PI Testing Plan

Garett Skaar
Team 4
HW 01 - Testing Plan
CS 430 - Fall 2018

We need to test our PI calculator and ensure that the output of our program is correct for all decimal places. The program will output PI calculated to the user defined decimal place or summation iterations.

Calculating PI

Testing our PI calculations is rather straightforward. The program will calculate PI using the Leibniz formula and output the result to the desired decimal place (user defined) or number of iterations (user defined). To test the correctness of our calculations, we must compare the program output to that of which we know is correct. There are numerous resources to get known PI values to a given decimal place.

Test cases should include several different sample inputs with known PI values.

Some sample test values (not nearly enough for proper testing):

Command Line Arguments	Calculated PI output (known)
-i 1	4.0
-i 20	3.09162380667
-i 1000	3.1405926538
-p 2	3.14
-p 5	3.14159
-p 20	3.14159265358979323846

(Note that Leibniz converges very slowly and large iteration values are needed to get legitimate PI values).

User Input/Edge Cases

The program depends on proper user input. Testing should check that user input is valid and can be used to run the program with accurate output.

Testing should include:

- Negative numbers (-p -1)
- Improper flags (anything other than -i or -p)
- Anything more or less than 2 arguments
- Proper argument ordering (-i 2 not 2 -i)
- 0 argument (-p 0)
- Maximum values for iterations and decimal places.

Proper usage messages should be displayed to help the user input valid command line arguments.

Memory Usage/Overflow

As stated above in user input, we must be aware of the amount of space we have for our PI output. We need to check that our calculations do not overflow our maximum allowed value for the double variable. Variable size is machine dependent and and requires a check while the calculation is running.

We must also be aware that computation time could take large amounts of time if the user inputs high values of iterations or decimal places. Perhaps a maximum value should be set for both options.

All testing should be done using the Google C++ Testing Framework.
https://www.ibm.com/developerworks/aix/library/au-googletestingframework.html