	25	0.89	13	0.87	7	1
100	2.5	50	0.91	25	0.86	13	0.87	7	0.88
		51	1	26	0.93	13	0.87	7	0.88
		51	0.94	26	0.93	13	0.87	7	0.78
		52	0.98	26	0.93	13	0.87	7	0.78
		52	0.93	26	0.93	13	0.81	7	0.88
-		53	1	27	0.9	14	0.88	7	0.78
		53	0.95	27	0.87	14	0.88	7	0.7
		54	1	27	0.9	14	0.93	7	0.78
		54	0.95	27	0.87	14	0.93	7	0.78
		55	1	28	0.93	14	0.93	7	0.7
		55	0.93	28	0.9	14	0.82	7	0.7
		56	1	28	0.93	14	0.82	7	0.7
	1	1 30		-0	3.73				
		56	0.93	28	0.93	14	0.88	7	0.88

ĺ	57	0.95	29	0.88	15	0.88	8	0.89
	58	1	29	0.88	15	0.94	8	0.8
	58	0.95	29	0.88	15	0.88	8	0.8
	59	1	30	0.91	15	0.88	8	0.8
	59	0.94	30	0.94	15	0.88	8	0.89
	60	0.98	30	0.91	15	0.88	8	0.8
	60	0.94	30	0.91	15	0.88	8	0.89
	61	1	31	0.91	16	0.94	8	0.8
	61	0.92	31	0.91	16	0.89	8	0.8
	62	1	31	0.91	16	0.89	8	0.8
	62	0.93	31	0.94	16	0.94	8	0.8
	63	0.98	32	0.91	16	0.89	8	0.8
	63	0.94	32	0.94	16	0.89	8	0.73
	64	1	32	0.91	16	0.8	8	0.8
	64	0.96	32	0.94	16	0.84	8	0.73
	65	1	33	0.92	17	0.94	9	0.9
	65	0.88	33	0.89	17	0.94	9	0.9
	66	0.99	33	0.92	17	0.89	9	0.82
	66	1	33	0.92	17	0.89	9	0.82
	67	0.94	34	0.92	17	0.89	9	0.82
	67	1	34	0.92	17	0.85	9	0.82
	68	0.94	34	0.92	17	0.94	9	0.82
	68	0.99	34	0.92	17	0.89	9	0.82
	69	0.95	35	0.9	18	0.95	9	0.82
	69	1	35	0.9	18	0.9	9	0.82
	70	0.95	35	0.92	18	0.9	9	0.82
	70	1	35	0.92	18	0.9	9	0.82
	71	0.93	36	0.92	18	0.95	9	0.82
	71	0.99	36	0.95	18	0.86	9	0.9
	72	0.95	36	0.92	18	0.86	9	0.9
	72	1	36	0.9	18	0.82	9	0.82
	73	0.94	37	0.9	19	0.9	10	0.91
	73	0.99	37	0.93	19	0.95	10	0.83
	74	0.95	37	0.93	19	0.9	10	1
	74	1	37	0.93	19	0.95	10	0.91
	75	0.95	38	0.95	19	0.93	10	0.83
	75	1	38	0.95	19	0.9	10	0.83
	76	0.95	38	0.9	19	0.86	10	0.91
	76	0.99	38	0.9	19	0.9	10	0.91
	77	0.95	39	0.93	20	0.91	10	0.91
	77	1	39	0.93	20	0.95	10	0.91
	78	0.95	39	0.91	20	0.95	10	0.83
	78	0.99	39	0.91	20	0.95	10	0.83
	79	0.94	40	0.93	20	0.91	10	0.91
	79	1	40	0.95	20	0.91	10	0.91
	80	0.95	40	0.91	20	0.91	10	0.83
	80	1	40	0.91	20	0.87	10	0.91
	81	0.95	41	0.93	21	0.95	11	0.92
	81	0.99	41	0.91	21	0.95	11	1
	82	0.9	41	0.93	21	0.91	11	0.92
	82	1	41	0.91	21	0.91	11	0.92
	83	1	42	0.95	21	0.91	11	0.92
	83	0.95	42	0.95	21	0.91	11	0.92
	84	1	42	0.89	21	0.91	11	0.85
	84	0.94	42	0.89	21	0.91	11	0.85
	85	0.99	43	0.91	22	0.92	11	0.79
	85	0.96	43	0.93	22	1	11	0.85
	86	1	43	0.93	22	0.92	11	0.92
	86	0.96	43	0.9	22	0.96	11	0.79

	87	1	44	0.96	22	0.92	11	0.85
	87	0.96	44	0.96	22	0.92	11	0.85
	88	1	44	0.92	22	0.92	11	0.85
	88	0.96	44	0.92	22	0.88	11	0.85
	89	0.99	45	0.92	23	0.96	12	0.92
	89	0.95	45	0.94	23	0.96	12	0.92
	90	1	45	0.92	23	0.96	12	0.92
	90	0.95	45	0.94	23	0.92	12	0.92
	91	1	46	0.96	23	0.96	12	0.92
	91	0.96	46	0.96	23	0.96	12	0.86
	92	0.99	46	0.92	23	0.92	12	0.92
	92	0.96	46	0.92	23	0.92	12	0.86
	93	1	47	0.94	24	0.96	12	0.86
	93	0.95	47	0.92	24	0.92	12	0.86
	94	1	47	0.94	24	0.96	12	0.86
	94	0.96	47	0.92	24	0.96	12	0.92
	95	1	48	0.96	24	0.92	12	0.8
	95	0.95	48	0.92	24	0.92	12	0.86
	96	1	48	0.91	24	0.92	12	0.92
	96	0.96	48	0.92	24	0.92	12	0.86
	97	0.99	49	0.92	25	0.96	13	1
	97	0.95	49	0.92	25	0.93	13	0.93
	98	1	49	0.94	25	0.93	13	0.93
	98	0.96	49	0.94	25	0.96	13	0.93
	99	1	50	0.96	25	0.93	13	1
	99	0.97	50	0.96	25	0.96	13	1
	100	1	50	0.93	25	0.89	13	0.87
	100	0.97	50	0.91	25	0.93	13	0.93
			51	0.94	26	0.96	13	0.93
			51	0.94	26	0.96	13	0.93
			51	0.94	26	0.96	13	0.87
			51	0.94	26	0.93	13	0.87
			52	0.95	26	0.96	13	0.87
			52	0.96	26	0.93	13	0.87
			52	0.91	26	0.9	13	0.87
			52	0.93	26	0.93	13	0.87
			53	0.95	27	0.96	14	1
			53	0.95	27	0.93	14	1
			53	0.93	27	0.96	14	0.93
			53	0.93	27	0.96	14	0.93
			54	0.96	27	0.93	14	0.93
			54	0.95	27	0.93	14	1
			54	0.93	27	0.93	14	0.88
			54	0.93	27	0.93	14	0.88
			55	0.95	28	0.93	14	0.93
			55	0.95	28	0.97	14	0.88
			55	0.95	28	0.97	14	0.93
			55	0.95	28	0.97	14	0.88
			56	0.97	28	0.93	14	0.93
			56	0.97	28	0.9	14	0.82
			56	0.93	28	0.93	14	0.88
			56	0.92	28	0.93	14	0.93
			57	0.93	29	0.97	15	0.94
			57	0.95	29	0.97	15	1
			57	0.95	29	0.94	15	0.94
			57	0.95	29	0.97	15	0.94
			58	0.95	29	0.94	15	0.94
			58	0.93	29	0.94	15	1
1	1	I	1	J	1	l **** '	1	1.1

S	1	1	l	50	0.02	20	L 0.01	1.5	0.04
				60	0.94	30	0.97	15	0.94
				60	0.92		0.91	15	0.88
				61	0.94		0.97	16	
				61	0.95	31	0.97	16	0.89
				61	0.94	31	0.97	16	0.94
				61	0.95	31	0.94	16	0.94
				62	0.97	31	0.94	16	0.94
				62	0.94	31	0.94	16	0.94
				62	0.94	31	0.94	16	0.89
				62	0.94	31	0.94	16	1
				63	0.95	32	0.97	16	0.89
				63	0.95	32	0.94	16	0.94
64				63	0.95	32	0.97	16	0.84
64 0.96 32 0.94 16 0.89 64 0.94 32 0.94 16 0.94 64 0.93 32 0.91 16 0.94 65 0.96 33 0.97 17 0.94 65 0.96 33 0.97 17 0.94 65 0.96 33 0.97 17 0.94 65 0.96 33 0.97 17 0.94 66 0.96 33 0.97 17 0.94 66 0.96 33 0.97 17 0.94 66 0.96 33 0.97 17 0.94 66 0.96 33 0.94 17 0.94 66 0.96 33 0.94 17 0.94 66 0.96 33 0.94 17 0.94 66 0.96 33 0.94 17 1 1 66 0.96 33 0.94 17 1 67 0.96 34 0.97 17 0.94 68 0.96 34 0.97 17 0.89 68 0.97 0.96 34 0.97 17 0.89 68 0.97 34 0.94 17 10 0.89 68 0.97 34 0.94 17 0.94 68 0.96 34 0.97 17 0.89 68 0.97 34 0.94 17 0.89 68 0.97 34 0.94 17 0.89 68 0.96 34 0.97 17 0.89 68 0.96 34 0.97 17 0.89 68 0.96 34 0.97 17 0.89 68 0.96 34 0.97 17 0.89 68 0.96 34 0.97 17 0.89 68 0.96 34 0.97 17 0.89 68 0.96 35 0.97 18 0.95 69 0.96 35 0.97 18 0.95 69 0.96 35 0.95 18 0.95 70 0.97 35 0.95 18 0.95 70 0.97 35 0.95 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 72 0.97 36 0.95 18 0.95 18 0.95 72 0.97 36 0.95 18 0.95				63	0.94	32	0.97	16	0.89
64 0.94 32 0.94 16 0.94 0.94 0.94 16 0.94 0.94 0.94 0.94 0.95 0.95 0.96 0.95 0.95 0.95 0.95 0.95 0.95 0.95 0.95				64	0.97	32	0.97	16	0.94
64 0.93 32 0.91 16 0.94 65 0.96 33 0.97 17 0.94 65 0.96 33 0.97 17 0.94 65 0.96 33 0.97 17 0.94 65 0.96 33 0.94 17 0.94 66 0.96 33 0.94 17 0.94 66 0.96 33 0.94 17 0.94 66 0.96 33 0.94 17 0.94 67 0.96 33 0.94 17 1 66 0.96 33 0.94 17 1 67 0.96 34 0.97 17 0.89 67 0.96 34 0.97 17 0.89 67 0.96 34 0.97 17 0.89 67 0.96 34 0.94 17 0.89 <td< td=""><td></td><td></td><td></td><td>64</td><td>0.96</td><td>32</td><td>0.94</td><td>16</td><td>0.89</td></td<>				64	0.96	32	0.94	16	0.89
65 0.96 33 0.97 17 0.94 66 0.96 33 0.97 17 0.94 66 0.94 33 0.97 17 0.94 66 0.96 33 0.97 17 0.94 66 0.96 33 0.97 17 0.94 66 0.96 33 0.94 17 0.94 66 0.96 33 0.94 17 0.94 66 0.96 33 0.94 17 0.94 66 0.96 33 0.94 17 11 11 66 0.96 33 0.94 17 11 67 0.96 34 0.97 17 0.89 67 0.96 34 0.97 17 0.89 68 0.97 34 0.97 17 0.89 68 0.97 34 0.97 17 0.89 68 0.96 34 0.97 17 0.89 68 0.96 34 0.97 17 0.89 68 0.96 34 0.97 17 0.89 68 0.96 34 0.97 17 0.89 68 0.96 34 0.97 17 0.89 68 0.96 34 0.97 17 0.89 68 0.96 34 0.97 17 0.89 68 0.96 34 0.97 17 0.89 68 0.96 34 0.97 17 0.89 68 0.96 34 0.97 17 0.89 68 0.96 34 0.97 17 0.89 68 0.96 34 0.97 17 0.89 69 0.96 35 0.97 18 0.95 69 0.96 35 0.97 18 0.95 69 0.96 35 0.95 18 0.95 69 0.96 35 0.95 18 0.95 70 0.97 35 0.95 18 11 70 0.97 0.97 35 0.95 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.99 0.96 35 0.97 18 0.95 71 0.99 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 72 0.97 36 0.97 18 0.95 73 0.99 18 0.95				64	0.94	32	0.94	16	0.94
65 0.96 33 0.97 17 0.94 65 0.94 33 0.97 17 0.94 66 0.96 33 0.94 17 0.94 66 0.96 33 0.94 17 0.94 66 0.96 33 0.94 17 0.94 66 0.96 33 0.94 17 0.94 66 0.96 33 0.94 17 1.7 66 0.96 33 0.94 17 1.7 67 0.96 34 0.97 17 0.89 67 0.96 34 0.97 17 0.89 67 0.96 34 0.97 17 0.89 67 0.96 34 0.97 17 0.89 67 0.96 34 0.97 17 0.89 68 0.97 34 0.97 17 0.89 68 0.97 34 0.94 17 0.89 68 0.97 34 0.94 17 0.89 68 0.97 34 0.94 17 0.89 68 0.97 34 0.94 17 0.89 68 0.97 34 0.94 17 0.89 68 0.96 34 0.94 17 0.89 68 0.96 34 0.94 17 0.89 68 0.96 34 0.94 17 0.89 68 0.92 34 0.94 17 0.94 68 0.92 34 0.97 17 0.94 68 0.92 34 0.97 18 0.95 69 0.96 35 0.97 18 0.95 69 0.96 35 0.97 18 0.95 69 0.96 35 0.97 18 0.95 69 0.96 35 0.95 18 0.95 70 0.97 35 0.95 18 0.95 70 0.96 35 0.95 18 11 0.95 70 0.96 35 0.95 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 72 0.97 36 0.95 18 0.95 18 0.95 72 0.97 36 0.95 18 0.95 18 0.95 72 0.97 36 0.95 18 0.95 18 0.95 72 0.97 36 0.95 18 0.95 18 0.95 72 0.97 36 0.95 18 0.95 18 0.95				64	0.93	32	0.91	16	0.94
65 0.94 33 0.97 17 0.94 66 0.96 33 0.94 17 0.94 66 0.96 33 0.94 17 0.94 66 0.96 33 0.94 17 0.94 66 0.96 33 0.94 17 19 1 66 0.96 33 0.94 17 19 1 66 0.96 33 0.94 17 11 1 66 0.96 33 0.94 17 1 67 0.96 34 0.97 17 0.89 67 0.96 34 0.97 17 0.89 67 0.96 34 0.97 17 0.89 68 0.97 34 0.97 17 0.89 68 0.97 34 0.97 17 0.89 68 0.97 34 0.94 17 0.94 68 0.96 34 0.97 17 0.89 68 0.96 34 0.97 17 0.89 68 0.96 34 0.97 17 0.89 68 0.96 34 0.97 17 0.89 68 0.96 34 0.97 17 0.89 68 0.96 34 0.97 17 0.94 68 0.96 34 0.97 17 0.94 68 0.92 34 0.97 17 0.94 68 0.92 34 0.97 17 0.94 68 0.92 34 0.97 17 0.94 69 0.96 35 0.97 18 0.95 69 0.96 35 0.97 18 0.95 69 0.96 35 0.97 18 0.95 69 0.96 35 0.95 18 0.95 70 0.97 35 0.95 18 0.95 70 0.97 35 0.95 18 0.95 70 0.97 35 0.95 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.92 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 72 0.97 36 0.95 18 0.95 18 0.95 72 0.97 36 0.95 18 0.95 18 0.95 72 0.97 36 0.95 18 0.95 18 0.95 72 0.97 36 0.95 18 0.95				65	0.96	33	0.97	17	0.94
65 0.93 33 0.94 17 0.94 66 0.96 33 0.94 17 0.89 66 0.96 33 0.94 17 0.94 66 0.96 33 0.94 17 1 66 0.96 33 0.94 17 1 67 0.96 34 0.97 17 0.89 67 0.96 34 0.94 17 0.94 67 0.96 34 0.97 17 0.89 67 0.96 34 0.97 17 0.89 67 0.94 34 0.97 17 0.89 68 0.97 34 0.97 17 0.89 68 0.92 34 0.94 17 0.94 68 0.92 34 0.97 17 0.94 69 0.96 35 0.97 18 0.95 <td< td=""><td></td><td></td><td></td><td>65</td><td>0.96</td><td>33</td><td>0.97</td><td>17</td><td>0.94</td></td<>				65	0.96	33	0.97	17	0.94
66 0.96 33 0.94 17 0.89 66 0.96 33 0.94 17 0.94 66 0.96 33 0.94 17 1 66 0.96 33 0.94 17 1 67 0.96 34 0.97 17 0.89 67 0.96 34 0.97 17 0.89 67 0.96 34 0.97 17 0.89 67 0.96 34 0.97 17 0.89 67 0.96 34 0.97 17 0.89 67 0.94 34 0.97 17 0.89 68 0.97 34 0.94 17 0.94 68 0.99 34 0.94 17 0.94 68 0.92 34 0.94 17 0.94 68 0.92 34 0.97 17 0.89 69 0.96 35 0.97 18 0.95 69 0.96 <td></td> <td></td> <td></td> <td>65</td> <td>0.94</td> <td>33</td> <td>0.97</td> <td>17</td> <td>0.94</td>				65	0.94	33	0.97	17	0.94
66 0.96 33 0.94 17 0.94 66 0.96 33 0.94 17 1 66 0.96 33 0.94 17 1 67 0.96 34 0.97 17 0.89 67 0.96 34 0.97 17 0.89 67 0.96 34 0.97 17 0.89 67 0.96 34 0.97 17 0.89 67 0.94 34 0.97 17 0.89 67 0.94 34 0.97 17 0.89 68 0.97 34 0.94 17 0.94 68 0.96 34 0.94 17 0.94 68 0.96 34 0.94 17 0.94 68 0.92 34 0.97 17 0.94 69 0.96 35 0.97 18 0.95 69 0.96 35 0.97 18 0.95 69 0.96 <td></td> <td></td> <td></td> <td>65</td> <td>0.93</td> <td>33</td> <td>0.94</td> <td>17</td> <td>0.94</td>				65	0.93	33	0.94	17	0.94
66 0.96 33 0.94 17 1 66 0.96 33 0.94 17 1 67 0.96 34 0.97 17 0.89 67 0.96 34 0.97 17 0.89 67 0.96 34 0.97 17 0.89 67 0.94 34 0.97 17 0.89 68 0.97 34 0.94 17 0.89 68 0.97 34 0.94 17 0.89 68 0.96 34 0.94 17 0.89 68 0.96 34 0.94 17 0.94 68 0.92 34 0.94 17 0.94 68 0.92 34 0.94 17 0.89 68 0.94 34 0.94 17 0.89 69 0.96 35 0.97 18 0.95 69 0.96 35 0.97 18 0.95 69 0.93 <td></td> <td></td> <td></td> <td>66</td> <td>0.96</td> <td>33</td> <td>0.94</td> <td>17</td> <td>0.89</td>				66	0.96	33	0.94	17	0.89
66 0.96 33 0.94 17 1 67 0.96 34 0.97 17 0.89 67 0.96 34 0.94 17 0.94 67 0.96 34 0.97 17 0.89 67 0.94 34 0.97 17 0.89 68 0.97 34 0.94 17 0.89 68 0.97 34 0.94 17 0.94 68 0.96 34 0.94 17 0.94 68 0.92 34 0.94 17 0.94 68 0.92 34 0.94 17 0.94 68 0.92 34 0.97 17 0.94 68 0.94 34 0.97 18 0.95 69 0.96 35 0.97 18 0.95 69 0.96 35 0.97 18 0.95 69 0.96 35 0.95 18 1 70 0.97 <td></td> <td></td> <td></td> <td>66</td> <td>0.96</td> <td>33</td> <td>0.94</td> <td>17</td> <td>0.94</td>				66	0.96	33	0.94	17	0.94
67 0.96 34 0.97 17 0.89 67 0.96 34 0.94 17 0.94 67 0.96 34 0.97 17 0.89 67 0.94 34 0.97 17 0.89 68 0.97 34 0.94 17 0.89 68 0.96 34 0.94 17 0.94 68 0.96 34 0.94 17 0.94 68 0.96 34 0.97 17 0.94 68 0.92 34 0.97 17 0.94 68 0.92 34 0.97 17 0.94 68 0.92 34 0.97 17 0.94 68 0.94 34 0.97 17 0.94 69 0.96 35 0.97 18 0.95 69 0.96 35 0.95 18 1 70 0.97 35 0.95 18 1 70 0.93 <td></td> <td></td> <td></td> <td>66</td> <td>0.96</td> <td>33</td> <td>0.94</td> <td>17</td> <td>1</td>				66	0.96	33	0.94	17	1
67 0.96 34 0.94 17 0.94 67 0.96 34 0.97 17 0.89 67 0.94 34 0.97 17 0.89 68 0.97 34 0.94 17 0.89 68 0.96 34 0.94 17 0.94 68 0.92 34 0.97 17 0.94 68 0.94 34 0.97 17 0.94 68 0.94 34 0.97 17 0.94 69 0.96 35 0.97 18 0.95 69 0.96 35 0.97 18 0.95 69 0.96 35 0.97 18 0.95 69 0.96 35 0.95 18 0.95 10 0.96 35 0.95 18 0.95 10 0.96 35 0.95 18 0.95 10 0.96 35 0.95 18 0.95 10 0.				66	0.96	33	0.94	17	1
67 0.96 34 0.97 17 0.89 67 0.94 34 0.97 17 0.89 68 0.97 34 0.94 17 0.89 68 0.96 34 0.94 17 0.94 68 0.92 34 0.97 17 0.94 68 0.94 34 0.94 17 0.89 69 0.96 35 0.97 18 0.95 69 0.96 35 0.97 18 0.95 69 0.96 35 0.97 18 0.95 69 0.96 35 0.95 18 0.95 69 0.96 35 0.95 18 0.95 70 0.97 35 0.95 18 0.95 70 0.97 35 0.95 18 0.95 70 0.93 35 0.95 18 0.95 71 0.96 36 0.97 18 0.95 71 0.				67	0.96	34	0.97	17	0.89
67 0.96 34 0.97 17 0.89 67 0.94 34 0.97 17 0.89 68 0.97 34 0.94 17 0.89 68 0.96 34 0.94 17 0.94 68 0.92 34 0.97 17 0.94 68 0.94 34 0.94 17 0.89 69 0.96 35 0.97 18 0.95 69 0.96 35 0.97 18 0.95 69 0.96 35 0.97 18 0.95 69 0.96 35 0.95 18 0.95 69 0.96 35 0.95 18 0.95 70 0.97 35 0.95 18 0.95 70 0.97 35 0.95 18 0.95 70 0.93 35 0.95 18 0.95 71 0.96 36 0.97 18 0.95 71 0.				67	0.96	34	0.94	17	0.94
68 0.97 34 0.94 17 0.89 68 0.96 34 0.94 17 0.94 68 0.92 34 0.97 17 0.94 68 0.94 34 0.94 17 0.89 69 0.96 35 0.97 18 0.95 69 0.96 35 0.97 18 0.95 69 0.96 35 0.95 18 0.95 69 0.96 35 0.95 18 0.95 69 0.96 35 0.95 18 0.95 10 0.97 35 0.95 18 0.95 10 0.96 35 0.95 18 0.95 10 0.96 35 0.95 18 0.95 10 0.96 35 0.95 18 0.95 10 0.96 35 0.95 18 0.95 11 0.96 36 0.95 18 0.95 12 0.									
68 0.97 34 0.94 17 0.89 68 0.96 34 0.94 17 0.94 68 0.92 34 0.97 17 0.94 68 0.94 34 0.94 17 0.89 69 0.96 35 0.97 18 0.95 69 0.96 35 0.97 18 0.95 69 0.96 35 0.95 18 0.95 69 0.96 35 0.95 18 0.95 69 0.96 35 0.95 18 0.95 10 0.97 35 0.95 18 0.95 10 0.96 35 0.95 18 0.95 10 0.96 35 0.95 18 0.95 10 0.96 35 0.95 18 0.95 10 0.96 35 0.95 18 0.95 11 0.96 36 0.95 18 0.95 12 0.				67	0.94	34	0.97	17	0.89
68 0.92 34 0.97 17 0.94 68 0.94 34 0.94 17 0.89 69 0.96 35 0.97 18 0.95 69 0.96 35 0.97 18 0.95 69 0.93 35 0.95 18 0.95 69 0.96 35 0.95 18 1 70 0.97 35 0.95 18 0.95 70 0.96 35 0.95 18 0.95 70 0.96 35 0.95 18 0.95 70 0.96 35 0.95 18 0.95 70 0.93 35 0.95 18 1 70 0.93 35 0.95 18 1 71 0.96 36 0.95 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96				68					
68 0.92 34 0.97 17 0.94 68 0.94 34 0.94 17 0.89 69 0.96 35 0.97 18 0.95 69 0.96 35 0.97 18 0.95 69 0.93 35 0.95 18 0.95 69 0.96 35 0.95 18 1 70 0.97 35 0.95 18 0.95 70 0.96 35 0.95 18 0.95 70 0.96 35 0.95 18 0.95 70 0.96 35 0.95 18 0.95 70 0.93 35 0.95 18 1 70 0.93 35 0.95 18 1 71 0.96 36 0.95 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96				68	0.96	34	0.94	17	0.94
68 0.94 34 0.94 17 0.89 69 0.96 35 0.97 18 0.95 69 0.96 35 0.97 18 0.95 69 0.96 35 0.95 18 0.95 69 0.96 35 0.95 18 1 70 0.97 35 0.95 18 0.95 70 0.96 35 0.95 18 0.95 70 0.96 35 0.95 18 0.95 70 0.96 35 0.95 18 0.95 70 0.93 35 0.95 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 72 0.97 36 0.95 18 0.95 72 0.97<									
69 0.96 35 0.97 18 0.95 69 0.96 35 0.97 18 0.95 69 0.93 35 0.95 18 0.95 69 0.96 35 0.95 18 1 70 0.97 35 0.95 18 0.95 70 0.96 35 0.95 18 0.95 70 0.95 35 0.95 18 1 70 0.93 35 0.95 18 1 70 0.93 35 0.95 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 72 0.97 36 0.95 18 0.99 72 0.96									
69 0.96 35 0.97 18 0.95 69 0.93 35 0.95 18 0.95 69 0.96 35 0.95 18 1 70 0.97 35 0.95 18 0.95 70 0.96 35 0.95 18 0.95 70 0.93 35 0.95 18 1 70 0.93 35 0.97 18 0.95 71 0.96 36 0.95 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 72 0.97 36 0.95 18 0.9 72 0.97 36 0.95 18 0.9 72 0.96 36 0.95 18 0.95 72 0.96									
69 0.93 35 0.95 18 0.95 69 0.96 35 0.95 18 1 70 0.97 35 0.95 18 0.95 70 0.96 35 0.95 18 0.95 70 0.93 35 0.95 18 1 70 0.93 35 0.97 18 0.95 71 0.96 36 0.95 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 72 0.97 36 0.95 18 0.9 72 0.97 36 0.95 18 0.9 72 0.96 36 0.95 18 0.95 72 0.96 36 0.95 18 0.95 72 0.94									
69 0.96 35 0.95 18 1 70 0.97 35 0.95 18 0.95 70 0.96 35 0.95 18 0.95 70 0.95 35 0.95 18 1 70 0.93 35 0.97 18 0.95 71 0.96 36 0.95 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 72 0.97 36 0.97 18 0.9 72 0.97 36 0.95 18 0.86 72 0.97 36 0.92 18 0.9 72 0.96 36 0.95 18 0.95 72 0.94 36 0.95 18 0.95 73 0.96									
70 0.97 35 0.95 18 0.95 70 0.96 35 0.95 18 0.95 70 0.95 35 0.95 18 1 70 0.93 35 0.97 18 0.95 71 0.96 36 0.95 18 0.95 71 0.96 36 0.97 18 0.95 71 0.92 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 72 0.97 36 0.95 18 0.86 72 0.97 36 0.92 18 0.9 72 0.96 36 0.95 18 0.95 72 0.94 36 0.95 18 0.95 72 0.94 36 0.95 18 0.95 72 0.94 36 0.95 19 0.95 73<									
70 0.96 35 0.95 18 0.95 70 0.95 35 0.95 18 1 70 0.93 35 0.97 18 0.95 71 0.96 36 0.95 18 0.95 71 0.96 36 0.97 18 0.95 71 0.92 36 0.97 18 0.95 71 0.96 36 0.97 18 0.95 71 0.96 36 0.97 18 0.9 72 0.97 36 0.95 18 0.86 72 0.97 36 0.92 18 0.9 72 0.96 36 0.95 18 0.95 72 0.94 36 0.95 18 0.95 72 0.94 36 0.95 18 0.95 73 0.96 37 0.95 19 0.95									
70 0.95 35 0.95 18 1 70 0.93 35 0.97 18 0.95 71 0.96 36 0.95 18 0.95 71 0.96 36 0.97 18 0.95 71 0.92 36 0.97 18 0.95 71 0.96 36 0.97 18 0.9 72 0.97 36 0.95 18 0.86 72 0.97 36 0.92 18 0.9 72 0.96 36 0.95 18 0.95 72 0.96 36 0.95 18 0.95 72 0.94 36 0.95 18 0.95 72 0.94 36 0.95 18 0.95 73 0.96 37 0.95 19 0.95									
70 0.93 35 0.97 18 0.95 71 0.96 36 0.95 18 0.95 71 0.96 36 0.97 18 0.95 71 0.92 36 0.97 18 0.95 71 0.96 36 0.97 18 0.9 72 0.97 36 0.95 18 0.86 72 0.97 36 0.92 18 0.9 72 0.96 36 0.95 18 0.95 72 0.96 36 0.95 18 0.95 72 0.94 36 0.95 18 0.95 73 0.96 37 0.95 19 0.95									
71 0.96 36 0.95 18 0.95 71 0.96 36 0.97 18 0.95 71 0.92 36 0.97 18 0.95 71 0.96 36 0.97 18 0.9 72 0.97 36 0.95 18 0.86 72 0.97 36 0.92 18 0.9 72 0.96 36 0.95 18 0.95 72 0.94 36 0.95 18 0.95 72 0.94 36 0.95 18 0.95 73 0.96 37 0.95 19 0.95									
71 0.96 36 0.97 18 0.95 71 0.92 36 0.97 18 0.95 71 0.96 36 0.97 18 0.9 72 0.97 36 0.95 18 0.86 72 0.97 36 0.92 18 0.9 72 0.96 36 0.95 18 0.95 72 0.94 36 0.95 18 0.95 72 0.94 36 0.95 18 0.95 73 0.96 37 0.95 19 0.95									
71 0.92 36 0.97 18 0.95 71 0.96 36 0.97 18 0.9 72 0.97 36 0.95 18 0.86 72 0.97 36 0.92 18 0.9 72 0.96 36 0.95 18 0.95 72 0.94 36 0.95 18 0.95 72 0.94 36 0.95 18 0.95 73 0.96 37 0.95 19 0.95									
71 0.96 36 0.97 18 0.9 72 0.97 36 0.95 18 0.86 72 0.97 36 0.92 18 0.9 72 0.96 36 0.95 18 0.95 72 0.94 36 0.95 18 0.95 73 0.96 37 0.95 19 0.95									
72 0.97 36 0.95 18 0.86 72 0.97 36 0.92 18 0.9 72 0.96 36 0.95 18 0.95 72 0.94 36 0.95 18 0.95 73 0.96 37 0.95 19 0.95									
72 0.97 36 0.92 18 0.9 72 0.96 36 0.95 18 0.95 72 0.94 36 0.95 18 0.95 73 0.96 37 0.95 19 0.95									
72 0.96 36 0.95 18 0.95 72 0.94 36 0.95 18 0.95 73 0.96 37 0.95 19 0.95									
72 0.94 36 0.95 18 0.95 73 0.96 37 0.95 19 0.95									
73 0.96 37 0.95 19 0.95									
73 0.96 37 0.95 19 0.95									
				73	0.96	37	0.95	19	0.95

I.	l.	I	ı	I	I.	l.	ı	ı	I
				73	0.96	37	0.95	19	1
				73	0.95	37	0.97	19	0.95
				74	0.97	37	0.95	19	0.95
				74	0.96	37	0.97	19	0.9
				74	0.93	37	0.93	19	1
				74	0.95	37	0.97	19	0.95
				75	0.95	38	0.93	19	0.95
				75	0.95	38	0.97	19	0.9
				75	0.95	38	0.97	19	0.95
				75	0.96	38	0.97	19	0.95
				76	0.95	38	0.95	19	0.95
				76	0.96	38	0.97	19	0.9
				76	0.95	38	0.97	19	0.9
				76	0.95	38	0.97	19	0.95
				77	0.96	39	0.95	20	0.95
				77	0.96	39	0.98	20	0.95
				77	0.96	39	0.98	20	1
				77	0.96	39	0.98	20	1
				78	0.96	39	0.98	20	0.95
				78	0.95	39	0.98	20	0.95
				78	0.94	39	0.93	20	1
				78	0.94	39	0.98	20	0.95
				79	0.98	40	0.95	20	0.95
				79	0.96	40	0.98	20	0.95
				79	0.93	40	0.95	20	0.91
				79	0.95	40	0.98	20	0.91
				80	0.96	40	0.95	20	0.91
				80	0.98	40	0.98	20	0.91
				80	0.96	40	0.95	20	0.95
				80	0.96	40	0.98	20	0.95
				81	0.96	41	0.95	21	0.95
				81	0.96	41	0.95	21	0.91
				81	0.96	41	0.98	21	1
				81	0.96	41	0.98	21	1
				82	0.98	41	0.95	21	0.95
				82	0.98	41	0.93	21	0.95
				82	0.94	41	0.95	21	0.95
				82	0.94	41	0.98	21	0.95
				83	0.97	42	0.95	21	0.95
				83	0.97	42	0.93	21	0.95
				83	0.97	42	0.98	21	0.95
				83	0.97	42	0.98	21	0.95
				84	0.98	42	0.98	21	0.91
				84	0.97	42	0.98	21	0.95
				84	0.95	42	0.95	21	0.95
				84	0.95	42	0.95	21	0.91
				85	0.97	43	0.96	22	0.96
				85	0.97	43	0.93	22	1
				85	0.97	43	0.96	22	0.96
				85	0.97	43	0.96	22	0.96
				86	0.97	43	0.96	22	0.96
				86	0.97	43	0.93	22	1
				86	0.96	43	0.98	22	1
				86	0.96	43	0.93	22	0.96
				87	0.97	44	0.98	22	1
				87	0.97	44	0.98	22	0.96
				87	0.96	44	0.96	22	0.96
				87	0.97	44	0.96	22	0.96
				88	0.98	44	0.96	22	0.96

			88	0.97	44	0.98	22	0.96
			88	0.97	44	0.96	22	0.92
			88	0.96	44	0.96	22	0.92
			89	0.97	45	0.98	23	0.96
			89	0.97	45	0.96	23	0.96
			89	0.96	45	0.94	23	0.96
			89	0.97	45	0.98	23	0.96
			90	0.98	45	0.96	23	0.92
			90	0.97	45	0.96	23	0.96
			90	0.96	45	0.98	23	1
					45			
			90	0.95		0.96	23	0.96
					46	0.96	23	
			91	0.97	46	0.98	23	0.92
			91	0.96	46	0.96	23	0.96
			91	0.96	46	0.94	23	0.92
			92	0.97	46	0.96	23	0.92
			92	0.96	46	0.96	23	0.96
			92	0.96	46	0.98	23	0.96
			92	0.95	46	0.96	23	0.92
			93	0.97	47	0.98	24	1
			93	0.98	47	0.96	24	0.96
			93	0.96	47	0.96	24	0.96
			93	0.97	47	0.96	24	0.96
			94	0.98	47	0.96	24	0.96
			94	0.97	47	0.98	24	0.96
			94	0.96	47	0.96	24	0.96
			94	0.95	47	0.96	24	0.96
			95	0.97	48	0.98	24	0.96
			95	0.95	48	0.98	24	0.92
			95	0.96	48	0.96	24	1
			95	0.97	48	0.94	24	0.96
			96	0.98	48	0.96	24	0.96
			96	0.97	48	0.98	24	0.96
			96	0.96	48	0.96	24	0.96
			96	0.95	48	0.98	24	0.92
			97	0.97	49	0.98	25	1
			97	0.97	49	0.98	25	0.96
			97	0.97	49	0.98	25	0.96
			97	0.96	49	0.96	25	0.96
			98	0.96	49	0.96	25	0.96
			98	0.97	49	0.96	25	0.96
			98	0.96	49	0.98	25	0.96
			98	0.95	49	0.98	25	0.96
			99	0.98	50	0.98	25	1
			99	0.96	50	0.98	25	0.96
			99	0.96	50	0.98	25	0.96
			99	0.96	50	0.94	25	0.96
			100	0.98	50	0.98	25	0.93
			100	0.98	50	0.98	25	0.93
			100	unknown	50	0.96	25	0.93
			100		50	0.96	25	0.93
			100	unknown				
					51	0.96	26	1
					51	0.96	26	1
1					1.51	0.98	26	0.96
					51	0.96	26	0.96
					51 51	0.96 0.98	26 26	0.96
					51 51 51	0.96 0.98 0.98	26 26 26	0.96 1 0.96
					51 51	0.96 0.98	26 26	0.96

S2	
52 0.96 26 0.96 6 52 0.95 26 0.96 6 1 52 0.96 26 0.96 8 2 0.98 2 0.93 26 0.93 9 2 0.98 26 0.93 26 0.93 27 0.96 26 0.93 27 0.96 26 0.93 27 0.93 27 0.93 27 0.93 27 0.93 27 0.93 27 0.96 28 27 0.96 28 27 0.96 28 27 0.96 28 27 0.96 28 27 0.96 28 27 0.96 28 27 0.96 28 27 0.96 28 27 0.96 28 27 0.96 28 27 1 28 27 0.96 27 1 28 27 0.96 27 1 28 27 0.96 27 1 28 27 0.96 27 0.96 28 27 0.96<	
52 0.95 26 0.96 52 0.96 26 0.96 52 0.98 26 0.96 52 0.98 26 0.93 52 0.98 26 0.93 53 0.98 27 1 53 0.98 27 0.96 53 0.98 27 0.96 53 0.98 27 0.96 53 0.98 27 0.96 53 0.98 27 0.96 53 0.98 27 0.96 53 0.98 27 1 53 0.98 27 1 53 0.98 27 1 53 0.98 27 1 53 0.98 27 1 54 0.96 27 1 54 0.96 27 0.96 54 0.98 27 0.96 54 0.96 27 0.96 54 0.96 27<	İ
52 0.96 26 0.96 52 0.98 26 0.99 52 0.96 26 0.93 52 0.98 26 0.93 53 0.98 27 1 53 0.98 27 0.96 53 0.98 27 0.96 53 0.98 27 0.96 53 0.98 27 0.96 53 0.98 27 0.96 53 0.98 27 0.96 53 0.98 27 1 53 0.98 27 1 53 0.98 27 1 53 0.98 27 1 53 0.98 27 1 53 0.98 27 1 54 0.96 27 0.96 54 0.96 27 0.93 54 0.96 27 0.93 54 0.96 27 0.93 54 0.96 27<	
52 0.98 26 0.96 52 0.96 26 0.93 52 0.98 26 0.93 53 0.98 27 1 53 0.98 27 0.96 53 0.98 27 0.96 53 0.98 27 0.96 53 0.98 27 0.96 53 0.98 27 0.96 53 0.98 27 0.96 53 0.98 27 0.96 53 0.98 27 0.96 53 0.98 27 1 53 0.98 27 1 53 0.98 27 1 54 0.96 27 0.96 54 0.98 27 0.96 54 0.98 27 0.93 54 0.96 27 0.93 54 0.98 27 0.96 </td <td></td>	
52 0.96 26 0.93 52 0.98 26 0.93 53 0.08 27 1 53 0.98 27 0.96 53 0.98 27 0.96 53 0.98 27 0.96 53 0.98 27 0.96 53 0.98 27 0.93 53 0.98 27 1 53 0.98 27 1 53 0.98 27 1 53 0.98 27 1 53 0.98 27 1 53 0.98 27 1 53 0.98 27 1 53 0.98 27 1 54 0.96 27 0.96 54 0.98 27 0.93 54 0.98 27 0.93 54 0.96 27 0.93	
52 0.98 26 0.93 53 0.98 27 1 53 0.98 27 0.96 53 0.98 27 0.96 53 0.98 27 0.96 53 0.98 27 0.96 53 0.98 27 0.93 53 0.98 27 1 53 0.98 27 1 53 0.98 27 1 53 0.98 27 1 53 0.98 27 1 53 0.96 27 1 54 0.96 27 0.96 54 0.98 27 0.93 54 0.98 27 0.93 54 0.96 27 0.96 54 0.96 27 0.93 54 0.96 27 0.93 54 0.96 27 0.93 54 0.96 27 0.93 54 0.96 27<	
S3	
53 0.98 27 0.96 53 0.98 27 0.96 53 0.98 27 0.96 53 0.98 27 0.93 53 0.98 27 1 53 0.98 27 1 53 0.96 27 1 53 0.98 27 1 53 0.96 27 1 54 0.96 27 0.96 54 0.98 27 0.96 54 0.98 27 0.96 54 0.98 27 0.96 54 0.96 27 0.93 54 0.96 27 0.93 54 0.96 27 0.93 54 0.96 27 0.93 54 0.96 27 0.93 54 0.96 27 0.93 54 0.96 27 0.93 55 0.98 28 1 55 0.98	
53 0.98 27 0.96 53 0.98 27 0.96 53 0.98 27 0.96 53 0.98 27 0.93 53 0.98 27 1 53 0.98 27 1 53 0.96 27 1 53 0.98 27 1 53 0.96 27 1 54 0.96 27 0.96 54 0.98 27 0.96 54 0.98 27 0.96 54 0.98 27 0.96 54 0.96 27 0.93 54 0.96 27 0.93 54 0.96 27 0.93 54 0.96 27 0.93 54 0.96 27 0.93 54 0.96 27 0.93 54 0.96 27 0.93 55 0.98 28 1 55 0.98	
S3	
53 0.98 27 0.96 53 0.98 27 0.93 53 0.98 27 1 53 0.96 27 1 53 0.98 27 1 53 0.98 27 1 54 0.96 27 0.96 54 0.98 27 0.96 54 0.98 27 0.93 54 0.96 27 0.96 54 0.96 27 0.96 54 0.96 27 0.93 54 0.96 27 0.96 54 0.96 27 0.93 54 0.96 27 0.93 54 0.96 27 0.93 54 0.96 27 0.93 54 0.96 27 0.93 55 0.98 28 0.97 55 0.98 28 0.97 55 0.96 28 0.97 55 0.95	
53 0.98 27 0.93 53 0.98 27 1 53 0.96 27 1 53 0.98 27 1 54 0.96 27 0.96 54 0.98 27 0.93 54 0.98 27 0.93 54 0.96 27 0.93 54 0.96 27 0.93 54 0.96 27 0.93 54 0.96 27 0.93 54 0.96 27 0.93 54 0.96 27 0.93 54 0.96 27 0.93 54 0.96 27 0.93 54 0.96 27 0.93 55 0.98 28 1 55 0.96 28 0.97 55 0.98 28 0.97 55 0.95 28 0.97 55 0.96 28 0.97 55 0.96	
53 0.98 27 1 53 0.96 27 1 53 0.98 27 1 54 0.96 27 0.96 54 0.98 27 0.96 54 0.98 27 0.93 54 0.96 27 0.96 54 0.96 27 0.96 54 0.96 27 0.96 54 0.96 27 0.93 54 0.96 27 0.93 54 0.96 27 0.93 54 0.96 27 0.93 54 0.96 27 0.93 54 0.96 27 0.93 55 0.98 28 1 55 0.96 28 0.97 55 0.98 28 0.97 55 0.95 28 0.97 55 0.96 28 1 55 0.96 28 1 55 0.96	
53 0.96 27 1 53 0.98 27 1 54 0.96 27 0.96 54 0.98 27 0.96 54 0.98 27 0.96 54 0.96 27 0.96 54 0.96 27 0.93 54 0.96 27 0.93 54 0.96 27 0.93 54 0.96 27 0.93 54 0.96 27 0.93 54 0.96 27 0.93 54 0.96 27 0.93 54 0.96 27 0.93 54 0.96 27 0.93 54 0.96 27 0.93 55 0.98 28 1 55 0.98 28 0.97 55 0.98 28 0.97 55 0.95 28 0.97 55 0.96 28 1 55 0.96	
53 0.98 27 1 54 0.96 27 0.96 54 0.98 27 0.96 54 0.98 27 0.93 54 0.96 27 0.96 54 0.96 27 0.93 54 0.96 27 0.93 54 0.96 27 0.93 54 0.96 27 0.93 54 0.96 27 0.93 54 0.96 27 0.93 54 0.96 27 0.93 54 0.96 27 0.93 55 0.98 28 1 55 0.98 28 0.97 55 0.96 28 0.97 55 0.95 28 0.97 55 0.96 28 0.97 55 0.96 28 0.97 55 0.96 28 0.97 55 0.96 28 0.97 55 0.96 <td></td>	
54 0.96 27 0.96 54 0.98 27 0.96 54 0.98 27 0.93 54 0.96 27 0.96 54 0.96 27 0.93 54 0.96 27 0.93 54 0.98 27 0.96 54 0.96 27 0.93 54 0.96 27 0.93 54 0.96 27 0.93 54 0.96 27 0.93 55 0.98 28 1 55 0.98 28 1 55 0.98 28 0.97 55 0.95 28 0.97 55 0.96 28 0.93 55 0.95 28 0.97 55 0.96 28 0.93 55 0.96 28 1 55 0.96 28 1 55 0.96 28 1 55 0.96 <	
54 0.98 27 0.96 54 0.98 27 0.93 54 0.96 27 0.96 54 0.96 27 0.93 54 0.98 27 0.96 54 0.96 27 0.93 54 0.96 27 0.96 55 0.98 28 1 55 0.96 28 0.97 55 0.96 28 0.97 55 0.95 28 0.97 55 0.95 28 0.97 55 0.96 28 0.97 55 0.96 28 0.97 55 0.96 28 0.97 55 0.96 28 0.97 55 0.96 28 1 55 0.96 28 0.97 55 0.96 28 1 55 0.96 28 1 55 0.96 28 0.97 56 0.98	
54 0.98 27 0.93 54 0.96 27 0.96 54 0.96 27 0.93 54 0.98 27 0.96 54 0.96 27 0.93 54 0.96 27 0.96 55 0.98 28 1 55 0.98 28 0.97 55 0.98 28 0.97 55 0.95 28 0.97 55 0.95 28 0.97 55 0.96 28 0.97 55 0.96 28 0.97 55 0.96 28 0.97 55 0.96 28 0.97 55 0.96 28 0.97 55 0.96 28 1 55 0.98 28 0.97 55 0.96 28 1 55 0.96 28 1 55 0.96 28 0.97 56 0.98	
54 0.96 27 0.96 54 0.96 27 0.93 54 0.98 27 0.96 54 0.96 27 0.93 54 0.96 27 0.96 55 0.98 28 1 55 0.98 28 0.97 55 0.96 28 0.97 55 0.98 28 0.97 55 0.95 28 0.97 55 0.98 28 0.97 55 0.98 28 0.93 55 0.96 28 0.97 55 0.98 28 0.97 55 0.96 28 0.97 55 0.96 28 1 55 0.96 28 1 55 0.96 28 0.97 56 0.98 28 0.97 56 0.98 28 0.93 56 0.97 28 0.93 56 0.97	
54 0.96 27 0.93 54 0.98 27 0.96 54 0.96 27 0.93 54 0.96 27 0.96 55 0.98 28 1 55 0.96 28 0.97 55 0.96 28 0.97 55 0.96 28 0.97 55 0.95 28 0.97 55 0.95 28 0.97 55 0.96 28 0.97 55 0.96 28 1 55 0.96 28 1 55 0.96 28 1 55 0.96 28 1 55 0.96 28 1 55 0.96 28 0.97 56 0.98 28 0.97 56 0.98 28 0.97 56 0.99 28 0.93 56 0.97 28 0.93	
54 0.98 27 0.96 54 0.96 27 0.93 54 0.96 27 0.96 55 0.98 28 1 55 0.96 28 0.97 55 0.98 28 0.97 55 0.95 28 0.97 55 0.95 28 0.97 55 0.98 28 0.93 55 0.96 28 1 55 0.96 28 1 55 0.96 28 1 55 0.96 28 1 55 0.96 28 1 55 0.96 28 1 56 0.98 28 0.97 56 0.98 28 0.97 56 0.95 28 0.93 56 0.95 28 0.97 56 0.97 28 0.97 56 0.97 28 0.97 56 0.98 28<	
54 0.96 27 0.93 54 0.96 27 0.96 55 0.98 28 1 55 0.96 28 0.97 55 0.98 28 0.97 55 0.95 28 0.97 55 0.95 28 0.97 55 0.98 28 0.93 55 0.96 28 1 55 0.96 28 1 55 0.96 28 1 55 0.96 28 0.97 56 0.98 28 0.97 56 0.98 28 0.97 56 0.95 28 0.93 56 0.97 28 0.93 56 0.97 28 0.97 56 0.97 28 0.97 56 0.98 28 0.93	
54 0.96 27 0.96 55 0.98 28 1 55 0.96 28 0.97 55 0.98 28 0.97 55 0.95 28 0.97 55 0.95 28 0.97 55 0.96 28 0.97 55 0.96 28 1 55 0.96 28 1 55 0.96 28 0.97 56 0.98 28 0.97 56 0.98 28 0.97 56 0.98 28 0.97 56 0.95 28 0.93 56 0.97 28 0.97 56 0.97 28 0.97 56 0.98 28 0.97 56 0.99 28 0.97 56 0.99 28 0.97 56 0.99 28 0.93	
55 0.98 28 1 55 0.96 28 0.97 55 0.98 28 0.97 55 0.95 28 0.97 55 0.95 28 0.97 55 0.98 28 0.93 55 0.96 28 1 55 0.96 28 0.97 56 0.98 28 0.97 56 0.98 28 0.97 56 0.98 28 0.97 56 0.95 28 0.93 56 0.97 28 0.93 56 0.97 28 0.93	
55 0.96 28 0.97 55 0.98 28 0.97 55 0.95 28 0.97 55 0.95 28 0.97 55 0.98 28 0.93 55 0.96 28 1 55 0.96 28 1 55 0.96 28 0.97 56 0.98 28 0.97 56 0.98 28 0.97 56 0.98 28 0.97 56 0.95 28 0.93 56 0.97 28 0.97 56 0.98 28 0.93	
55 0.98 28 0.97 55 0.95 28 0.97 55 0.95 28 0.97 55 0.98 28 0.93 55 0.96 28 1 55 0.96 28 0.97 56 0.98 28 0.97 56 0.98 28 0.97 56 0.98 28 0.97 56 0.95 28 0.93 56 0.97 28 0.97 56 0.98 28 0.93 56 0.98 28 0.93	
55 0.95 28 0.97 55 0.95 28 0.97 55 0.98 28 0.93 55 0.96 28 1 55 0.96 28 0.97 56 0.98 28 1 56 0.98 28 0.97 56 0.98 28 0.97 56 0.95 28 0.93 56 0.97 28 0.93 56 0.98 28 0.97 56 0.95 28 0.93 56 0.98 28 0.97 56 0.98 28 0.93	
55 0.95 28 0.97 55 0.98 28 0.93 55 0.96 28 1 55 0.96 28 0.97 56 0.98 28 1 56 0.98 28 0.97 56 0.98 28 0.97 56 0.95 28 0.93 56 0.97 28 0.97 56 0.98 28 0.93 56 0.98 28 0.93	
55 0.98 28 0.93 55 0.96 28 1 55 0.96 28 0.97 56 0.98 28 1 56 0.98 28 0.97 56 0.98 28 0.97 56 0.98 28 0.97 56 0.95 28 0.93 56 0.97 28 0.97 56 0.98 28 0.93	
55 0.96 28 1 55 0.96 28 0.97 56 0.98 28 1 56 0.98 28 0.97 56 0.98 28 0.97 56 0.95 28 0.93 56 0.97 28 0.97 56 0.98 28 0.97 56 0.98 28 0.93	
55 0.96 28 0.97 56 0.98 28 1 56 0.98 28 0.97 56 0.98 28 0.97 56 0.95 28 0.93 56 0.97 28 0.97 56 0.98 28 0.93	
55 0.96 28 0.97 56 0.98 28 1 56 0.98 28 0.97 56 0.98 28 0.97 56 0.95 28 0.93 56 0.97 28 0.97 56 0.98 28 0.93	
56 0.98 28 1 56 0.98 28 0.97 56 0.98 28 0.97 56 0.95 28 0.93 56 0.97 28 0.97 56 0.98 28 0.93	
56 0.98 28 0.97 56 0.98 28 0.97 56 0.95 28 0.93 56 0.97 28 0.97 56 0.98 28 0.93	
56 0.98 28 0.97 56 0.95 28 0.93 56 0.97 28 0.97 56 0.98 28 0.93	
56 0.95 28 0.93 56 0.97 28 0.97 56 0.98 28 0.93	
56 0.97 28 0.97 56 0.98 28 0.93	
56 0.98 28 0.93	
56 0.97 28 0.97	
57 0.98 29 1	
57 0.97 29 0.97	
57 0.98 29 0.97	
57 0.97 29 0.97	
57 0.98 29 0.97	
57 0.97 29 0.97	
57 0.98 29 0.97	
57 0.97 29 0.97	
58 0.97 29 1	
58 0.98 29 0.97	
58 0.98 29 0.97	
58 0.97 29 0.94	
58 0.98 29 0.94	
58 0.98 29 0.97	
58 0.97 29 0.97	I
58 0.97 29 0.97	
59 0.98 30 0.97	
59 0.98 30 0.97	
59 0.98 30 0.97	

	ì	ı			ı			1
					59	0.95	30	0.97
Section Sect					59	0.98	30	1
S9					59	0.97	30	0.97
					59	0.97	30	0.97
					59	0.97	30	0.97
					60	1	30	0.94
					60	0.97	30	0.97
					60	0.98	30	0.94
					60	0.97	30	0.97
60					60	0.98	30	0.97
60								
61 0.95 31 0.97 61 0.98 31 0.97 61 0.97 31 0.97 61 0.97 31 1 61 0.97 31 1 61 0.97 31 1 62 1 31 1 62 0.97 31 0.97 62 0.98 31 0.97 62 0.97 31 0.97 62 0.97 31 0.97 63 0.98 31 0.97 64 0.97 31 0.97 65 0.98 31 0.97 66 0.97 31 0.97 67 0.97 31 0.97 68 0.97 31 0.97 69 0.97 31 0.97 60 0.97 31 0.97 60 0.97 31 0.97 61 0.97 31 0.97 62 0.97 31 0.97 63 0.98 32 0.97 64 0.97 32 0.97 65 0.98 32 0.97 66 0.98 32 0.97 67 0.98 32 0.97 68 0.98 32 0.97 69 0.98 32 0.97 60 0.98 32 0.97 61 0.98 32 0.97 62 0.97 32 0.97 63 0.98 32 0.97 64 0.97 32 0.97 65 0.98 32 0.97 66 0.98 32 0.97 67 0.98 32 0.97 68 0.98 32 0.97 69 0.98 32 0.97 60 0.98 32 0.97 60 0.98 32 0.97 60 0.98 32 0.97 60 0.98 32 0.97 60 0.98 32 0.97 60 0.98 32 0.97								
61 0.98 31 0.97 61 0.97 31 0.97 61 0.97 31 1 61 0.97 31 1 62 1 31 1 62 0.97 31 0.97 63 0.98 31 0.97 64 0.97 31 0.97 65 0.98 31 0.97 67 0.97 31 0.97 68 0.97 31 0.97 69 0.97 31 0.97 60 0.97 31 0.97 61 0.97 31 0.97 62 0.97 31 0.94 63 0.98 32 0.97 63 0.98 32 0.97 63 0.98 32 0.97 64 0.97 32 0.97 65 0.97 32 0.97 66 0.97 32 0.97 67 0.97 32 0.97 68 0.98 32 0.97 69 0.98 32 0.97 60 0.98 32 0.97 60 0.98 32 0.97 60 0.98 32 0.97 60 0.98 32 0.97 60 0.98 32 0.97 60 0.98 32 0.97 60 0.98 32 0.97 60 0.98 32 0.97 60 0.98 32 0.97 60 0.98 32 0.97 60 0.98 32 0.97 60 0.98 32 0.97 60 0.98 32 0.97 60 0.98 32 0.97 60 0.99 32 0.97 60 0.99 32 0.97 60 0.99 32 0.97								
62 1 31 1 62 0.97 31 0.97 62 0.98 31 0.97 62 0.97 31 0.97 62 0.97 31 0.94 62 0.97 31 0.94 62 0.97 31 0.94 62 0.97 31 0.94 62 0.97 31 0.94 62 0.97 31 0.94 62 0.97 31 0.94 63 0.98 32 0.97 63 0.98 32 0.97 63 0.98 32 0.97 63 0.98 32 0.97 63 0.98 32 0.97 63 0.98 32 0.97 63 0.95 32 0.97 63 0.98 32 0.97 64 1 32 0.97 64 0.97 32 0.97 64 0.97								
62 0.97 31 0.97 62 0.98 31 0.97 62 0.97 31 0.97 62 0.97 31 0.97 62 0.97 31 0.94 62 0.97 31 0.94 62 0.97 31 0.94 62 0.97 31 0.94 62 0.97 31 0.97 63 0.98 32 0.97 63 0.98 32 0.97 63 0.98 32 0.97 63 0.98 32 0.97 63 0.98 32 0.97 63 0.98 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 65 0.98 32 0.97 66 0.98 32 0.97 66 0.99 32 0.97 66 0.99 32 0.97 66 0.99 32 0.97 67 0.90 0.90 0.90 0.90 0.90 0.90 0.90 0.9								
62 0.98 31 0.97 62 0.97 31 0.97 62 0.97 31 0.94 62 0.97 31 0.97 62 0.97 31 0.97 62 0.97 31 0.97 63 0.98 32 0.97 63 0.98 32 0.97 63 0.98 32 0.97 63 0.98 32 0.97 63 0.98 32 0.97 63 0.98 32 0.97 63 0.98 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 65 0.98 32 0.97 66 0.98 32 0.97 66 0.98 32 0.97 66 0.98 32 0.97 66 0.98 32 0.97 67 0.98 0.98 0.98 68 0.99 0.98 0.98 69 0.99 0.99 69 0					62		31	
62 0.97 31 0.97 62 0.97 31 0.94 62 0.97 31 0.97 62 0.97 31 0.94 62 0.97 31 0.94 62 0.97 31 0.94 62 0.97 31 0.97 63 0.98 32 0.97 63 0.98 32 0.97 63 0.98 32 0.97 63 0.98 32 0.97 63 0.98 32 0.97 63 0.98 32 0.97 63 0.98 32 0.97 63 0.98 32 0.97 63 0.98 32 0.97 63 0.98 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 65 0.98 32 0.97 66 0.98 32 0.97					62	0.97	31	0.97
62 0.97 31 0.94 62 0.97 31 0.97 62 0.97 31 0.94 62 0.97 31 0.97 63 0.98 32 0.97 63 0.98 32 0.97 63 0.98 32 0.97 63 0.98 32 0.97 63 0.95 32 0.97 63 0.98 32 0.94 63 0.98 32 0.97 63 0.98 32 0.97 63 0.98 32 0.97 63 0.98 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.					62	0.98	31	0.97
62 0.97 31 0.97 62 0.97 31 0.94 62 0.97 31 0.97 63 0.98 32 0.97 63 0.98 32 0.97 63 0.98 32 0.97 63 0.95 32 0.97 63 0.98 32 0.97 63 0.98 32 0.97 63 0.98 32 0.97 63 0.98 32 0.97 63 0.95 32 0.97 63 0.95 32 0.97 63 0.95 32 0.97 64 1 32 0.94 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97<					62	0.97	31	0.97
62 0.97 31 0.94 62 0.97 31 0.97 63 0.98 32 0.97 63 0.98 32 0.97 63 0.98 32 0.97 63 0.98 32 0.97 63 0.98 32 0.97 63 0.98 32 0.97 63 0.98 32 0.97 63 0.98 32 0.97 64 1 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 65 0.98 32 0.97 66 0.98 32 0.97 67 0.97 68 0.97 69 0.97 69 0.97 60 0.97 60 0.97 60 0.98 60 0.97 60 0.97 60 0.98 60 0.97 60 0.98 60 0.99					62	0.97	31	0.94
62 0.97 31 0.97 63 0.98 32 0.97 63 0.98 32 0.97 63 0.98 32 0.97 63 0.98 32 0.97 63 0.98 32 0.97 63 0.98 32 0.97 63 0.98 32 0.97 63 0.98 32 0.97 63 0.97 32 0.97 64 1 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 65 0.98 32 0.97 66 0.97 32 0.97 67 0.97 0.97 68 0.97 0.97 69 0.97 0.97 60 0.97 0.97 60 0.97 0.97 60 0.97 61 0.97 62 0.97 63 0.98 0.97 64 0.97 65 0.98 0.98 0.97					62	0.97	31	0.97
63 0.98 32 0.97 63 0.98 32 0.97 63 0.98 32 0.97 63 0.98 32 0.97 63 0.98 32 0.97 63 0.98 32 0.97 63 0.98 32 0.97 63 0.97 32 0.97 63 0.98 32 0.97 64 1 32 0.94 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 65 0.98 32 0.97 66 0.97 32 0.97 67 0.97 0.97 68 0.98 0.97 69 0.97 0.97 60 0.97 0.97 60 0.97 0.97 60 0.97 0.97 60 0.97 0.97 60 0.97 0.97 60 0.98 0.97 60 0.97 0.97 60 0.97					62	0.97	31	0.94
63 0.98 32 0.97 63 0.98 32 0.97 63 0.95 32 0.97 63 0.98 32 0.97 63 0.98 32 0.97 63 0.97 32 0.97 63 0.95 32 0.97 63 0.95 32 0.97 64 1 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 65 0.98 33 0.97					62	0.97	31	0.97
63 0.98 32 0.97 63 0.98 32 0.97 63 0.98 32 0.94 63 0.97 32 0.97 63 0.98 32 0.97 63 0.95 32 0.97 64 1 32 0.94 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 65 0.98 32 0.97 66 0.97 32 0.97 66 0.98 32 0.97					63	0.98	32	0.97
63 0.98 32 0.97 63 0.98 32 0.97 63 0.98 32 0.94 63 0.97 32 0.97 63 0.98 32 0.97 63 0.95 32 0.97 64 1 32 0.94 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 65 0.98 32 0.97 66 0.97 32 0.97 66 0.98 32 0.97					63	0.98	32	0.97
63 0.95 32 0.97 63 0.98 32 0.94 63 0.97 32 0.97 63 0.98 32 0.97 63 0.98 32 0.97 64 1 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 65 0.98 33 0.97								
63 0.98 32 0.94 63 0.97 32 0.97 63 0.98 32 0.97 63 0.98 32 0.97 64 1 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.98 32 0.97 64 0.97 32 0.97 64 0.98 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.98 32 0.97 65 0.98 33 0.97								
63 0.97 32 0.97 63 0.95 32 0.97 63 0.98 32 0.97 64 1 32 0.94 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.98 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 65 0.98 33 0.97								
63 0.95 32 0.97 63 0.98 32 0.97 64 1 32 0.94 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.98 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 65 0.98 33 0.97								
63 0.98 32 0.97 64 1 32 0.94 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.98 32 0.97 64 0.98 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 65 0.98 33 0.97								
64 1 32 0.94 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.98 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.96 32 0.94 65 0.98 33 0.97 65 0.98 33 0.97								
64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.98 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.96 32 0.94 65 0.98 33 0.97 65 0.98 33 0.97								
64 0.97 32 0.97 64 0.97 32 0.97 64 0.98 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.96 32 0.94 65 0.98 33 0.97 65 0.98 33 0.97								
64 0.97 32 0.97 64 0.98 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.96 32 0.94 65 0.98 33 0.97 65 0.98 33 0.97								
64 0.98 32 0.97 64 0.97 32 0.97 64 0.97 32 0.97 64 0.96 32 0.94 65 0.98 33 0.97 65 0.98 33 0.97								
64 0.97 32 0.97 64 0.97 32 0.97 64 0.96 32 0.94 65 0.98 33 0.97 65 0.98 33 0.97								
64 0.97 32 0.97 64 0.96 32 0.94 65 0.98 33 0.97 65 0.98 33 0.97								
64 0.96 32 0.94 65 0.98 33 0.97 65 0.98 33 0.97								
65 0.98 33 0.97 65 0.98 33 0.97								
65 0.98 33 0.97								0.94
						0.98	33	0.97
					65	0.98	33	0.97
65 0.97 33 0.97					65	0.97	33	0.97
65 0.96 33 0.97					65	0.96	33	0.97
65 0.98 33 0.97					65	0.98	33	0.97
65 0.98 33 0.97					65	0.98	33	0.97
65 0.97 33 1					65	0.97	33	1
65 0.97 33 0.97								
66 1 33 1								
66 0.97 33 0.97								
66 0.99 33 0.94								
66 0.97 33 0.97								
66 0.97 33 0.97					00	0.97	33	0.97

1	1				ı			1
					66	0.97	33	0.97
					66	0.97	33	1
					67	1	34	1
					67	0.99	34	0.97
					67	0.99	34	0.97
					67	0.99	34	0.97
					67	0.97	34	0.97
					67	0.96	34	0.97
					67	0.97	34	0.97
					67	0.97	34	0.94
					68	0.99	34	0.97
					68	0.97	34	1
					68	0.97	34	0.97
					68	0.99	34	0.94
					68	0.96	34	0.94
					68	0.97	34	0.94
					68	0.97	34	0.97
					68	0.97	34	0.94
					69	1	35	1
					69	0.99	35	0.97
					69	0.99	35	1
					69	0.99	35	0.97
					69	0.96	35	0.97
					69	0.99	35	0.97
					69	0.99	35	
								0.97
					69	0.97	35	1
					70	0.99	35	0.97
					70	0.97	35	0.97
					70	0.97	35	0.97
					70	0.99	35	0.97
					70	0.97	35	0.97
					70	0.97	35	0.97
					70	0.97	35	0.97
					70	0.99	35	0.95
					71	1	36	0.97
					71	0.99	36	0.97
					71	0.99	36	0.95
					71	0.99	36	1
					71	0.96	36	0.95
					71	0.97	36	0.97
					71	0.97	36	0.97
					71	0.99	36	0.97
					72	0.99	36	0.97
					72	0.99	36	0.97
					72	0.97	36	0.95
					72	1	36	0.97
					72	0.97	36	0.95
					72	0.97	36	0.95
					72	0.97	36	0.95
					72	0.97	36	0.95
					73	1	37	0.97
					73	0.99	37	0.97
					73	0.97	37	0.97
					73	0.99	37	0.97
					73	0.97	37	0.97
					73	0.97	37	0.97
					73	0.97	37	0.97
					73	0.97	37	0.97
					74	1	37	0.95
	1	I	1		<u> </u>	l	1	1

1	1	1	1		ı	1		1
					74	0.99	37	0.97
					74	0.99	37	0.97
					74	0.99	37	0.95
					74	0.96	37	0.97
					74	0.97	37	0.95
					74	0.97	37	0.97
					74	0.97	37	0.95
					75	1	38	0.97
					75	0.99	38	0.97
					75	0.99	38	0.97
					75	0.99	38	1
					75	0.97	38	1
					75	0.97	38	0.97
					75	0.99	38	0.97
					75	0.97	38	0.97
					76	1	38	0.97
					76	0.97	38	0.97
					76	0.99	38	0.95
					76	1	38	0.95
					76	0.96	38	0.97
					76	0.97	38	0.95
					76	0.97	38	1
					76	0.97	38	0.97
					77	0.99	39	0.98
					77	0.99	39	0.98
					77	0.99	39	1
					77	1	39	0.98
					77	0.97	39	0.98
					77	0.97	39	0.98
					77	0.97	39	0.95
					77	0.97	39	0.98
					78	1	39	0.98
					78	0.99	39	0.98
					78	0.98	39	0.98
					78	0.99	39	0.95
					78	0.96	39	0.98
					78	0.96	39	0.95
					78	0.98	39	0.95
					78	0.98	39	0.95
					79	1	40	0.98
					79	0.99	40	0.98
					79	0.98	40	0.98
					79	0.98	40	0.98
					79	0.96	40	1
					79	0.98	40	0.98
					79	0.98	40	0.98
					79		40	0.98
						0.98		
					80	1 0.00	40	0.98
					80	0.99	40	0.98
					80	0.98	40	0.98
					80	0.99	40	0.98
					80	0.98	40	0.98
					80	0.98	40	0.98
					80	0.98	40	0.98
					80	0.98	40	0.98
					81	1	41	0.98
					81	0.98	41	0.98
					81	0.98	41	0.98
					81	0.99	41	1

1	1	1		ı			1
				81	0.98	41	1
				81	0.98	41	0.98
				81	0.98	41	0.98
				81	0.98	41	0.98
				82	0.99	41	1
				82	0.99	41	0.98
				82	0.99	41	0.95
				82	1	41	0.95
				82	0.98	41	0.98
				82	0.98	41	0.98
				82	0.98	41	0.95
				82	0.99	41	0.98
				83	1	42	0.98
				83	0.99	42	1
				83	0.98	42	1
				83	0.98	42	1
				83	0.98	42	0.98
				83	0.98	42	0.98
				83	0.97	42	0.98
				83	0.97	42	0.98
				84	1	42	1
				84	0.99	42	0.98
				84	0.99	42	0.98
				84	0.99	42	0.98
				84	0.98	42	0.98
				84	0.98	42	0.98
				84	0.98	42	0.95
				84	0.98	42	0.98
				85	0.99	43	0.98
				85	0.99	43	0.98
				85	0.98	43	1
				85	0.99	43	1
				85	0.98	43	0.98
				85	0.98	43	1
				85	0.97	43	1
				85	0.98	43	1
				86	1	43	0.96
				86	0.99	43	0.98
				86	0.99	43	0.96
				86	0.99	43	0.96
				86	0.97	43	0.98
				86	0.97	43	0.96
				86	0.98	43	0.98
				86	0.97	43	0.96
				87	1	44	1
				87	0.99	44	0.98
				87	0.98	44	1
				87	0.99	44	0.98
				87	0.99	44	0.98
				87	0.98	44	0.98
				87	0.98	44	0.96
				87	0.98	44	0.98
				88	0.99	44	0.98
				88	0.99	44	0.98
				88	0.99	44	0.98
				88	1	44	0.98
				88	0.98	44	0.96
				88	0.98	44	0.96
				 88	0.98	44	0.96
							

	ı	ì			I.	ı		ı
					88	0.98	44	0.98
					89	0.99	45	0.98
					89	0.99	45	1
					89	0.98	45	1
S9					89	0.99	45	1
S9					89	0.98	45	0.98
					89	0.98	45	0.98
					89	0.98	45	1
					89	0.98	45	0.98
90 0.98 45 0.98 90 0.98 45 0.96 90 1 45 0.98 90 0.98 45 0.98 90 0.98 45 0.98 90 0.98 45 0.98 90 0.97 45 0.98 90 0.97 45 0.98 90 0.99 45 0.98 91 0.99 46 1 91 0.99 46 1 91 0.99 46 1 91 0.99 46 1 91 0.99 46 1 91 0.99 46 1 91 0.99 46 1 91 0.99 46 1 91 0.99 46 0.98 91 0.98 46 0.98 91 0.98 46 0.98 91 0.98 46 0.98 91 0.98 46 0.98 92 0.98 46 0.98 92 0.98 46 0.98 92 0.98 46 0.98 93 0.99 47 1 94 0.98 47 0.98 94 0.99 47 0.9					90	1	45	0.98
90 0.98 45 0.96								
90								
90								
90								
91 0.99 46 1 91 0.99 46 0.98 91 0.99 46 0.98 91 0.99 46 0.98 91 0.99 46 0.98 91 0.98 46 0.98 91 0.98 46 0.98 91 0.98 46 1 91 0.98 46 0.98 91 0.98 46 1 91 0.98 46 0.98 91 0.98 46 0.98 92 0.98 46 0.98 92 0.98 46 0.98 92 0.99 46 0.98 92 0.99 46 0.98 92 0.99 46 0.98 92 0.99 46 0.98 92 0.99 47 1 93 0.99 47 1 93 0.99 47 0.98 93 0.99 47 1 93 0.98 47 0.98 94 0.98 47 0.98 94 0.98 47 0.98								
91 0.99 46 10 0.98 91 0.99 46 0.98 91 0.99 46 0.98 91 0.99 46 0.98 91 0.98 46 0.98 91 0.98 46 1 91 0.97 46 1 1 1 46 0.98 92 1 1 46 0.98 92 0.98 46 0.98 92 0.98 46 0.98 92 0.98 46 0.98 92 0.98 46 0.98 92 0.98 46 0.98 92 0.99 46 0.98 92 0.99 46 0.98 93 0.99 47 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1								
91 0.99 46 0.98 91 0.99 46 0.98 91 0.99 46 0.98 91 0.98 46 0.96 91 0.98 46 0.96 91 0.97 46 1 92 1 46 0.98 92 0.98 46 0.96 92 0.98 46 0.96 92 0.99 46 0.98 92 0.99 46 0.98 92 0.97 46 0.98 92 0.97 46 0.98 92 0.97 46 0.98 92 0.98 46 0.98 92 0.98 46 0.96 92 0.98 46 0.98 93 0.99 47 1 93 0.99 47 1 93 0.99 47 1 93 0.99 47 0.98 94 0.98 47 0.98 94 0.98 47 0.98 94 0.98 47 0.98 94 0.98 47 0.98 94 0.98 47 0.98 94 0.98 47 0.98 94 0.98 47 0.98 94 0.98 47 0.98 94 0.98 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.98 47 0.98 94 0.99 47 0.98 94 0.98 47 0.98 94 0.99 47 0.98 94 0.98 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.98 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.98 47 0.98 94 0.99 47 0.98 94 0.98 47 0.98 94 0.99 47 0.98 94 0.98 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98								
91 0.99 46 0.98 91 0.98 46 0.96 91 0.98 46 0.96 91 0.98 46 1 91 0.97 46 1 92 1 46 0.98 92 0.98 46 0.96 92 0.98 46 0.96 92 0.98 46 0.98 92 0.99 46 0.98 92 0.97 46 0.98 92 0.97 46 0.98 92 0.97 46 0.98 92 0.98 46 0.96 92 0.98 46 0.96 92 0.98 46 0.96 92 0.98 46 0.98 93 0.99 47 1 93 0.99 47 1 94 0.99 47 0.98 95 0.95 0.95 96 0.95 0.95 96 0.95 0.95 97 0.95 0.95 98 0.95 0.95 99 0.95 0.95 90 0.95								
91 0.98 46 0.96 91 0.97 46 1 92 1 46 0.98 92 0.98 46 0.98 92 0.98 46 0.98 92 0.99 46 0.98 92 0.99 46 0.98 92 0.99 46 0.98 92 0.99 46 0.98 92 0.98 46 0.98 92 0.99 47 1 93 0.98 47 0.98 93 0.98 47 0.98 93 0.99 47 0.98 94 0.98 47 0.98 94 0.99 47 0.98								
91 0.98 46 1 91 0.97 46 1 91 0.97 46 1 92 1 46 0.98 92 0.98 46 0.96 92 0.99 46 0.98 92 0.97 46 0.98 92 0.97 46 0.98 92 0.97 46 0.98 92 0.98 46 0.98 92 0.97 46 0.98 92 0.98 46 0.98 92 0.98 46 0.98 92 0.98 46 0.98 92 0.98 46 0.98 92 0.98 46 0.98 92 0.98 46 0.98 93 1 47 1 93 0.99 47 1 93 0.99 47 1 93 0.99 47 1 93 0.99 47 1 93 0.98 47 0.98 93 0.98 47 0.98 93 0.98 47 0.98 93 0.98 47 0.98 93 0.98 47 0.98 93 0.98 47 0.98 94 0.99 47 0.98						0.99	46	0.98
91 0.97 46 1 92 1 46 0.98 92 0.98 46 0.96 92 0.99 46 0.98 92 0.97 46 0.98 92 0.99 46 0.98 92 0.97 46 0.98 92 0.98 46 0.98 92 0.98 46 0.98 92 0.98 46 0.98 92 0.98 46 0.98 92 0.98 46 0.98 92 0.98 46 0.98 92 0.98 46 0.98 92 0.98 47 1 93 0.99 47 1 93 0.99 47 1 93 0.99 47 1 93 0.99 47 1 93 0.99 47 1 93 0.99 47 0.98 93 0.98 47 0.98 94 0.98 47 0.98 94 0.98 47 0.98 94 0.98 47 0.98					91	0.98	46	0.96
92 1 46 0.98 92 0.98 46 0.96 92 0.99 46 0.98 92 0.99 46 0.98 92 0.97 46 0.98 92 0.98 46 0.96 92 0.98 46 0.98 92 0.98 46 0.98 92 0.98 46 0.98 92 0.98 46 0.98 92 0.98 46 0.98 92 0.98 46 0.98 93 1 47 1 93 0.99 47 1 93 0.99 47 1 93 0.99 47 1 93 0.99 47 1 93 0.99 47 0.98 94 0.98 47 0.98 95 0.98 47 0.98 96 0.98 97 0.98 47 0.98 98 0.99 47 0.98 99 0.99 47 0.98 99 0.99 47 0.98 99 0.99 47 0.98 99 0.99 47 0.98					91	0.98	46	1
92 0.98 46 0.96 92 0.98 46 0.98 92 0.99 46 0.98 92 0.99 46 0.98 92 0.97 46 0.98 92 0.98 46 0.98 92 0.98 46 0.98 92 0.98 46 0.98 92 0.98 46 0.96 92 0.98 46 0.98 92 0.98 46 0.98 93 1 47 0.98 93 0.99 47 0.98 93 0.99 47 0.98 93 0.99 47 0.98 93 0.99 47 0.98 93 0.99 47 0.98 94 0.99 47 0.98 94 0.98 47 0.98					91	0.97	46	1
92 0,98 46 0,98 92 0,99 46 0,98 92 0,97 46 0,98 92 0,97 46 0,98 92 0,98 46 0,96 92 0,98 46 0,98 92 0,98 46 0,98 92 0,98 46 0,98 92 0,98 46 0,98 92 0,98 46 0,98 93 1 47 1 93 0,99 47 1 93 0,99 47 1 93 0,99 47 1 93 0,99 47 1 93 0,98 47 0,98 93 0,98 47 0,98 93 0,98 47 0,98 93 0,98 47 0,98 93 0,98 47 0,98 93 0,98 47 0,98 93 0,98 47 0,98 93 0,98 47 0,98 94 0,99 47 0,98 94 0,99 47 0,98 94 0,99 47 0,98 94 0,99 47 0,98 94 0,99 47 0,98 94 0,99 47 0,98 94 0,99 47 0,98 94 0,99 47 0,98 94 0,99 47 0,98					92	1	46	0.98
92 0,99 46 0,98 92 0,97 46 0,98 92 0,98 46 0,96 92 0,98 46 0,98 92 0,98 46 0,98 92 0,98 46 0,98 92 0,98 46 0,98 93 1 47 1 93 0,99 47 1 93 0,99 47 1 93 0,99 47 1 93 0,99 47 1 93 0,99 47 1 93 0,99 47 1 93 0,98 47 0,98 93 0,98 47 0,98 93 0,98 47 0,98 93 0,98 47 0,98 93 0,98 47 0,98 93 0,98 47 0,98 93 0,98 47 0,98 93 0,98 47 0,98 93 0,98 47 0,98 94 0,99 47 0,98 94 0,99 47 0,98 94 0,98 47 0,98 94 0,98 47 0,98 94 0,98 47 0,98 94 0,98 47 0,98					92	0.98	46	0.96
92 0.97 46 0.98 92 0.98 46 0.96 92 0.98 46 0.98 92 0.98 46 0.98 93 1 47 0.98 93 0.99 47 1 93 0.99 47 1 93 0.99 47 1 93 0.99 47 1 93 0.99 47 1 93 0.99 47 1 93 0.99 47 1 93 0.99 47 1 93 0.99 47 1 93 0.98 47 0.98 94 0.98 47 0.98 94 0.98 47 0.98					92	0.98	46	0.98
92 0.98 46 0.96 92 0.98 46 0.98 92 0.98 46 0.98 93 1 47 1 93 0.99 47 1 93 0.99 47 1 93 0.98 47 0.98 93 0.98 47 0.98 93 0.98 47 0.98 93 0.98 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98					92	0.99	46	0.98
92 0.98 46 0.98 92 0.98 46 0.98 93 1 47 1 93 0.99 47 1 93 0.99 47 1 93 0.99 47 1 93 0.99 47 1 93 0.99 47 1 93 0.98 47 1 93 0.98 47 1 93 0.98 47 0.98 93 0.98 47 0.98 93 0.98 47 0.98 93 0.98 47 0.98 94 0.99 47 0.98 94 0.98 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98					92	0.97	46	0.98
92 0.98 46 0.98 92 0.98 46 0.98 93 1 47 1 93 0.99 47 1 93 0.99 47 1 93 0.99 47 1 93 0.99 47 1 93 0.99 47 1 93 0.98 47 1 93 0.98 47 1 93 0.98 47 0.98 93 0.98 47 0.98 93 0.98 47 0.98 93 0.98 47 0.98 94 0.99 47 0.98 94 0.98 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98					92	0.98	46	0.96
92 0.98 46 0.98 93 1 47 1 93 0.99 47 1 93 0.99 47 0.98 93 0.99 47 1 93 0.99 47 1 93 0.99 47 1 93 0.99 47 1 93 0.98 47 1 93 0.98 47 0.98 93 0.98 47 0.98 93 0.98 47 0.98 94 0.99 47 0.98 94 0.98 47 0.98 94 0.98 47 0.98 94 0.98 47 0.98 94 0.98 47 0.98 94 0.98 47 0.98 94 0.98 47 0.98 94 0.98 47 0.98 94 0.98 47 0.98 94 0.98 47 0.98 94 0.98 47 0.98 94 0.98 47 0.98								
93 1 47 1 93 0.99 47 1 93 0.99 47 0.98 93 0.99 47 1 93 0.99 47 1 93 0.98 47 1 93 0.98 47 0.98 93 0.98 47 0.98 93 0.98 47 0.98 93 0.98 47 0.98 93 0.98 47 0.98 94 0.99 47 0.98 94 0.98 47 0.98 94 0.98 47 0.98 94 0.98 47 0.98 94 0.98 47 0.98 94 0.98 47 0.98 94 0.98 47 0.98 94 0.98 47 0.98 94 0.98 47 0.98								
93 0.99 47 1 93 0.99 47 0.98 93 0.99 47 1 93 0.99 47 1 93 0.99 47 1 93 0.98 47 1 93 0.98 47 0.98 93 0.98 47 0.98 93 0.98 47 0.98 93 0.98 47 0.98 94 0.99 47 0.98 94 0.98 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98								
93 0.99 47 0.98 93 0.99 47 1 93 0.98 47 1 93 0.98 47 1 93 0.98 47 0.98 93 0.98 47 0.98 93 0.98 47 0.98 93 0.98 47 0.98 94 0.99 47 0.98 94 0.98 47 0.98 95 0.98 47 0.98 96 0.98 47 0.98 97 0.98 47 0.98 98 0.98 47 0.98 99 0.98 47 0.98 99 0.98 47 0.98 99 0.98 47 0.98 99 0.98 47 0.98 99 0.98 47 0.98 99 0.98 47 0.98								
93 0.99 47 1 93 0.98 47 1 93 0.98 47 0.98 93 0.98 47 0.98 93 0.98 47 0.98 93 0.98 47 0.98 94 0.99 47 0.98 94 0.98 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98								
93 0.98 47 1 93 0.98 47 0.98 93 0.98 47 0.98 93 0.98 47 0.98 93 0.98 47 0.98 94 0.99 47 0.98 94 0.98 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98								
93 0.98 47 0.98 93 0.98 47 0.98 93 0.98 47 0.98 93 0.98 47 0.98 94 0.99 47 0.98 94 0.98 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.98 47 0.96 94 0.98 47 0.96								
93 0.98 47 0.98 93 0.98 47 0.98 94 0.99 47 0.98 94 0.98 47 0.98 94 0.98 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.98 47 0.96 94 0.98 47 0.96 94 0.98 47 0.98								
93 0.98 47 0.98 94 0.99 47 0.98 94 0.98 47 0.98 94 0.98 47 0.98 94 0.98 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.98 47 0.96 94 0.98 47 0.96 94 0.98 47 0.98 94 0.98 47 0.98								
94 0.99 47 0.98 94 0.98 47 0.98 94 0.98 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.99 47 0.96 94 0.98 47 0.96 94 0.98 47 0.98 94 0.98 47 0.98 94 0.99 47 0.98								
94 0.98 47 0.98 94 0.98 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.98 47 0.96 94 0.98 47 0.98 94 0.98 47 0.98 94 0.99 47 0.98								
94 0.98 47 0.98 94 0.99 47 0.98 94 0.98 47 0.96 94 0.98 47 0.96 94 0.98 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98								
94 0.99 47 0.98 94 0.98 47 0.96 94 0.98 47 0.98 94 0.98 47 0.98 94 0.99 47 0.98 94 0.98 47 0.98								
94 0.98 47 0.96 94 0.98 47 0.98 94 0.99 47 0.98 94 0.99 47 0.98 94 0.98 47 0.98								
94 0.98 47 0.98 94 0.99 47 0.98 94 0.98 47 0.98							47	0.98
94 0.99 47 0.98 94 0.98 47 0.98						0.98	47	0.96
94 0.98 47 0.98					94	0.98	47	0.98
					94	0.99	47	0.98
95 1 48 098					94	0.98	47	0.98
					95	1	48	0.98
95 0.99 48 1					95	0.99	48	1
95 0.99 48 0.98					95	0.99	48	0.98
95 0.99 48 1								
95 0.99 48 0.96								
95 0.97 48 0.98								
95 0.98 48 0.98								
95 0.98 48 0.98								
96 0.99 48 0.98								
96 0.99 48 0.98					96	0.99	48	0.98

1	1	1	1		ì	İ			ı
						96	0.99	48	0.98
						96	0.99	48	0.96
						96	0.99	48	0.96
						96	0.98	48	0.96
						96	0.98	48	0.98
						96	0.97	48	0.96
						97	1	49	0.98
						97	0.99	49	1
						97	0.98	49	1
						97	0.99	49	0.98
						97	0.97	49	0.98
						97	0.97	49	1
						97	0.97	49	0.98
						97	0.99	49	0.98
						98	0.99	49	0.96
						98	0.98	49	0.98
						98 98	0.99	49	0.96
							0.99	49	0.96
						98	0.98	49	0.98
						98	0.98	49	0.98
						98	0.99	49	0.98
						98	0.99	49	0.98
						99	1	50	1
						99	0.99	50	1
						99	0.98	50	0.98
						99	0.99	50	0.96
						99	0.98	50	0.98
						99	0.97	50	1
						99	0.98	50	0.98
						99	0.98	50	0.98
						100	0.99	50	0.98
						100	0.98	50	0.98
						100	0.99	50	0.98
						100	1	50	0.98
						100	0.99	50	0.98
						100	0.98	50	0.98
						100	0.98	50	0.98
						100	0.99	50	0.96
								51	0.98
								51	0.98
								51	0.98
								51	1
								51	0.98
								51	0.98
								51	0.98
								51	0.98
								51	0.98
								51	0.98
								51	0.98
								51	0.96
								51	0.96
								51	0.96
								51	0.96
								51	0.98
								52	1
								52	0.98
								52	0.98
								52	0.98
								52	0.98
L			1	1	<u> </u>	İ	1	1	1

	1 1	1	I	I	1	l	I	I	I
								52	0.98
								52	0.98
S								52	1
								52	0.98
Section Sect								52	0.98
ST ST ST ST ST ST ST ST								52	0.98
S								52	0.98
								52	0.98
								52	0.96
S									
Signature Sign									
S3 0.98 S3									
S3									
S3									
S3 0.98 S3									
S3									
S									
S3									
S3									
S3									
S3 0.98 S3 0.98 S3 0.98 S3 0.98 S3 0.98 S3 0.98 S3 0.98 S4 1 S4 0.98 S4 0.96 S4 0.96 S4 0.96 S4 0.96 S4 0.96 S4 0.96 S4 0.96 S4 0.96 S5 0.98 S5 0.									
S3								53	0.98
S3 0.98								53	0.98
S4								53	0.96
S4								53	0.98
54 0.98 54 0.98 54 0.98 54 0.98 54 0.98 54 0.98 54 0.98 54 0.98 54 0.98 54 0.98 54 0.98 54 0.98 54 0.96 54 0.96 54 0.96 54 0.96 54 0.96 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.								54	1
S4								54	0.98
S4								54	0.98
S4									
54 0.98 54 0.98 54 0.98 54 0.98 54 0.98 54 0.98 54 0.98 54 0.96 54 0.96 54 0.96 54 0.96 54 0.96 54 0.96 55 0.98 55 0.									
54 0.98 54 0.98 54 0.98 54 0.98 54 0.98 54 0.98 54 0.96 54 0.96 54 0.96 54 0.96 54 0.96 54 0.96 55 0.98 55 0.									
54 0.98 54 0.98 54 0.98 54 0.98 54 0.98 54 0.96 54 0.96 54 0.96 54 0.96 54 0.96 54 0.96 55 0.98 55 0.									
54 0.98 54 0.98 54 0.98 54 0.98 54 0.96 54 0.96 54 0.96 54 0.96 54 0.96 55 0.98									
54 0.98 54 0.98 54 0.96 54 0.96 54 0.96 54 0.96 54 0.96 54 0.96 55 0.98 55 0.									
54 0.98 54 0.96 54 0.96 54 0.96 54 0.96 54 0.96 54 0.96 55 0.98 55 0.									
54 0.96 54 0.96 54 0.96 54 0.96 54 0.96 55 0.98									
54 0.96 54 0.96 54 0.96 54 0.96 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98									
54 0.96 54 0.96 54 0.96 54 0.96 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98									
54 0.96 54 0.96 55 0.98								54	
54 0.96 55 0.98									
55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 1 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98									
55 0.98 55 1 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 1 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98								54	0.96
55 1 1 1 2 3 35 4 55 55 55 1 55 1 55 1 55 1 55 1 2 3 4 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98								55	0.98
55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 1 55 1 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98								55	0.98
55 0.98 55 0.98 55 0.98 55 0.98 55 1 55 1 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98								55	1
55 0.98 55 0.98 55 0.98 55 0.98 55 1 55 1 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98									0.98
55 0.98 55 0.98 55 0.98 55 1 55 1 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98									0.98
55 0.98 55 1 55 1 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98 55 0.98									
55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 0.98 55 0.98									
55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 0.98 55 0.98									
55 55 55 55 55 55 55 55 55 55 55 55 55 55 0.98 55 0.98									
55 0.98 55 0.98 55 0.98 55 0.98 55 0.98									
55 0.98 55 0.98 55 0.98									
55 0.98 55 0.98									
55 0.98									
								55	0.98
55 0.98								55	0.98

l I			I	l 1	İ	I	I	I
							56	1
							56	0.98
							56	1
							56	0.98
							56	0.98
							56	0.98
							56	0.98
							56	0.97
							56	0.98
							56	0.98
							56	0.98
							56	0.97
							56	0.97
							56	0.97
							56	0.98
							56	0.98
							57	1
							57	0.98
							57	1
							57	0.98
							57	0.98
							57	0.98
							57	0.98
							57	0.98
							57	0.98
							57	0.98
							57	0.97
							57	0.98
							57	0.98
							57	0.98
							57	0.97
							57	0.98
							58	1
							58	0.98
							58	0.98
							58	0.98
							58	0.97
							58	0.98
							58	0.98
							58	0.98
							58	0.98
							58	0.98
							58	0.98
							58	0.97
							58	0.98
							58	0.98
							58	0.98
							58	0.98
							59	1
							59	1
							59	0.98
							59	0.98
							59	1
							59	0.98
							59	0.98
			I .					
1							59	11
							59	0.08
							59	0.98

l I			l I	ĺ	I	l	1
						59	0.98
						59	0.97
						59	0.97
						59	0.98
						59	0.97
						60	1
						60	1
						60	0.98
						60	1
						60	0.98
						60	0.97
						60	0.98
						60	0.97
						60	0.98
						60	0.97
						60	0.98
						60	0.98
						60	0.98
						60	0.98
						60	0.98
						60	0.98
						61	1
						61	1
						61	0.98
						61	0.98
						61	0.98
						61	0.98
						61	0.98
						61	0.98
						61	0.98
						61	1.02
						61	0.97
						61	0.98
						61	0.98
						61	0.97
						61	0.97
						61	0.98
						62	0.98
						62	1
						62	0.98
						62	0.97
						62	0.98
						62	0.98
						62	0.98
						62	0.98
						62	0.98
						62	0.98
						62	1
						62	0.97
						62	0.97
						62	0.98
						62	0.98
						62	0.97
						63	1
						63	1
						63	1
						63	0.98
1						62	1
						63	0.98

	I.	l	I	ı	1	1	I	I	I
								63	0.97
								63	0.98
								63	0.98
								63	1
								63	1
								63	0.98
								63	0.97
								63	0.98
								63	0.98
								64	0.98
								64	0.97
								64	0.97
								65	1
								65	1
65								65	0.97
65								65	0.97
65 0.98 65 0.98 65 0.98 65 0.98 65 0.98 65 0.98 65 0.98 65 0.98 65 0.98 65 0.98 65 0.98 65 0.98 65 0.98 65 0.98 65 0.98 65 0.97 66 0.99									
1.02									
65 0.98 65 0.98 65 0.98 65 0.98 65 0.97 65 0.98 66 0.99									
65 0.98 65 0.98 65 0.97 65 0.98 66 0.99									
65 0.98 65 0.97 65 0.98 65 0.97 65 0.98 66 0.99 0.99 66 0.99									
65 0.97 65 0.98 66 0.99 66 0.99 66 0.99 66 0.99 66 0.97 66 0.99 66 0.99 66 0.99 66 1 66 1 66 1 66 0.99 66 0.99 66 0.99 66 0.99 66 0.99 66 0.99 66 0.99									
65 0.98 66 0.99 66 0.99 66 0.99 66 0.99 66 0.99 66 0.99 66 0.99 66 0.99 66 0.99 66 1 66 1 66 1 66 0.99 66 1 66 0.99 66 1 66 0.99 66 1 66 0.99 66 1 66 0.99 66 1 66 0.99 66 1									
66 0.99 66 0.99 66 0.99 66 0.99 66 0.97 66 0.99 66 0.99 66 0.99 66 1 66 1 66 1 66 0.99 66 0.99 66 0.99 66 0.99 66 0.99 66 0.99 66 0.99 66 0.99 66 0.99									
66 0.99 66 0.99 66 0.99 66 0.97 66 0.99 66 0.99 66 0.99 66 1 66 1 66 1 66 1 66 0.99 66 0.99 66 0.99 66 0.99 66 0.99 66 0.99 66 0.99 66 0.99 66 0.99									
66 0.99 66 0.99 66 0.97 66 0.99 66 0.99 66 0.99 66 1 66 1 66 1 66 1 66 0.99 66 0.99 66 0.99 66 0.99 66 0.97 66 0.94 66 0.99									
66 0.99 66 0.97 66 0.99 66 0.99 66 0.99 66 1 66 1 66 1 66 1 66 0.99 66 0.99 66 0.99 66 0.97 66 0.94 66 0.99									
66 0.99 66 0.99 66 0.99 66 0.99 66 1 66 1 66 0.99 66 1 66 0.99 66 0.99 66 0.99 66 0.99 66 0.99 66 0.99						 		66	0.99
66 0.99 66 0.99 66 0.99 66 1 66 1 66 0.99 66 1 66 0.99 66 0.99 66 0.99 66 0.99 66 0.99 66 0.99								66	0.99
66 0.99 66 1 66 1 66 1 66 0.99 66 1 66 0.99 66 0.99 66 0.99 66 0.99 66 0.99 66 0.99 66 0.99								66	0.97
66 0,99 66 1 66 1 66 1 66 1 66 0,99 66 0,99 66 0,99 66 0,99 66 0,99 66 0,99								66	0.99
66 1 66 1 66 1 66 0.99 66 0.99 66 0.97 66 0.94 66 0.99								66	0.99
66 1 66 1 66 1 66 0.99 66 0.99 66 0.97 66 0.94 66 0.99								66	0.99
66 1 66 1 66 0.99 66 0.99 66 0.97 66 0.94 66 0.99									
66 1 66 0.99 66 0.99 66 0.97 66 0.94 66 0.99									
66 0.99 66 0.97 66 0.97 66 0.94 66 0.99									
66 0.99 66 0.97 66 0.94 66 0.99									
66 0.97 66 0.94 66 0.99									
66 0.94 66 0.99									
66 0.99									
67 0.99									
								67	0.99

I	I	I	 	1	I	ı	I
						67	1
						67	0.99
						67	0.99
						67	0.99
						67	0.99
						67	0.99
						67	1
						67	1
						67	1
						67	0.99
						67	0.99
						67	0.97
						67	0.99
						67	0.99
						67	0.99
						68	1
						68	1
						68	0.99
						68	0.99
						68	1
						68	0.97
						68	0.99
						68	0.99
						68	0.99
						68	1
						68	0.99
						68	0.99
						68	0.99
						68	0.99
						68	0.99
						68	0.97
						69	1
						69	0.99
						69	0.99
						69	0.99
						69	0.99
						69	0.99
						69	0.99
						69	0.99
						69	1
						69	0.99
						69	1
						69	1
						69	0.97
						69	0.99
						69	0.99
						69	0.99
						70	0.99
						70	0.99
						70	0.99
						70	0.99
						70	0.99
						70	0.99
						70	0.99
						70	0.99
						70	1
						70	0.99
						70	0.99
						70	1

l I			I	l I		l.	I	I	l
								70	0.99
								70	0.97
								70	0.92
								70	0.99
								71	1
								71	1
								71	1
								71	0.99
								71	0.99
								71	0.99
								71	0.99
								71	0.99
								71	0.95
								71	1
								71	1
								71	0.99
								71	0.97
								71	0.97
								71	0.97
								71	0.97
								72	1
								72	1
								72	1
								72	1
								72	0.99
								72	1
								72	0.99
								72	0.99
								72	0.99
								72	1.01
								72	1
								72	0.99
								72	0.99
								72	0.99
								72	0.99
								72	0.99
								73	1
								73	1
								73	0.99
								73	0.97
								73	1
								73	1
								73	0.99
								73	0.99
								73	0.99
								73	0.99
								73	0.99
								73	0.99
					-			73	0.97
								73	0.97
								73	0.99
								73	0.97
								74	1
								74	0.99
								74	0.99
								74	0.99
								74	0.97
								74 74	0.99

	l		l I	1	I	l	l.
						74	1
						74	1
						74	1
						74	0.99
						74	0.99
						74	0.99
						74	0.97
						74	0.99
						74	0.99
						75	0.99
						75	1
						75	0.97
						75	0.99
						75	1
						75	1
						75	0.99
						75	0.99
						75	1
						75	0.97
						75	1
						75	0.99
						75	0.99
						75	0.99
						75	0.99
						75	0.97
						76	1
						76	1
						76	1
						76	0.99
						76	0.99
						76	0.97
						76	0.99
						76	1
						76	0.99
						76	1
						76	0.99
						76	0.99
						76	0.97
						76	0.97
						76	0.99
						76	0.99
						77	1
						77	0.99
						77	0.99
						77	0.99
						77	0.99
						77	0.99
						77	0.99
						77	1
						77	0.99
						77	1
						77	1
						77	0.99
						77	0.99
						77	0.99
			1	İ		77	0.99
						77	0.97
						77 78 78	0.97 0.99

1 1		l	l	l I	l	I	l	l
							78	0.99
							78	0.99
							78	0.99
							78	1
							78	0.99
							78	0.99
							78	0.99
							78	0.98
							78	0.99
							78	0.99
							78	0.98
							78	0.99
							78	0.98
							78	0.99
							79	1
							79	1
							79	0.99
							79	0.99
							79	1
							79	0.99
							79	1
							79	0.99
							79	0.99
							79	0.99
							79	1
							79	0.99
							79	0.99
							79	0.98
							79	0.98
							79	0.99
							80	0.99
							80	1
							80	0.99
							80	0.99
							80	1
							80	0.99
							80	1
							80	0.99
							80	0.99
							80	0.99
							80	1
							80	0.99
							80	0.98
							80	0.99
							80	0.98
							80	0.99
							81	1
							81	1
							81	0.99
							81	0.99
							81	1
							81	0.99
							81	0.99
							81	1
							81	0.99
							81	0.99
		1	I					
							81	1 0 99
							81	0.99
							81 81	0.99 0.98 0.99

l I	l	I	l I		l	l	1
						81	0.99
						81	0.98
						81	0.99
						82	1
						82	0.99
						82	0.99
						82	0.99
						82	1
						82	0.99
						82	0.99
						82	1
						82	0.99
						82	0.99
						82	1
						82	1
						82	0.99
						82	0.98
						82	0.99
						82	0.98
						83	1
						83	1
						83	0.98
						83	1
						83	1
						83	0.99
						83	0.99
						83	0.98
						83	0.99
						83	0.99
						83	1
						83	1
						83	0.98
						83	0.99
						83	0.98
						83	0.98
						84	1
						84	1
						84	1
						84	1
						84	1
						84	0.99
						84	0.99
						84	0.99
						84	0.99
						84	1
						84	1
						84	0.99
						84	0.98
						84	0.98
						84	0.99
						84	0.99
						85	1
						85	1
						85	0.98
						85	0.98
						85	1
					1	85	0.98
							0.00
						85 85	0.99

I	I	I	 	1	I	ı	I
						85	0.99
						85	0.99
						85	0.99
						85	0.99
						85	0.99
						85	0.99
						85	0.98
						85	0.99
						86	1
						86	1
						86	0.99
						86	1
						86	0.99
						86	0.99
						86	1
						86	0.99
						86	0.99
						86	0.99
						86	1
						86	0.99
						86	0.99
						86	0.99
						86	0.99
						86	0.98
						87	1
						87	0.99
						87	1
						87	1
						87	0.99
						87	1
						87	0.99
						87	0.99
						87	0.99
						87	0.99
						87	0.99
						87	0.99
						87	0.99
						87	0.99
						87	0.98
						87	0.99
						88	0.99
						88	1
						88	1
						88	1
						88	1
						88	0.99
						88	0.99
						88	0.99
						88	0.99
						88	0.99
						88	1
						88	0.99
						88	0.99
						88	0.98
						88	0.99
						88	0.99
						89	1
						89	0.99
						89	0.99

I	I	I	 	I.	I	I	I
						89	1
						89	0.99
						89	0.99
						89	0.99
						89	0.99
						89	1
						89	0.99
						89	1
						89	0.99
						89	0.99
						89	0.99
						89	0.99
						89	0.99
						90	0.99
						90	1
						90	1
						90	0.99
						90	0.99
						90	0.99
						90	0.99
						90	0.99
						90	0.99
						90	0.99
						90	0.99
						90	0.99
						90	0.98
						90	0.99
						90	0.99
						90	0.99
						91	1
						91	1
						91	1
						91	0.99
						91	1
						91	0.99
						91	0.99
						91	0.99
						91	1
						91	0.99
						91	1
						91	0.99
						91	0.99
						91	0.99
						91	0.99
						91	0.98
						92	0.99
						92	1
						92	1
						92	0.99
						92	0.99
						92	0.99
						92	0.99
						92	0.99
						92	0.99
						92	0.99
						92	1
						92	1
						92	0.99
						92	0.98

1	ı			l I	1	l	1	l
							92	0.98
<u> </u>							92	0.98
							93	0.99
							93	1
							93	1
							93	1
							93	0.99
							93	0.99
							93	1
							93	0.99
							93	1
							93	0.99
							93	1
							93	0.99
							93	0.98
							93	0.98
							93	
+								0.99
 							93	0.99
+							94	1
 							94	1
							94	1
							94	0.99
							94	0.99
							94	0.99
							94	0.98
							94	0.99
							94	0.99
							94	0.99
							94	1
							94	0.99
							94	0.99
							94	0.98
							94	0.98
							94	0.99
							95	0.99
<u> </u>							95	
 							95	1
							95	0.99
							95	0.99
							95	0.99
							95	0.99
							95	0.99
					 		95	0.99
							95	0.99
							95	1
							95	1
							95	0.99
							95	0.99
							95	0.99
							95	0.98
							96	1
							96	1
							96	1
							96	0.99
								
							96	0.99
							96	0.98
							96	1
							96 96	0.99

I	I	I	 	I.	I	ı	I
						96	0.99
						96	1
						96	0.99
						96	0.98
						96	0.99
						96	0.99
						96	0.99
						97	1
						97	1
						97	1
						97	0.99
						97	0.99
						97	1
						97	0.99
						97	0.99
						97	0.99
						97	0.99
						97	0.99
						97	0.99
						97	0.98
						97	0.98
						97	0.98
						97	0.98
						98	1
						98	1
						98	1
						98	0.99
						98	0.99
						98	0.99
						98	0.99
						98	0.99
						98	0.99
						98	0.99
						98	0.99
						98	0.99
						98	0.98
						98	0.99
						98	0.98
						98	0.99
						99	1
						99	1
						99	1
						99	0.99
						99	0.99
						99	0.99
						99	1
						99	0.99
						99	0.99
						99	0.99
						99	1
						99	0.99
						99	0.98
						99	0.98
						99	0.98
						99	0.99
						100	1
						100	1
						100	1
						100	0.99

				100	0.99
				100	0.99
				100	0.99
				100	0.98
				100	0.99
				100	0.98
				100	0.99
				100	1
				100	0.99
				100	0.99
				100	0.98
				100	0.99

Files:

Thesis documents in PDF form.

Directories:

code->thesis_code: code for the comptorrent executable.

code->thesis_code->comptracker: code for the php tracker. The matching sql is in the thesis appendices.

code->thesis_code->comptorrents: includes a sample comptorrent file.

experiment_original_data: contains data that is too big to be fully described in the thesis document.

See comptorrent.org for more up to date information