JugglerC - Generate data for 50 dollar award, Console version
Developed in Borland C++ 2.0. Converted to Borland C++ 3.0 for Windows.
Recompiled with Borland C++ 3.1 for Windows.
and now in Visual C# .Net

The program JugglerC is an upgrade to the program JugglerW that was distributed in file JUGGLE31.ZIP in 1992.

Juggler Sequence defined in Dr. Clifford A. Pickover's new book "Computers and the Imagination." This book has scores of educational and recreational experiments that can be done on a personal computer.

At the end of Appendix C in this book you will find:

". . . An award of 50 dollars is offered by the publisher for a printout of the largest Juggler number computed by readers. The award will be given on or about September, 1993, and the sequence will also be published in the Juggernaut. Currently, the largest juggler number computed is a 45,391-digit giant for the starting number 30817. It was computed by Harry J. Smith using his own software package to perform multiple precision integer arithmetic. His package is written in the object-oriented programming language Turbo Pascal 5.5 by Borland International, Inc. His juggler package is a subset of his super-precision calculator software which computes transcendental functions to thousands of decimal places. Write him to obtain the software: Harry J. Smith, 19628 Via Monte Drive, Saratoga, CA 95070. . . ."

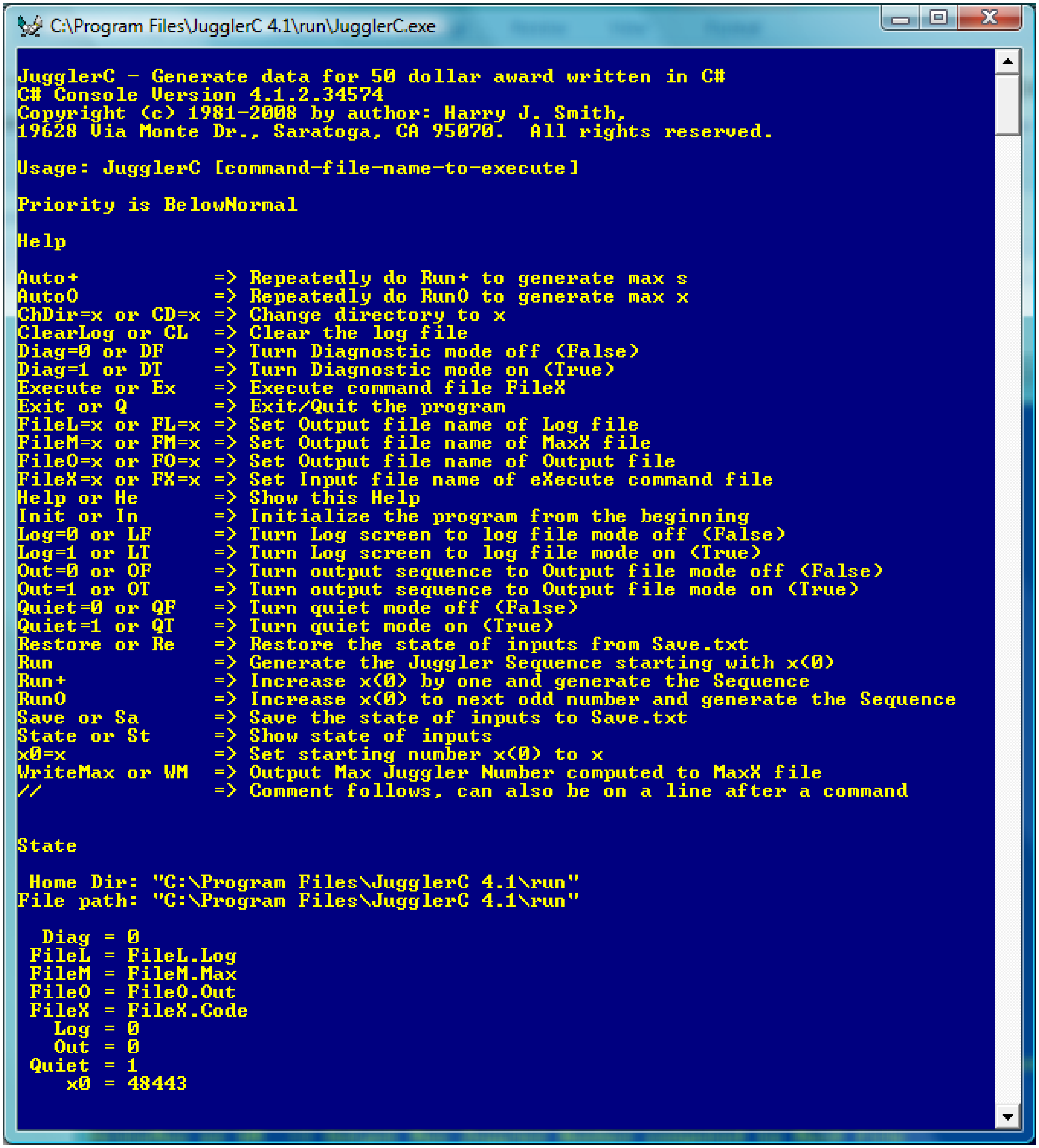
Juggler Sequence also defined in Nov 1990 issue of Algorithm in PERSONAL PROGRAMS by Clifford A. Pickover.

Pickover's definition of a juggler sequence:

```
input positive integer, x
repeat
  if x is even
    then x <-- [x^(1/2)]
    else x <-- [x^(3/2)]
until x=1
```

Running the program without arguments -

When the program JugglerC is started with no command line arguments, a console window is displayed.



```
JugglerC - Generate data for 50 dollar award written in C#
C# Console Version 4.1.2.34574
Copyright (c) 1981-2008 by author: Harry J. Smith,
19628 Via Monte Dr., Saratoga, CA 95070.  All rights reserved.

Usage: JugglerC [command-file-name-to-execute]

Priority is BelowNormal

Help

Auto+           => Repeatedly do Run+ to generate max s
AutoO           => Repeatedly do RunO to generate max x
ChDir=x or CD=x => Change directory to x
ClearLog or CL  => Clear the log file
Diag=0 or DF    => Turn Diagnostic mode off (False)
Diag=1 or DT    => Turn Diagnostic mode on (True)
Execute or Ex   => Execute command file FileX
Exit or Q       => Exit/Quit the program
FileL=x or FL=x => Set Output file name of Log file
FileM=x or FM=x => Set Output file name of MaxX file
FileO=x or FO=x => Set Output file name of Output file
FileX=x or FX=x => Set Input file name of eXecute command file
Help or He      => Show this Help
Init or In      => Initialize the program from the beginning
Log=0 or LF     => Turn Log screen to log file mode off (False)
Log=1 or LT     => Turn Log screen to log file mode on (True)
Out=0 or OF     => Turn output sequence to Output file mode off (False)
Out=1 or OT     => Turn output sequence to Output file mode on (True)
Quiet=0 or QF   => Turn quiet mode off (False)
Quiet=1 or QT   => Turn quiet mode on (True)
Restore or Re   => Restore the state of inputs from Save.txt
Run             => Generate the Juggler Sequence starting with x(0)
Run+            => Increase x(0) by one and generate the Sequence
RunO            => Increase x(0) to next odd number and generate the Sequence
Save or Sa      => Save the state of inputs to Save.txt
State or St     => Show state of inputs
x0=x            => Set starting number x(0) to x
WriteMax or WM  => Output Max Juggler Number computed to MaxX file
//              => Comment follows, can also be on a line after a command


State

 Home Dir: "C:\Program Files\JugglerC 4.1\run"
File path: "C:\Program Files\JugglerC 4.1\run"

  Diag = 0
 FileL = FileL.Log
 FileM = FileM.Max
 FileO = FileO.Out
 FileX = FileX.Code
   Log = 0
   Out = 0
 Quiet = 1
    x0 = 48443
```

The window is scrollable and resizable. It contains a listing of the help file JugglerCHelp.txt that contains the list of commands that the program can execute. It also shows that the State initial commands have been executed.

```
JugglerC - Generate data for 50 dollar award written in C#
C# Console Version 4.1.2.34574
Copyright (c) 1981-2008 by author: Harry J. Smith,
19628 Via Monte Dr., Saratoga, CA 95070.  All rights reserved.

Usage: JugglerC [command-file-name-to-execute]
```

```
Priority is BelowNormal

Help

Auto+            => Repeatedly do Run+ to generate max s
AutoO            => Repeatedly do RunO to generate max x
ChDir=x or CD=x => Change directory to x
ClearLog or CL  => Clear the log file
Diag=0 or DF    => Turn Diagnostic mode off (False)
Diag=1 or DT    => Turn Diagnostic mode on (True)
Execute or Ex   => Execute command file FileX
Exit or Q       => Exit/Quit the program
FileL=x or FL=x => Set Output file name of Log file
FileM=x or FM=x => Set Output file name of MaxX file
FileO=x or FO=x => Set Output file name of Output file
FileX=x or FX=x => Set Input file name of eXecute command file
Help or He      => Show this Help
Init or In      => Initialize the program from the beginning
Log=0 or LF     => Turn Log screen to log file mode off (False)
Log=1 or LT     => Turn Log screen to log file mode on (True)
Out=0 or OF     => Turn output sequence to Output file mode off (False)
Out=1 or OT     => Turn output sequence to Output file mode on (True)
Quiet=0 or QF   => Turn quiet mode off (False)
Quiet=1 or QT   => Turn quiet mode on (True)
Restore or Re   => Restore the state of inputs from Save.txt
Run             => Generate the Juggler Sequence starting with x(0)
Run+            => Increase x(0) by one and generate the Sequence
RunO            => Increase x(0) to next odd number and generate the Sequence
Save or Sa      => Save the state of inputs to Save.txt
State or St     => Show state of inputs
x0=x            => Set starting number x(0) to x
WriteMax or WM  => Output Max Juggler Number computed to MaxX file
//              => Comment follows, can also be on a line after a command


State

 Home Dir: "C:\ProgramD\VC#\Test\JugglerC\bin\Debug"
File path: "C:\ProgramD\VC#\Test\JugglerC\bin\Debug"

  Diag = 0
 FileL = FileL.Log
 FileM = FileM.Max
 FileO = FileO.Out
 FileX = FileX.Code
   Log = 0
   Out = 0
 Quiet = 1
    x0 = 48443
------------------------------------------------------------------------------
```

The commands are -

Auto+ => Repeatedly do Run+ to generate max s:

This does a Run command and then repeatedly does a Run+ command until aborted by the operator.

AutoO => Repeatedly do RunO to generate max x:

This does a Run command and then repeatedly does a RunO command until aborted by the operator. An exception to this is made if x0 is initially 1. In this case 2 will be included in the consecutive x0s, 1, 2, 3, 5, 7, ...

ChDir=x or CD=x => Change directory to x:

Changes the directory used for commands ClearLog, Execute, FileL=x, FileM=x, FileO=x, FileX=x, Help, Restore, Save, and WriteMax. FileR.Sum and FileS.Sum are also written in this current directory. The original directory when the program starts is called the Home directory and is used for the Execute, Help, and Restore commands if the corresponding file is not in the current directory.

ClearLog or CL => Clear the log file:

If the log file is open, the file is closed and reopened. If it is currently closed, it opened and then closed. In either case it is cleared. Initially the log file name is FileL.Log.

The cleared log file will have up to five lines of data like:

```
Log file "FileL.Log" Cleared 08/20/2008 12:50:10 PM
JugglerC - Generate data for 50 dollar award written in C#
C# Console Version 4.1.x.xxxxx
Run on: Harry's Intel 3 GHz Pentium 4 - Dell XPS Gen 2 - Windows XP Pro SP2
DGV4T641 by Harry Smith
```

The fourth line is generated by having something like:

```
SET SYSTEM=Harry's Intel 3 GHz Pentium 4 - Dell DGV4T641 - Windows XP Pro SP2
```

in your AutoExec.Bat file. This does not work for Windows Vista. In this case you get something like:

```
Log file "FileL.Log" Cleared 08/20/2008 11:10:00 AM
JugglerC - Generate data for 50 dollar award written in C#
C# Console Version 4.1.x.xxxxx
Run on: DG7JWRF1 by Harry Smith
```

Diag=0 or DF => Turn Diagnostic mode off (False):

The diagnostic mode can give some extra output to see what’s going on.

Diag=1 or DT => Turn Diagnostic mode on (True).

Execute or Ex => Execute command file FileX:

This reads in the file specified in the FileX=x command and executes each line as a JugglerC command. Blank lines cause a blank line to be printed. Lines starting with // or some spaces and // are printed followed by a blank line.

Exit or Q => Exit/Quit the program:

This terminates the execution of the program.

FileL=x or FL=x => Set Output file name of Log file:

Here x specifies the name of the log file that logs all of screen output. For all commands with a file name, if a file name does not contain a period ".", an extension is added. Like .Log, .Max, .Out, or .Code.

FileM=x or FM=x => Set Output file name of MaxX file:

Here x specifies the name of the file to receive the Max Juggler Number computed since the program was initialized.

FileO=x or FO=x => Set Output file name of Output file:

Here x specifies the name of the file to receive a listing of all juggler numbers in the last juggler sequence computed.

FileX=x or FX=x => Set Input file name of eXecute command file:

Here x specifies the name of the file that will be read by the Execute command.

Help or He => Show this Help:

This reads in the file JugglerCHelp.txt and displays its contents.

Init or In => Initialize the program from the beginning:

This restarts the program as if it had been terminates and restarted without command line arguments.

Log=0 or LF => Turn Log screen to log file mode off (False):

When logging mode is on, all output to the screen is logged to a disk file. See the FileL=x command for specifying a file name for this purpose.

Log=1 or LT => Turn Log screen to log file mode on (True):

When logging mode is on, all output to the screen is logged to a disk file. See the FileL=x command for specifying a file name for this purpose.

Out=0 or OF => Turn output sequence to Output file mode off (False):

When this mode is on, the output file receives a listing of all juggler numbers in the last juggler sequence computed. See the FileO=x command.

Out=1 or OT => Turn output sequence to Output file mode on (True):

When this mode is on, the output file receives a listing of all juggler numbers in the last juggler sequence computed. See the FileO=x command.

Quiet=0 or QF => Turn quiet mode off (False):

When the quiet mode is off, all of the status messages are displayed.

Quiet=1 or QT => Turn quiet mode on (True):

When the quiet mode is on, some of the status messages are not displayed.

Restore or Re => Restore the state of inputs from Save.Code:

This reads in the file Save.Code and executes each line as a JugglerC command. Normally the Save.Code has been written by the Save command.

Run => Generate the Juggler Sequence starting with x(0):

x(0) is used as the starting value and a juggler sequence is generated.

Run+ => Increase x(0) by one and generate the Sequence:

x(0) is increased by one, then used as the starting value and a juggler sequence is generated.

RunO => Increase x(0) to next odd number and generate the Sequence:

x(0) is increased to the next odd number, then used as the starting value and a juggler sequence is generated.

Save or Sa => Save the state of inputs to Save.Code:

This writes the Save.Code file containing the commands needed to restore JugglerC's state to its current value. An example of Save.Code contents is:

```
ChDir = C:\Program Files\JugglerC 4.1\run
Diag = 0
FileL = FileL.Log
FileM = FileM.Max
FileO = FileO.Out
FileX = FileX.Code
Log = 0
Out = 0
Quiet = 1
x0 = 1
```

State or St => Show state of inputs:

This shows the home directory and the state of the commands that would be saves by the Save command. For example:

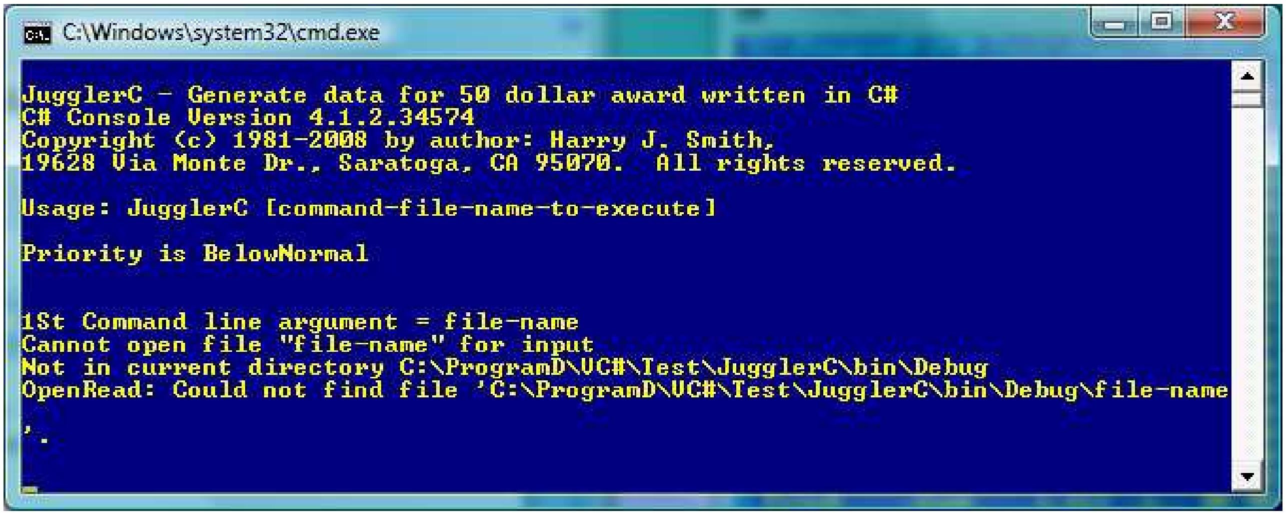
```
 Home Dir: "C:\Program Files\JugglerC 4.1\run"
File path: "C:\Program Files\JugglerC 4.1\run"

  Diag = 0
 FileL = FileL.Log
 FileM = FileM.Max
 FileO = FileO.Out
 FileX = FileX.Code
   Log = 0
   Out = 0
 Quiet = 1
    x0 = 48443
```

// => Comment follows, can also be on a line after a command:

If // starts in column one or is preceded by blanks, it shows on the screen but is considered a NoOp command. If the // is preceded by a command on the same line, the command is executed.

Running the program with arguments -

When the program JugglerC is started with command line arguments, only the first argument is used. This first argument is taken as the name of a file to be executed and a console window is displayed.



```
JugglerC - Generate data for 50 dollar award written in C#
C# Console Version 4.1.2.34574
Copyright (c) 1981-2008 by author: Harry J. Smith,
19628 Via Monte Dr., Saratoga, CA 95070.  All rights reserved.

Usage: JugglerC [command-file-name-to-execute]

Priority is BelowNormal


1St Command line argument = file-name
Cannot open file "file-name" for input
Not in current directory C:\ProgramD\VC#\Test\JugglerC\bin\Debug
OpenRead: Could not find file 'C:\ProgramD\VC#\Test\JugglerC\bin\Debug\file-name
'.
```

```
JugglerC - Generate data for 50 dollar award written in C#
C# Console Version 4.1.2.34574
Copyright (c) 1981-2008 by author: Harry J. Smith,
19628 Via Monte Dr., Saratoga, CA 95070.  All rights reserved.

Usage: JugglerC [command-file-name-to-execute]

Priority is BelowNormal
```

```
1St Command line argument = file-name
Cannot open file "file-name" for input
Not in current directory C:\ProgramD\VC#\Test\JugglerC\bin\Debug
OpenRead: Could not find file 'C:\ProgramD\VC#\Test\JugglerC\bin\Debug\file-name

'.
```

The program is initialized; FileX is set to the first command line argument and then executed by the Execute command.

Generating a sequence -

To generate a Juggler sequence, start JugglerC and execute the command

```
Run
```

If the defaults are not what you want, you can change them.

```
x0=x
FileL=x
FileM=x
FileO=x
Log=1
Out=1
Run
```

The x in the File commands can specify a full path like
C:\My Documents\JugglerOut\FileM1.Max.
Or simply a file name like FileM1.Max.
Or just FileM1 and the .Max will be added.
In the latter case, the file will be written to the same folder as the program's .exe file.

Screen messages -

Error messages are:

```
      Cannot open file "{filename}" for input
      Cannot open file "{filename}" for output
      File write error {error-message}
      I do not understand '{command}'
      Numbers too large for FHT multiply, digits = {digits} > 2^26 = 67,108,864
      Not in current directory "{directory}"
      Not in home directory "{directory}"
      OpenAppend: {error-message}
      OpenRead: {error-message}
      OpenWrite: {error-message}
      Path not found!
      Run aborted due to error in calculations
```

Other informational messages:

```
      1St Command line argument = {argument}
      D  Down started  i = {step}  N = {digits}   T = {running time}  DT =
{delta time} sec.
      Directory changed to "{directory}"
      Directory name = “{directory}”
      Directory not changed “{directory}”
      File “{filename}” opened for reading
      File “{filename}” opened for writing
      File “{filename}” verified and closed
      Full name = “{directory}\{filename}”
      Home directory is “{directory}”
      Juggler Sequence ended at {date} {time}
      Juggler Sequence starting at {date} {time} with x(0) = {x(0)}
      Length = {length}, Steps = {steps}
      Log file "{filename}" Cleared {date} {time}
      Log file "{filename}" Closed {date} {time}
      Log file "{filename}" Opened for Append {date} {time}
      Max steps = {steps} for x0 = {x(0)}
      N = number of digits in x, T = running time, DT = delta time
      Priority is BelowNormal
      Save file “Save.Code” updated
      SqRt ended     T = {running time}  DT = {delta time} sec.
      To abort calculations, Press A <Enter> or Ctrl-C
      U  Up started    i = {step}  N = {digits}   T = {running time}  DT = {delta time} sec.
      Usage: JugglerC [command-file-name-to-execute]
      x * x^2 ended  T = {running time}  DT = {delta time} sec.
      x({step}) max with {digits} digits for x(0) = {x(0)}
      x^2 ended      T = {running time}  DT = {delta time} sec.
```

Program distribution –

This program and the older JUGGLE31.ZIP are available at my website:

http://www.geocities.com/hjsmithh/

in the **Files to Download** section

http://www.geocities.com/hjsmithh/download.html#JugglerW

When you install the program using the distribution file JugglerC41?.exe or JugglerC41?.zip, a folder is created with 4 subfolders:

```
├── JugglerC 4.1
│   ├──doc
│   ├──run
│   ├──src
│   └──test
```

The main folder has two files sseexec.dat and SSEun.dat. These are needed to be able to uninstall the program

The doc subfolder has the following files:

```
     !.txt      (Information on version)
     Fixes.txt (A list of features and fixes added to each version)
     JugglerC.doc (This documentation in a Microsoft Word file)
     JugglerC.txt (Plain ASCII text copy of the JugglerC.doc file without graphics)
```

The run subfolder has the following files:

```
      !.txt
      JugglerC.exe
      JugglerCHelp.txt
      Save.Code
```

The .exe file is executed from there.

The src subfolder has all the source files needed for development:

```
      !.txt
      App.ico
      AssemblyInfo.cs
      Common.cs
      COPYING.txt
      FHTMult.cs
      Juggler.cs
      JugglerC.csproj
      JugglerC.sln
      JugglerC.suo
      JugglerC.csproj.user
      MultiID.cs
      XIMult.cs
```

The test subfolder has the files I use for testing the program. They are:

```
      !.txt
      Error6.txt
      FileL.Log
      FileL.Log1
      FileM.Max
      FileO.Out
      FileO.Out1
      FileO.Out2
      FileO.Out3
      FileO.Out4
      FileO.Out5
      FileR.Sum
      FileR.Sum6
      FileS.Sum
      FileS.Sum6
      FileX.Bat
      FileX.Code
      FileX.Log
      JugglerCHelp.txt
      Run.Bat
      Run.Code
      Run111.Bat
      Run111.Code
      Save.Code
      Test.Bat
      TestArgs.Bat
      WhatForTst.txt
      X.Bat
      X.Code
      X.Log
```

The program can generate the following files:

```
      FileL.Log
      FileM.Max - Max Juggler Number computed
      FileO.Out - Output file
      FileR.Sum – Summery of all Max Juggler Number found
      FileS.Sum - Summery of all Max steps found
      Save.Code - Saved state
```

An example of FileR.Sum:

```
-----------------------------------
x(0) max with 1 digits for x(0) = 1
x(0) max with 1 digits for x(0) = 2
x(3) max with 2 digits for x(0) = 3
x(2) max with 3 digits for x(0) = 9
x(3) max with 5 digits for x(0) = 25
x(8) max with 14 digits for x(0) = 37
x(9) max with 27 digits for x(0) = 113
x(17) max with 82 digits for x(0) = 173
x(47) max with 271 digits for x(0) = 193
x(32) max with 5929 digits for x(0) = 2183
x(54) max with 8201 digits for x(0) = 11229
x(25) max with 11723 digits for x(0) = 15065
x(43) max with 23889 digits for x(0) = 15845
x(39) max with 45391 digits for x(0) = 30817
x(60) max with 972463 digits for x(0) = 48443
x(148) max with 1909410 digits for x(0) = 2,75485
x(99) max with 1952329 digits for x(0) = 12,67909
x(89) max with 2855584 digits for x(0) = 22,64915
x(67) max with 7996276 digits for x(0) = 58,12827
x(90) max with 41564193 digits for x(0) = 71,10201  (not final)
```

An example of FileS.Sum:

```
-----------------------------------
Max steps = 0 for x0 = 1
Max steps = 1 for x0 = 2
Max steps = 6 for x0 = 3
Max steps = 7 for x0 = 9
Max steps = 9 for x0 = 19
Max steps = 11 for x0 = 25
Max steps = 17 for x0 = 37
Max steps = 19 for x0 = 77
Max steps = 43 for x0 = 163
Max steps = 73 for x0 = 193
Max steps = 75 for x0 = 1119
Max steps = 80 for x0 = 1155
Max steps = 88 for x0 = 4065
Max steps = 96 for x0 = 4229
Max steps = 107 for x0 = 4649
Max steps = 131 for x0 = 7847
Max steps = 166 for x0 = 13325
Max steps = 193 for x0 = 34175
Max steps = 201 for x0 = 59739
Max steps = 258 for x0 = 78901
Max steps = 263 for x0 = 6,36731
Max steps = 268 for x0 = 11,22603
Max steps = 271 for x0 = 13,01535
Max steps = 298 for x0 = 22,63913
Max steps = 335 for x0 = 59,47165
```

The end -

Report any errors by sending me a letter, an e-mail or call me at my home phone.

-Harry

Harry J. Smith
19628 Via Monte Dr.
Saratoga, CA 95070-4522, USA

Home Phone: 1 408 741-0406
E-mail: hjsmithh@sbcglobal.net
Website: http://www.geocities.com/hjsmithh/