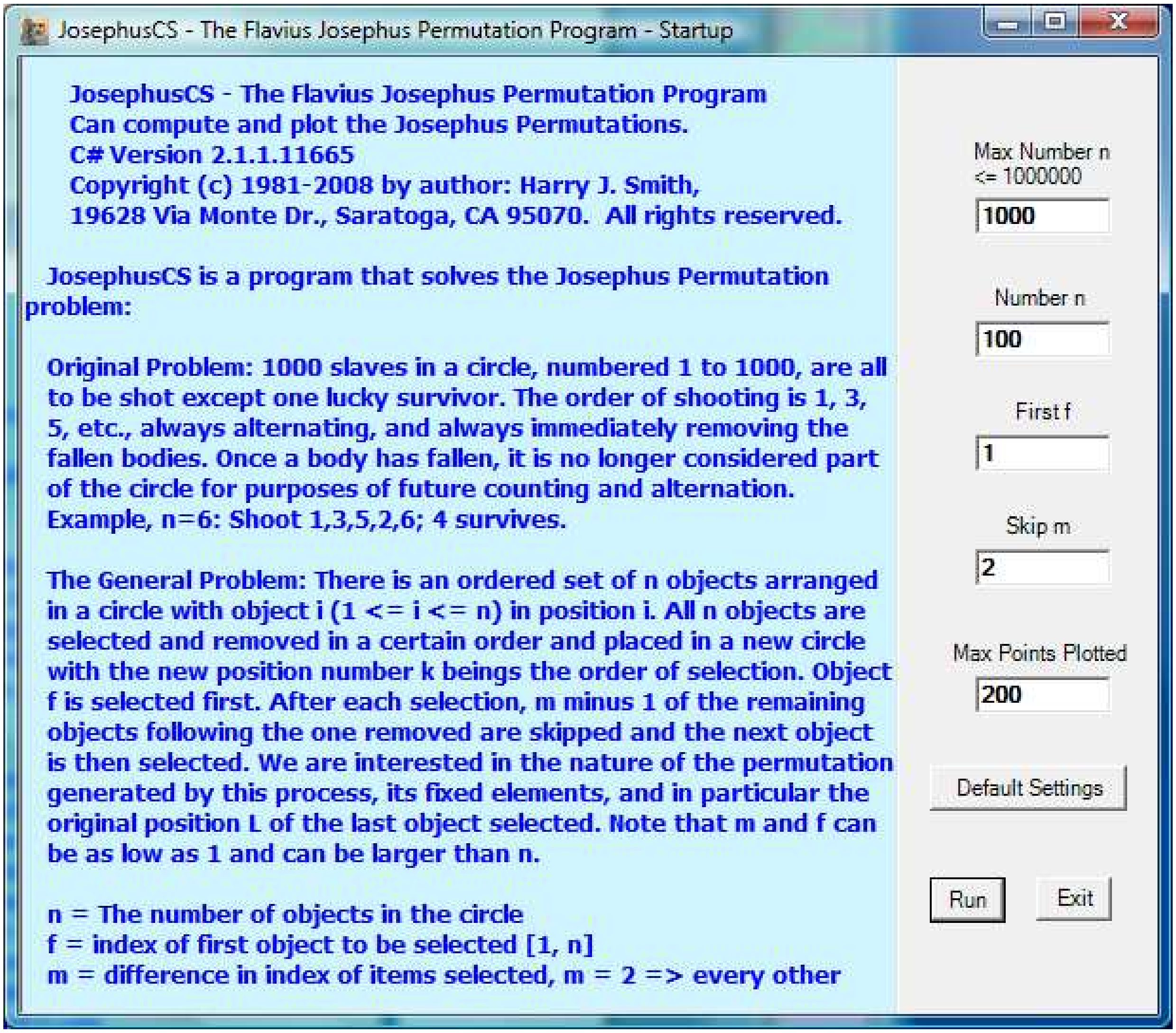
This document applies to the C# Version **2.1.1.11665** of JosephusCS,
Copyright (c) 1981-2008 by author: Harry J. Smith, Saratoga, CA.

Introduction -

When you execute the program JosephusCS.exe it responds by displaying the Startup form:



JosephusCS - The Flavius Josephus Permutation Program - Startup

JosephusCS - The Flavius Josephus Permutation Program
Can compute and plot the Josephus Permutations.
C# Version 2.1.1.11665
Copyright (c) 1981-2008 by author: Harry J. Smith,
19628 Via Monte Dr., Saratoga, CA 95070. All rights reserved.

JosephusCS is a program that solves the Josephus Permutation problem:

Original Problem: 1000 slaves in a circle, numbered 1 to 1000, are all to be shot except one lucky survivor. The order of shooting is 1, 3, 5, etc., always alternating, and always immediately removing the fallen bodies. Once a body has fallen, it is no longer considered part of the circle for purposes of future counting and alternation. Example, n=6: Shoot 1,3,5,2,6; 4 survives.

The General Problem: There is an ordered set of n objects arranged in a circle with object i (1 <= i <= n) in position i. All n objects are selected and removed in a certain order and placed in a new circle with the new position number k beings the order of selection. Object f is selected first. After each selection, m minus 1 of the remaining objects following the one removed are skipped and the next object is then selected. We are interested in the nature of the permutation generated by this process, its fixed elements, and in particular the original position L of the last object selected. Note that m and f can be as low as 1 and can be larger than n.

n = The number of objects in the circle
f = index of first object to be selected [1, n]
m = difference in index of items selected, m = 2 => every other

Max Number n <= 1000000: 1000

Number n: 100

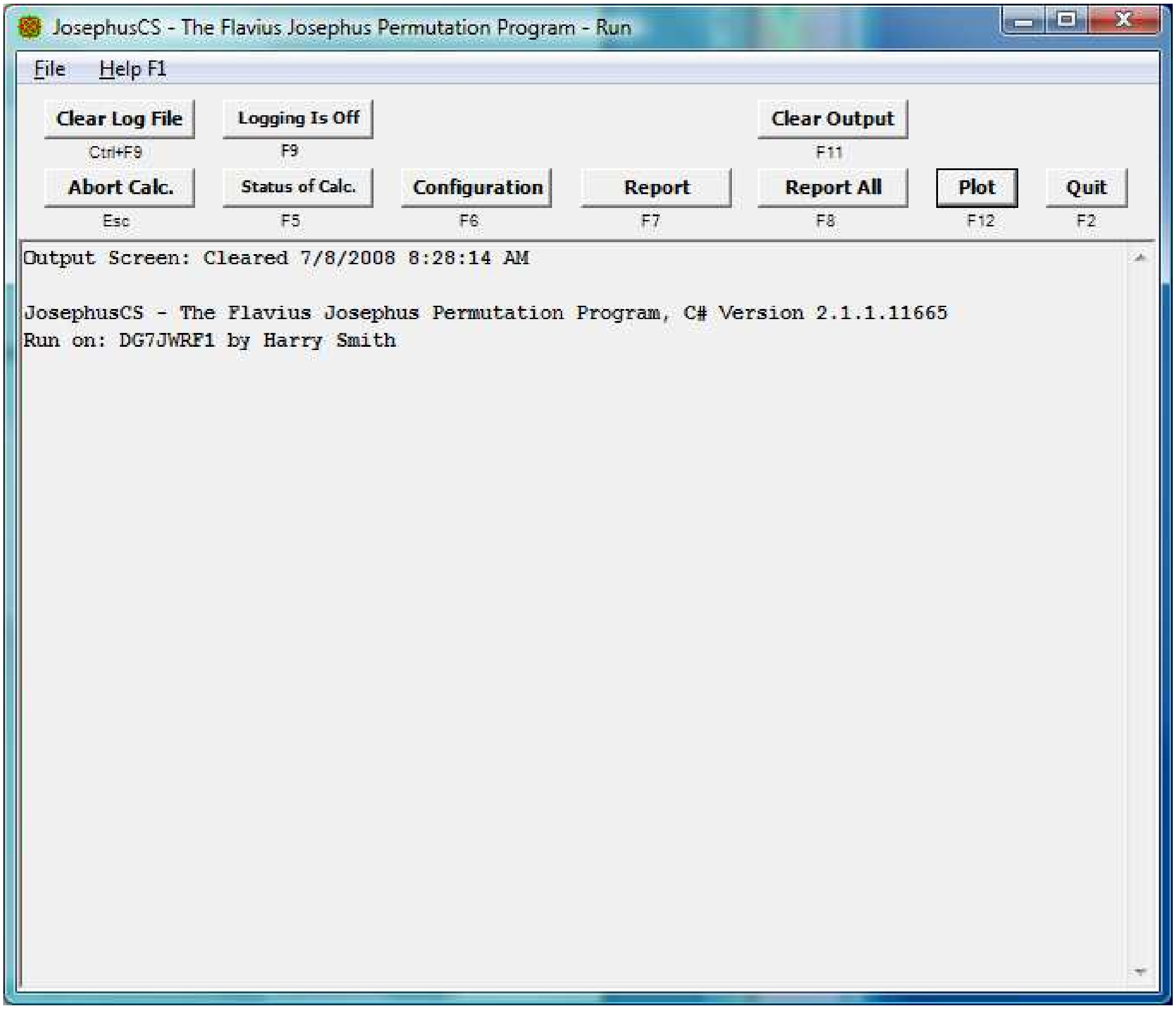
First f: 1

Skip m: 2

Max Points Plotted: 200

Just click the Run button and the Run form is displayed.

Run form -



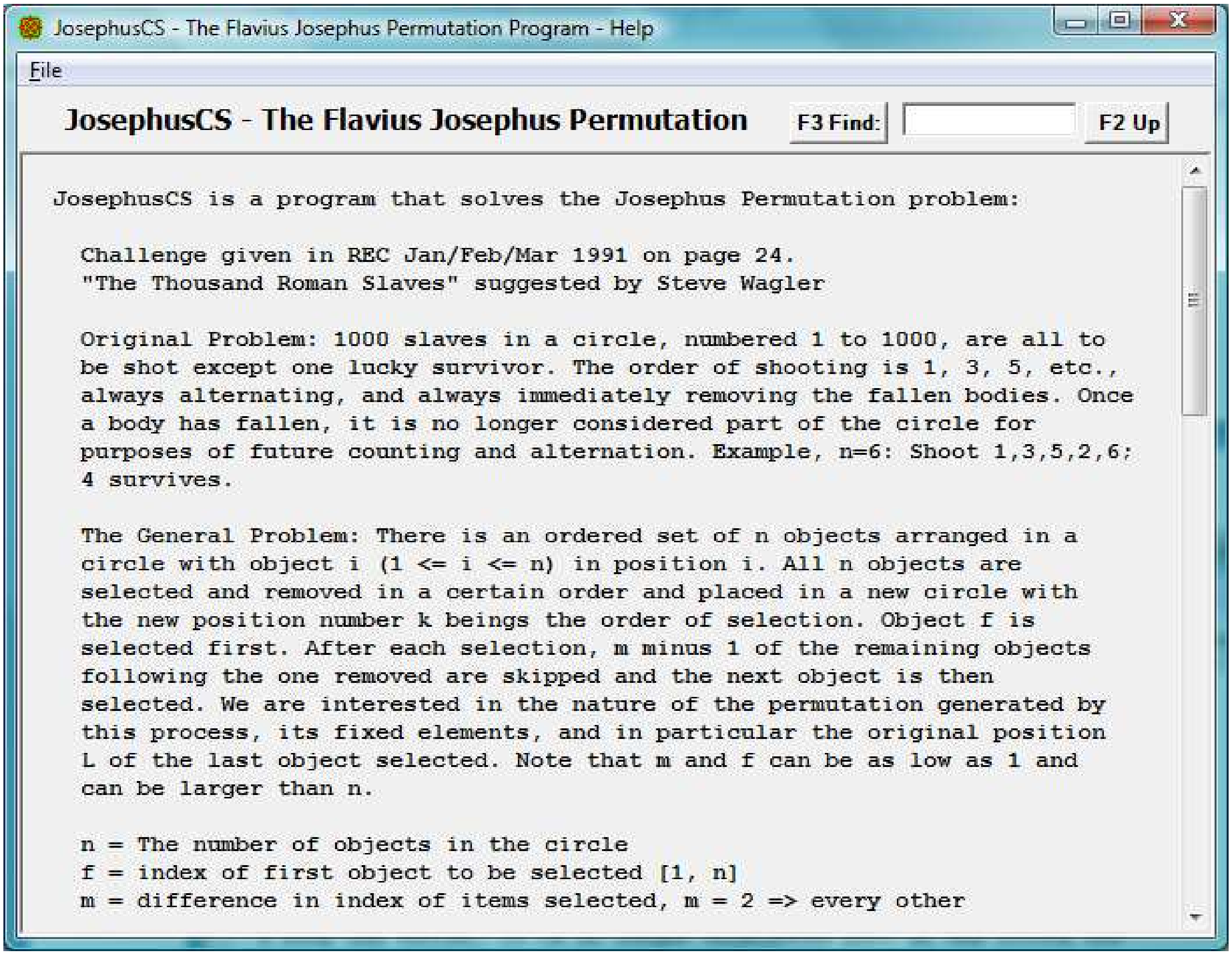
```
Output Screen: Cleared 7/8/2008 8:28:14 AM

JosephusCS - The Flavius Josephus Permutation Program, C# Version 2.1.1.11665
Run on: DG7JWRF1 by Harry Smith
```

The Run form has one large text box for text output.

If the program is executed from the DOS prompt with one or more arguments, the arguments are ignored.

Pressing the Help F1 key causes the Help form to be displayed:


JosephusCS - The Flavius Josephus Permutation Program - Help
File
JosephusCS - The Flavius Josephus Permutation
F3 Find:
F2 Up

JosephusCS is a program that solves the Josephus Permutation problem:

Challenge given in REC Jan/Feb/Mar 1991 on page 24.
"The Thousand Roman Slaves" suggested by Steve Wagler

Original Problem: 1000 slaves in a circle, numbered 1 to 1000, are all to be shot except one lucky survivor. The order of shooting is 1, 3, 5, etc., always alternating, and always immediately removing the fallen bodies. Once a body has fallen, it is no longer considered part of the circle for purposes of future counting and alternation. Example, n=6: Shoot 1,3,5,2,6; 4 survives.

The General Problem: There is an ordered set of n objects arranged in a circle with object i (1 <= i <= n) in position i. All n objects are selected and removed in a certain order and placed in a new circle with the new position number k beings the order of selection. Object f is selected first. After each selection, m minus 1 of the remaining objects following the one removed are skipped and the next object is then selected. We are interested in the nature of the permutation generated by this process, its fixed elements, and in particular the original position L of the last object selected. Note that m and f can be as low as 1 and can be larger than n.

n = The number of objects in the circle
f = index of first object to be selected [1, n]
m = difference in index of items selected, m = 2 => every other

The Help form is scrollable and resizable. It helps explains what the program does.

JosephusCS is a program that solves the Josephus Permutation problem:

Challenge given in REC Jan/Feb/Mar 1991 on page 24.
"The Thousand Roman Slaves" suggested by Steve Wagler

Original Problem: 1000 slaves in a circle, numbered 1 to 1000, are all to be shot except one lucky survivor. The order of shooting is 1, 3, 5, etc., always alternating, and always immediately removing the fallen bodies. Once a body has fallen, it is no longer considered part of the circle for purposes of future counting and alternation. Example, n=6: Shoot 1,3,5,2,6; 4 survives.

The General Problem: There is an ordered set of n objects arranged in a circle with object i (1 <= i <= n) in position i. All n objects are selected and removed in a certain order and placed in a new circle with the new position number k beings the order of selection. Object f is selected first. After each selection, m minus 1 of the remaining objects following the one removed are skipped and the next object is then selected. We are interested in the nature of the permutation generated by this process, its fixed elements, and in particular the original position L of the last object selected. Note that m and f can be as low as 1 and can be larger than n.

n = The number of objects in the circle
f = index of first object to be selected [1, n]
m = difference in index of items selected, m = 2 => every other

"Max Pointed Plotted" can be set to a value from 1 to 1000000. The nominal starting value is 200 and it can only be changed on the Startup form. Large values of this parameter can cause the Plot form to take a very long time before the plot appears and the program may have to be terminated manually by clicking the X in the upper right hand corner of the Plot form or the Run form. If n is greater than Max Points Plotted, only the first Max Points Plotted points will be plotted.

See Knuth, The Art of Computer Programming, Vol. 1, Pages 158-159, 181, 516, 521 and Vol. 3, Pages 18-19, 579.

```
--------------------------------------------------------------------------------

            +------------Function Keys on Run form-------------+
            |   F1  => Display Help form                       |
            |   F2  => Quit and go back to the Startup form    |
            |   F5  => Get Status of calculation               |
            |   F6  => Display Configuration form              |
            |   F7  => Outputs a simple permutation report     |
            |   F8  => Outputs a detailed permutation report   |
            |   F9  => Toggle Logging to Log file on/off       |
            |   F11 => Clear output text box                   |
            |   F12 => Brings up the Plot form and plots       |
            |   Ctrl+F9 => Clear Log File                      |
            |   ESC => Interrupt/Abort a long calculation      |
            +--------------------------------------------------+
--------------------------------------------------------------------------------
```

Example output:

```
--------------------
  n = 1000,  f = 1,  m = 2:  Object 976 was selected last!
  There is 1 fixed element:  1.
--------------------
  n = 1000,  f = 32767 => 767,  m = 32767:  Object 481 was selected last!
  There are 5 fixed elements:  41, 178, 619, 710, 718.
--------------------
  n = 1000,  f = 1,  m = 1:  Object 1000 was selected last!
  There are 1000 fixed elements:  1, 2, 3, 4, 5, 6, 7, 8, 9, 10, ... .
--------------------
  Execution order, when object i was selected:
  5 4 6 1 3 8 7 2

  Order of execution, object # selected on k'th selection:
  4 8 5 2 1 3 7 6

  The resulting permutation expressed in cyclic notation:
  (1, 5, 3, 6, 8, 2, 4)(7)

  The fixed element in this Josephus permutation is:
  7

  n = 8,  f = 4,  m = 4:  Object 6 was selected last!
  There is 1 fixed element:  7.
--------------------
```

-Harry

Function keys -

The function keys described here or used on the Run form.

F1 => Display Help form.

F2 => Quit the Run form and go back to the Startup form:

If a calculation is running, indicated by the word **Running** displayed on the Run form, when F2 is pressed, it is treated as an Abort Calc. request, with the message:

```
When **Running**, select Quit twice to quit
```

F5 => Get Status of calculation:

Press the F5 key while a computation is being performed and a message like

```
Orbits: <xx.xx> Percent done, Memory in use = <xxxxxxx>, DT = <xx.xx> sec.
```

will be displayed in the output text box. This is the same as the “Status of Calc.” command button.

F6 => Display Configuration form:

Press the F6 key and the Configuration form will appear. F6 is the same as the “Configuration” command button.

F7 => Outputs a simple permutation report:

Press the F7 key and a simple permutation report will be output. For example:

```
n = 100,  m = 2,  f = 1:  Object 72 was selected last!
There is 1 fixed element:  1.
```

F7 is the same as the “Report” command button.

F8 => Outputs a detailed permutation report:

Press the F8 key and a detailed permutation report will be output. For example:

```
Execution order, when object i was selected:
1 51 2 76 3 52 4 99 5 53 6 77 7 54 8 89 9 55 10 78 11 56 12 95 13 57 14 79 15 58
 16 90 17 59 18 80 19 60 20 98 21 61 22 81 23 62 24 91 25 63 26 82 27 64 28 96 2
9 65 30 83 31 66 32 92 33 67 34 84 35 68 36 100 37 69 38 85 39 70 40 93 41 71 42
 86 43 72 44 97 45 73 46 87 47 74 48 94 49 75 50 88

Order of execution, object # selected on k'th selection:
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51 53 55 5
7 59 61 63 65 67 69 71 73 75 77 79 81 83 85 87 89 91 93 95 97 99 2 6 10 14 18 22
 26 30 34 38 42 46 50 54 58 62 66 70 74 78 82 86 90 94 98 4 12 20 28 36 44 52 60
 68 76 84 92 100 16 32 48 64 80 96 24 56 88 40 8 72

The resulting permutation expressed in cyclic notation:
(1)(51, 26, 57, 29, 15, 8, 99, 50, 63, 32, 90, 73, 37, 19, 10, 53, 27, 14, 54, 6
4, 92, 87, 44, 81, 41, 21, 11, 6, 52, 82, 71, 36, 80, 93, 47, 24, 95, 48, 91, 46
, 62, 66, 67, 34, 59, 30, 58, 65, 33, 17, 9, 5, 3, 2)(76, 85, 43, 22, 56, 96, 94
, 74, 69, 35, 18, 55, 28, 79, 40, 98, 75, 38, 60, 83, 42, 61, 31, 16, 89, 45, 23
, 12, 77, 39, 20, 78, 70, 68, 84, 86, 72, 100, 88, 97, 49, 25, 13, 7, 4)

The fixed element in this Josephus permutation is:
1

n = 100,  m = 2,  f = 1:  Object 72 was selected last!
```

There is 1 fixed element: 1.

F8 is the same as the “Report All” command button.

F9 => Toggle Logging to Log file on/off:

If the “Logging screen to log file mode is off it will be turned on, if on it will be turned off. F9 is the same as the “Logging is On/Off” command button.

F11 => Clear output text box:

This clears the output text box. F11 is the same as the “Clear Output” command button.

F12 => Brings up the Plot form and plots:

If the Plot form is not up, this causes the Plot form to come up and to plot the currently defined permutation. F12 is the same as the “Plot” command button.

Ctrl+F9 => Clear Log File:

This clears the log file and writes a header message in it.

ESC => Interrupt/Abort a long process:

The ESC key from the Run form while a calculation is running is the same as the “Abort Calc” command button. It can be used after the program has been asked to perform a task that is taking longer than the operator is willing to wait.

One of the messages:

```
Manually aborted before Orbits finished
Manually aborted before report finished
```

will appear. The first one appears if we were in the process of computing orbits of the permutation. The second appears if we were in the process of giving a detailed report of the permutation.

Command buttons -

There are 10 command buttons on the Run form:

Abort Calc.: Same as ESC key.
Status of Calc.: Same as F5 key.
Configuration: Same as F6 key.
Report: Same as F7 key.
Report All: Same as F8 key.
Plot: Same as F12 key.
Quit: Quit the Run form and go back to the Startup form. Same as F2 key
Clear Log File: Clears the log file, same as Ctrl+F9 key.
Logging Is On/Off: Same as F9 key.
Clear Output: Same as F11 key.

If the Quit button, File menu Exit, the Run form's Close button Alt+F4, or Upper right hand X button is selected when a calculation is running, indicated by **Running** being displayed on the Run form, the calculation is automatically aborted. The message

```
      When **Running**, select Quit twice to quit
```

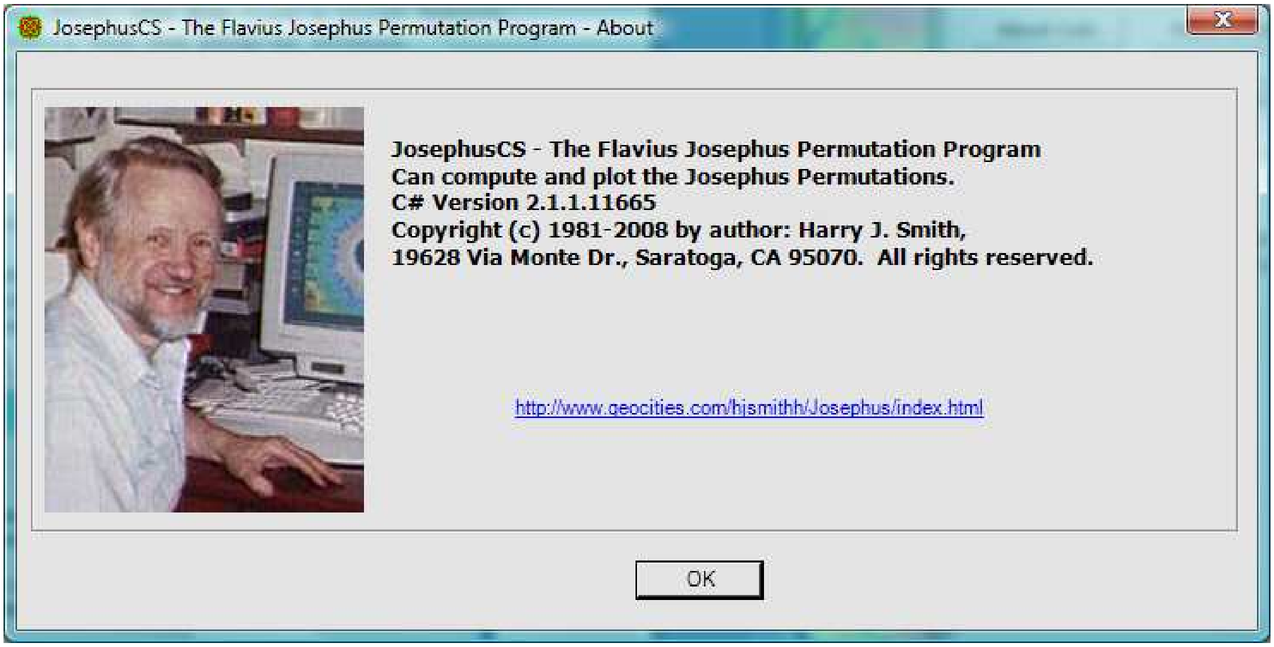
is displayed in the text output text box and a second quit request is required to quit.

If a calculation is not running, the File menu Exit, the Run form's Close button Alt+F4, or Upper right hand X button will totally end the program without stopping at the Startup form. The Quit button or F2 will close all open forms and return to the Startup form with the previously selected parameters.

The Run form has a File and a Help menu button. The File menu has Exit. The Help menu has Help form and About...

Exit is the same as the Quit button except it does a complete exit.

Help form is the same as the F1 key and causes the Help form to be displayed. About will bring up a dialog box with Application Title, Version, Application Description and a URL to obtain the latest version of the program:



JosephusCS - The Flavius Josephus Permutation Program - About

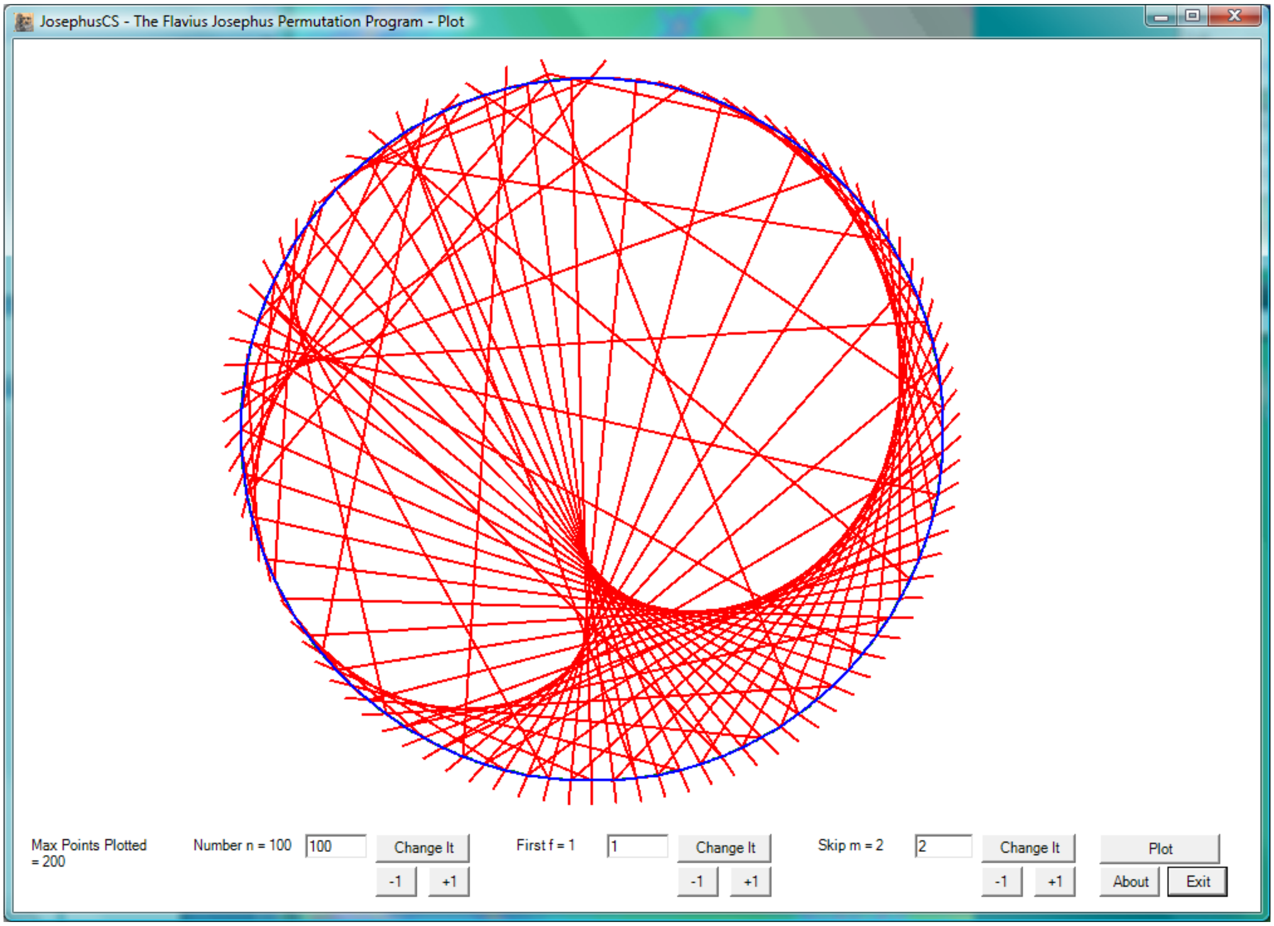
JosephusCS - The Flavius Josephus Permutation Program
Can compute and plot the Josephus Permutations.
C# Version 2.1.1.11665
Copyright (c) 1981-2008 by author: Harry J. Smith,
19628 Via Monte Dr., Saratoga, CA 95070. All rights reserved.

http://www.geocities.com/hjsmithh/Josephus/index.html

Click the OK button, press Enter, or hit the escape button to remove this box.

Click the Plot button and the Plot form is displayed and the currently selected permutation is plotted.

Plot form –



JosephusCS - The Flavius Josephus Permutation Program - Plot

Max Points Plotted = 200

Number n = 100 100

First f = 1 1

Skip m = 2 2

This is an example of the plot form plotting the case of n = 100, f = 1, and m = 2.

Object n is plotted at the very top of the circle. Object 1 is the next point to the right clockwise.

The red lines are drawn from each item in the original circle to where it ends up in the new circle. The extra part of the line outside of the circle shows that it is a point on the new circle. Some plots may have a small extra part of the line outside of the circle that should be ignored.

If n is greater than Max Points Plotted (selectable on the Startup Frame), only the first Max Points Plotted points will be plotted.

The “Number n”, “First f”, and “Skip m” all work the same. To change one of these, edit its value in the TextBox and press Enter or click the “Change It” button. If the edited value is unacceptable, the “Change It” button will be disabled and the value will not change. If you clear the TextBox and try to accept it, the current value will be displayed in the TextBox.

If you click the “-1” button under the “Change It” button and the current value is greater than 1, the current value will be reduced by one, the value in the TextBox will be updated and a new plot will be generated.

If you click the “+1” button under the “Change It” button and the current value is less than Max Number n, the current value will be increased by one, the value in the TextBox will be updated and a new plot will be generated.

If you click the “Plot” button, a new plot will be generated with the current values of n, f, and m.

The “About” button, will bring up a dialog box with Application Title, Version, Application Description and a URL to obtain the latest version of the program.

The “Exit” button will terminate the current plot calculation if any, close the Plot form and go to the Run form.

Messages and error reports -

There are several different messages and/or error reports that are a result of user action (or conceivably an error in the JosephusCS program):

```
JosephusCS - The Flavius Josephus Permutation Program, C# Version 2.1.xxxx.xxxxx
Run on: <system name> by <user name>

Output Screen: Cleared mm/dd/yyyy hh:mm:ss am/pm

Orbits: <xx.xx> Percent done, Memory in use = <xxxxxxx>, DT = <xx.xx> sec.

Manually aborted before Orbits finished

Manually aborted before report finished

When **Running**, select Quit twice to quit

Log file <file name> Opened for Append mm/dd/yyyy hh:mm:ss am/pm
Full name = <directory name>\<file name>

Log file <file name> Closed mm/dd/yyyy hh:mm:ss am/pm

File <file name> opened for writing
Full name = <directory name>\<file name>
Writing file <file name>
Log file <file name> Cleared mm/dd/yyyy hh:mm:ss am/pm
Full name = <directory name>\<file name>

File JosephusCSHelp.txt opened for reading
Full name = <directory name>\JosephusCSHelp.txt

Cannot open file JosephusCSHelp.txt for input
Not in current directory <directory>
OpenRead: Could not find file <directory>\JosephusCSHelp.txt'.

Cannot open <file name> for output
Full name = <directory name>\<file name>
OpenAppend: Access to the path '<directory name>\<file name>' is denied.
Aborting file write...

Cannot open <file name> for output
Full name = <directory name>\<file name>
OpenWrite: Access to the path '<directory name>\<file name>' is denied.
I/O operation aborted by operator!
Aborting file write...
```

Other forms –

Other forms that can be displayed to help in using the program are the Startup, Help, Configuration, Change File Path/Browse For Folder, Select a File, and Plot forms. On all forms except the Run form, pressing the ESC key will remove the form.

Startup form -

The Startup form is shown above and has five parameters that can be set before the program runs:

“Max Number n <= 1000000” can be set to a value from 1 to 1000000 and it can only be changed on this Startup form. The nominal starting value is 1000. Number n cannot be set larger than this.

“Number n” can be set to a value from 1 to Max Number n. The nominal starting value is 100 and it can be changed on the Plot form. n = The number of objects in the circle.

“First f” can be set to a value from 1 to 1000000. The nominal starting value is 1 and it can be changed on the Plot form. f = index of first object to be selected [1, n].

“Skip m” can be set to a value from 1 to 1000000. The nominal starting value is 2 and it can be changed on the Plot form. m = difference in index of items selected, m = 2 => every other item.

“Max Pointed Plotted” can be set to a value from 1 to 1000000. The nominal starting value is 200 and it can only be changed on this Startup form. Large values of this parameter can cause the Plot form to take a very long time before the plot appears and the program may have to be terminated manually by clicking the X in the upper right hand corner of the Plot form or the Run form. If n is greater than Max Points Plotted, only the first Max Points Plotted points will be plotted.

These five values are tested as they are edited. Zeros on the left or spaces on the left or right are removed so you will not see them. These numbers are displayed in **bold** face font iff they are acceptable values. If any of the five values are unacceptable, the Run button is disabled.

A “Default Settings” button is provided to reset the five parameters to their original values.

The Run button is used to bring up the Run form. It can be used after a Run form exits to reinitialize JosephusCS and start running again. The Exit button will terminate the program.

Help form -

The Help form was shown and partially described above. It has a File menu button that has a menu with Print Setup, Print..., Print Preview and Exit. The Print Setup will bring up the Windows printer setup dialog box so you can select the printer to use. The Print... will start the printing process to start printing the help file. This will normally bring up the printers dialog box. The Print Preview does just that, it can also be used for printing. The Exit does the same thing as pressing the escape key; the Help form is removed.

This form also has a “F3 Find:” button, a “F2 Up” button and a small text box for entering a string of characters to find. Pressing F3 is the same as clicking the “F3 Find:” button, the text in the small text box is searched for. Pressing F2 is the same as clicking the “F2 Up” button, same as F3 except the search is done from the current location towards the top of the large text box.

If during a find the text is not found, the sound from NotFound.wav is heard. If it is found, but only by starting over from the other end of the text, the sound from Wrap.wav is heard. This sound process uses the system file winmm.dll. If that file cannot be found, there will be no sound.

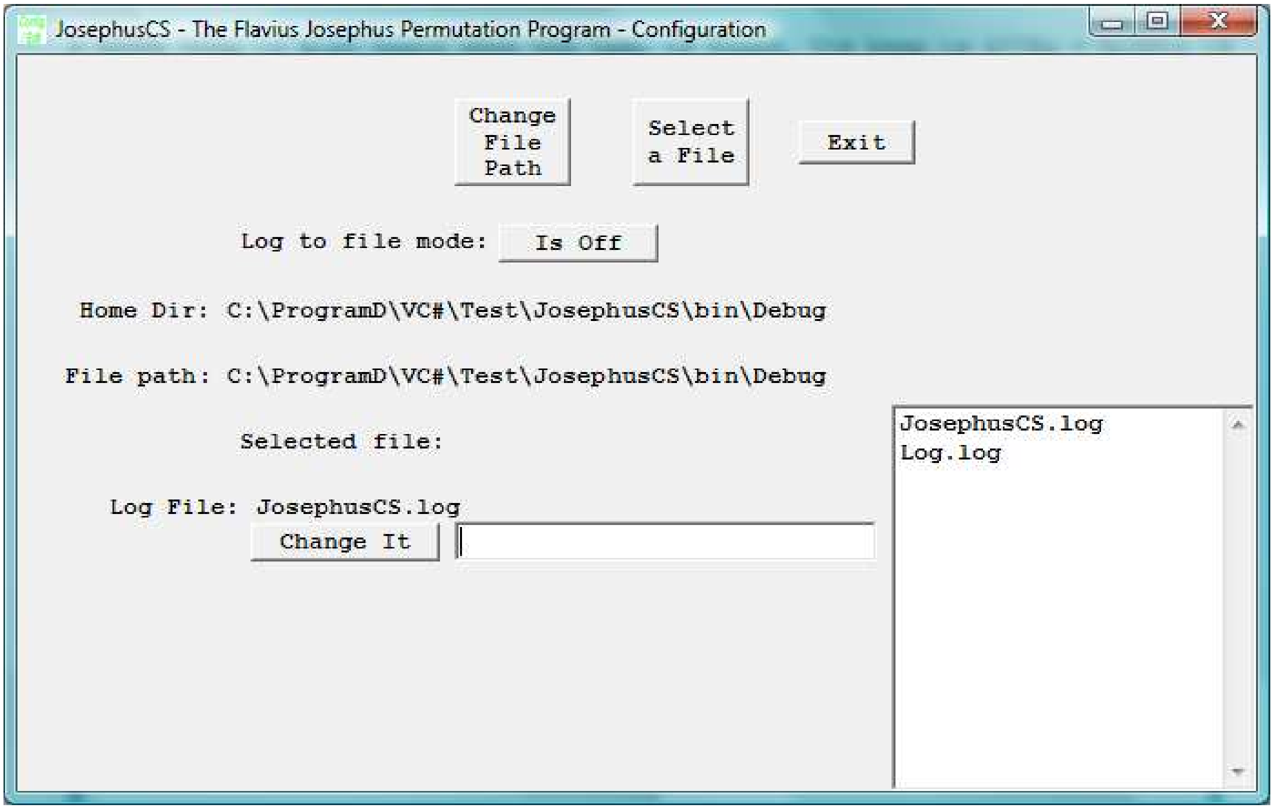
A short beep is also heard when the text is found. The beep is 277Hz = middle C# for a normal search and 554Hz one octave above for a search up. The beep process uses the system file kernel32.dll. If that file cannot be found, there will be no beep.

Text can be copied from the large text box to the small text box by selecting the text with the left mouse button and pressing Ctrl-F.

The Help form is unique in that more than one copy can be brought up at the same time. This may or not be useful, but it illustrates a technique in Windows programming.

Configuration form -

The Configuration form is displayed when function key F6 is pressed or the "Configuration" command button is selected on the Run form:



JosephusCS - The Flavius Josephus Permutation Program - Configuration

Log to file mode: Is Off

Home Dir: C:\ProgramD\VC#\Test\JosephusCS\bin\Debug

File path: C:\ProgramD\VC#\Test\JosephusCS\bin\Debug

Selected file:

Log File: JosephusCS.log

JosephusCS.log
Log.log

This form allows you to change:

Log to file mode.
Name of Log file.

It also shows the Home Directory, File path, and the name of the log file if Log to file mode is on. The Browse For Folder form can be brought up by selecting the "Change File Path" command button, and the Select a file form can be brought up by selecting "Select a File" command button.

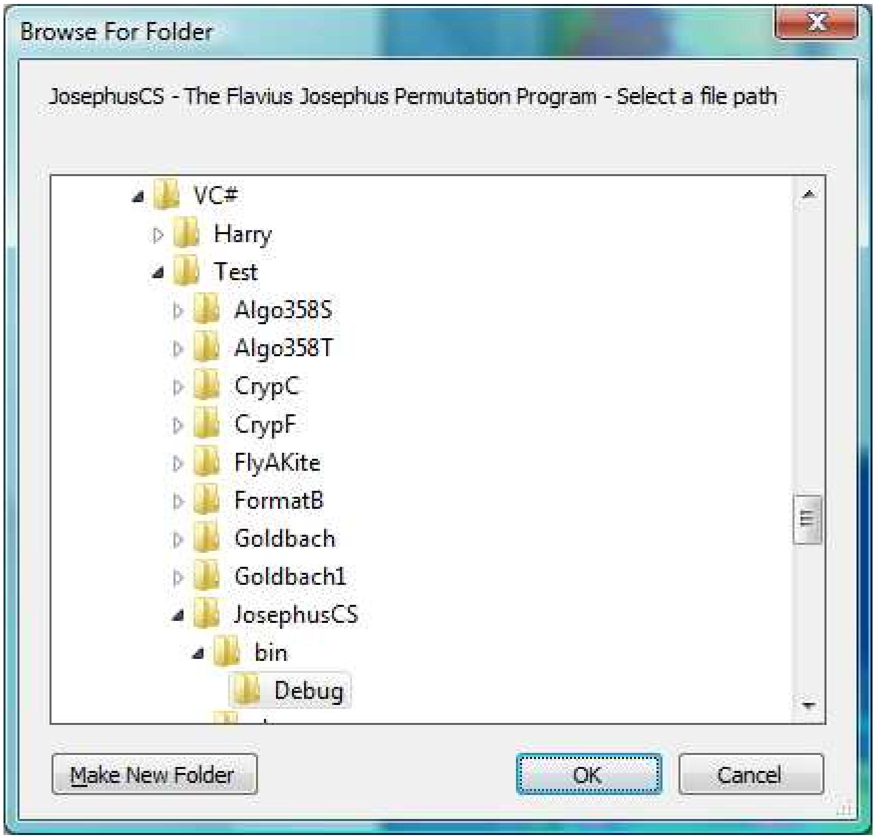
Clicking the "Change It" button when its text box is empty will bring up the current value. It can be edited, and a second click with the text box not empty will accept the entry. A space at the right or a Return/Enter key will also accept it.

The Log to file mode can be changed/toggled by clinking the corresponding button. If the mode is on, the button will says “Is On”, then clicking it will turn the mode off and the button will change to “Is Off”. If the mode is off, the button will says “Is Off”, then clicking it will turn the mode on and the button will change to “Is On”.

The Log to file mode and name of Log file interact in that if the file name is changed while a log file is open, the file name of the open file will not change until the log file is closed and reopened.

Browse For Folder form -

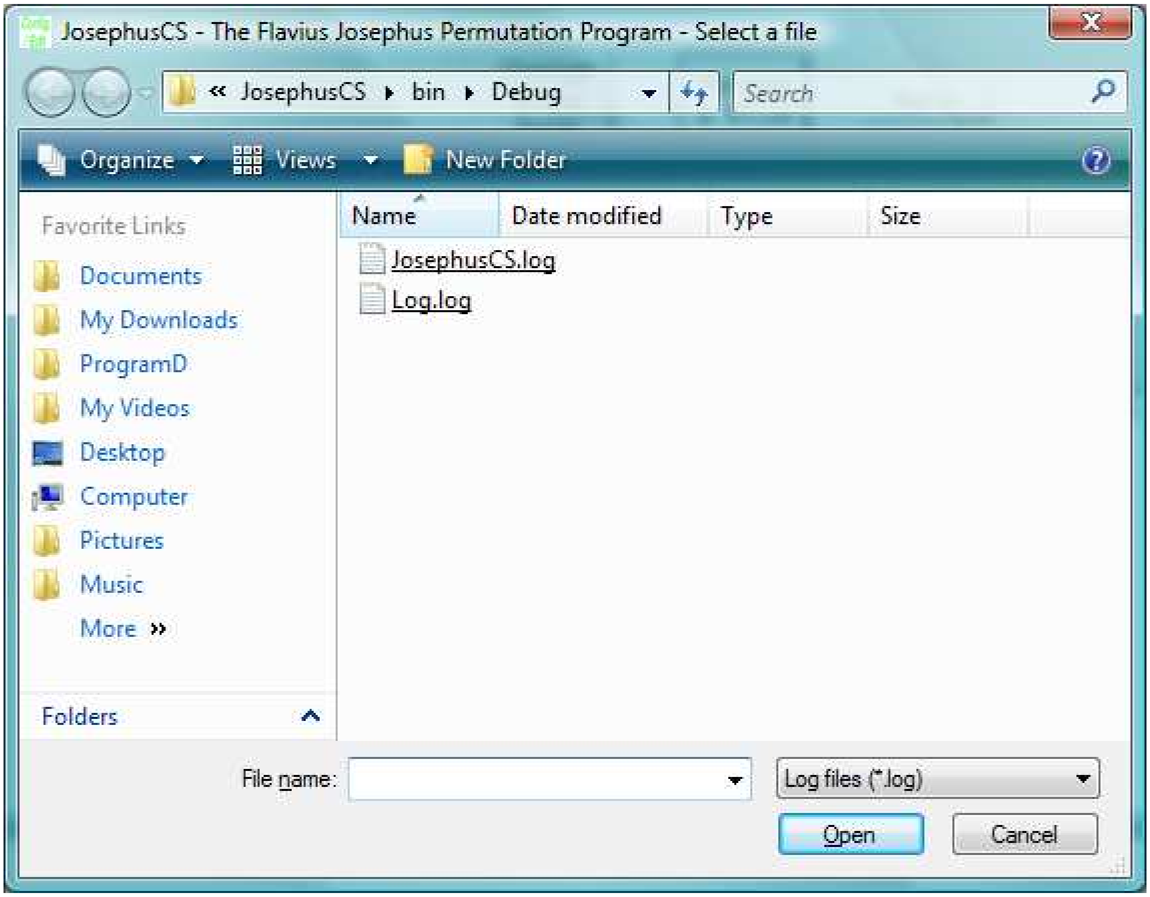
The Browse For Folder form is displayed when the “Change File Path” command button is selected on the Configuration form:



To change the current directory to be used for the Log file, use this form to select a path. This form is resizable as is the Run, Help, and Select a file forms.

Select a File form -

The Select a file form is displayed when the “Select a File” command button is selected on the Configuration form:



Log files (*.log)

This provides a way to browse and select a file. If a file is selected, it can then be used to set a file name on the Configuration form by clicking the file name “Change It” buttons. The first click enters the file name in the text box and a second click will accept it. This form can also be used to change the file path.

The list box in the lower right hand corner of the Configuration form is a list of the files in the currently selected file path. The wild card selection for these files (*.log, or *.*) is the same as last used on the Select a file form.

How things are computed -

```
int[] a;  // Working array of objects
          // Ends up with a[i] = when object i was selected
int[] b;  // Record of objects selected
          // Ends up with b[i] = object # selected on k'th selection
int c;    // Index to current object being scanned
int f;    // # of first object to be selected
int i;    // Utility index
int k;    // Number of objects selected (killed) so far
int m;    // m for selecting every m'th object
int n;    // The total number of objects in the circle to start with
int p;    // Index to previous object
int r;    // Remaining number of objects in the circle
int s;    // Number of fixed (stationary) elements in permutation
int maxS; // Up to maxS saved fixed elements (maxS = 10)
int[] x;  // Saved fixed elements

void Orbits() // Determine the orbits
// uses n, f, and m as input
// computes a[], b[], and s
```

```
{
    for (i = 1; i < n; i++)
        a[i] = i+1;                 // Set position i to point to next position
    a[n] = 1;
    p = 1 + ((f % n) + n - 2) % n;  // Init previous position based on f
    k = 1;  r = n;
    do
    {
        if ((1 < k) && (k < n))     // If not first or last selection
        {
            for (i = 1; i <= (m - 1) % r; i++)  // Move m-1 positions (zero OK)
                p = a[p];
        }
        c = a[p];                   // Update current position
        b[k] = c; c = a[c];         // Select the object at c
        a[a[p]] = k;                // Save execution order
        k++; r--;
        a[p] = c;                   // Re-link circular chain of pointers
    } while (k <= n);               // Until all are selected
    a[c] = n;                       // a[c] was clobbered, so restore it
    s = 0;                          // Count number of fixed elements
    for (i = 1; i <= n; i++)
        if (b[i] == i)
            s++;
} // Orbits

void ReportDetails()  // Generate detailed output
// uses n, a[], and b[] as input
{
    WriteLn("Execution order, when object i was selected:");
    for (i = 1; i <= n; i++)
        Write(a[i] + " ");
    WriteLn();  WriteLn();
    WriteLn("Order of execution, object # selected on k'th selection:");
    for (i = 1; i <= n; i++)
        Write(b[i] + " ");
    WriteLn();  WriteLn();
    WriteLn("The resulting permutation expressed in cyclic notation:");
    p = b[1];  c = 1;
    int[] t = new int[n + 1];   // temporary working array of objects
    for (i = 1; i <= n; i++) t[i] = a[i];
    do
    {
        Write("(");
        do
        {
            if (c != 0)
            {
                Write("" + c);
                t[p] = 0; p = c;
            }
            c = t[p];
            if (c != 0) Write(", ");
        } while (c != 0);
        Write(")");
        i = 0;
        do
        {
            i++;  c = t[i];
        }
        while ((c == 0) && (i != n));
        if (c != 0) p = b[c];
    } while (c != 0);
    WriteLn();  WriteLn();
    ReportFixed();
} // ReportDetails

void ReportFixed() // Generate output of all fixed elements
// uses n, s, and b[] as input
{
    string Ss;                      // String "s" or "" if s = 1
    string Sis;                     // String "are" or "is" if s = 1 }
    if (s == 1)                     // Setup is or are
    {
        Ss = ""; Sis = "is";
    }
    else
    {
        Ss = "s"; Sis = "are";
    }
    Common.WriteLn("The fixed element" + Ss + " in this Josephus permutation " +
Sis + ":");
    for (i = 1; i <= n; i++)
    {
        if (b[i] == i)                  // Output fixed elements
            Common.Write(i + " ");
    }
    if (s == 0) Common.Write("None!");
    Common.WriteLn();
    Common.WriteLn();
} // ReportFixed

void ReportResults() // Generate normal output
// uses n, f, m, s, maxS, b[], and x[] as input
{
    GenerateFixed();
    int nx = 0;                     // Number of saves fixed elements
    string Ss;                      // String "s" or "" if s = 1
    string Sis;                     // String "are" or "is" if s = 1 }
    if (s == 1)                     // Setup is or are
    {
        Ss = ""; Sis = "is";
    }
    else
    {
        Ss = "s"; Sis = "are";
    }
    Common.Write("n = " + n + ",  m = " + m + ",  f = " + f);
    if (f > n) Common.Write(" => " + 1 + (f - 1) % n);
    Common.WriteLn(":  Object " + b[n] + " was selected last!");
    Common.Write("There " + Sis + " " + s + " fixed element" + Ss);
    if (s > 0)
    {
        nx = s;
        if (nx > maxS) nx = maxS;
        Common.Write(":  ");
        for (i = 1; i <= nx; i++)
        {
            Common.Write("" + x[i]);
            if (i != nx) Common.Write(", ");
        }
    }
    if (s > nx) Common.Write(", ... ");
    Common.WriteLn(".");  Common.WriteLn();
} // ReportResults
```

```
GenerateFixed() // Generate array of first fixed elements
// uses n, maxS, and b[] as input
// computes x[]
{
    Int sl = 0;                 // number of fixed elements found
    for (i = 1; i <= n; i++)
    {
        if (b[i] == i)
        {
            sl++;
            if (sl <= maxS)
                x[sl] = i;      // Save up to maxS fixed elements
            else
                return;
        }
    }
} // GenerateFixed
```

Distribution files –

The distribution files can be downloaded from my website:

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

in the **Files to Download** section

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

When you install the program using the distribution file JosephusCS21?.exe or JosephusCS21?.zip, a folder is created with 3 subfolders:

```
├──JosephusCS 2.1
│  ├──doc
│  ├──run
│  └──src
```

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 4 files:

00.txt	(A short list of features)
Fixes.txt	(A list of features and fixes added to each version)
JosephusCS.doc	(Microsoft Word file)
JosephusCS.txt	(Plain ASCII text copy of the .doc file without graphics)

There is also available on my website a copy of the .doc file in an Adobe Acrobat Reader file JosephusCS.pdf.

http://www.geocities.com/hjsmithh/Josephus/index.html .

The run subfolder has the following 7 files

00.txt
JosephusCS.exe
JosephusCS.log
JosephusCSHelp.txt
Log.log
NotFound.wav
Wrap.wav

The .exe file is executed from there.

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

```
00Note.txt
AboutForm.cs
AboutForm.resx
AssemblyInfo.cs
Common.cs
ConfigEdit.ico
ConfigForm.cs
ConfigForm.resx
COPYING.txt
cs.ico
Global.cs
harry.jpg
HelpForm.cs
HelpForm.resx
JosephusCS.cs
JosephusCS.csproj
JosephusCS.ico
JosephusCS.sln
JosephusCS.suo
JosephusCS.csproj.user
PlotControl.cs
PlotControl.resx
PlotForm.cs
PlotForm.resx
RunForm.cs
RunForm.resx
StartupForm.cs
StartupForm.resx
```

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/