Michael Owusu

Com Sci 161

4/27/2014

Supplemental Problem 6 Write-up

**Discussion of Testing:**

The coding for this problem was not very extensive. The only testing I found necessary beyond basic troubleshooting was testing to make sure that my program’s calculations are correct.

1. A possible problem was the incorrect implementation of calculations to check whether a number is “Syracuse” or not.

To test this, I did the necessary calculations to test the if a range of numbers were indeed “Syracuse”. I wrote the expected outcomes out by hand and compared them to the outcomes of my program. I tested a few ranges for correctness and since the rest of the numbers follow the same code, I knew they were right.

2. A test case to be considered is when the calculations to check “Syracuse” take over 150 attempts. In this case the program should stop and declare the number a potential non-”Syracuse” number.

To test whether or not this was this case, I identified a few numbers that I knew took a lot of calculations to check “Syracuse” so I found where the 150th calculation and compared it with my expected outcome. When it returned the error message I was expecting, I knew that my code was working in this respect.

**Evidence For Correctness**

My program’s purpose is to show that all integers between 1 and 2000 inclusive are "Syracuse". In the following sample output, you can see that my programs calculations are correct. My program can produce these calculations for all values from 1 to 2000. It does the method outlined in the prompt for calculating “oneness” and even has options for manageable outputs. I know my program is correct because its calculations agree with one’s I have done by hand.

**Sample Output:**

Michaels-MacBook-Pro:CSC161 Michael$ gcc -o supp6 supp6.c && ./supp6

**supp6.c:103:5: warning: implicit declaration of function 'sleep' is invalid in C99**

**[-Wimplicit-function-declaration]**

sleep(1);

**^**

1 warning generated.

Welcome!

This program checks that all integers between 1 and 2000 inclusive are 'Syracuse'.

To check if a number is 'Syracuse' this program divides n by 2 when n is even.

If n is odd it is multiplied by 3 and incremented by 1.

This process is repeated until n reaches 1.

If after 150 calculations n does not equal 1, calculations for that number are stopped and that number is declared a potential non-'Syracuse' number. (Further calculations needed to prove it is indeed 'Syracuse').

Now...

Please select the range of numbers (1-2000 inclusive) you would like to check for 'Syracuse'.

Select your starting number:

10

Select your ending number:

12

Beginning calculations...

n of 0 = 10

n now = 5

n now = 16

n now = 8

n now = 4

n now = 2

n now = 1

n finally = 1

and 10 is therefore a 'Syracuse' number!

n of 0 = 11

n now = 34

n now = 17

n now = 52

n now = 26

n now = 13

n now = 40

n now = 20

n now = 10

n now = 5

n now = 16

n now = 8

n now = 4

n now = 2

n now = 1

n finally = 1

and 11 is therefore a 'Syracuse' number!

n of 0 = 12

n now = 6

n now = 3

n now = 10

n now = 5

n now = 16

n now = 8

n now = 4

n now = 2

n now = 1

n finally = 1

and 12 is therefore a 'Syracuse' number!

END PROGRAM.