CPSC 275 Introduction to Computer Systems

Programming Project 3

November 26, 2018

Due December 10, 2018

A ***Collatz Sequence*** is a sequence of positive integers *C* defined for any positive integer *n* as follows:

$$C_0 = n$$

$$C_{k+1} = \begin{cases} C_k / 2, & if\ C_k\ even \\ 3 * C_k + 1, & if\ C_k\ odd \end{cases}$$

For example, the Collatz sequence starting at *n = 12* is: 12, 6, 3, 10, 5, 16, 8, 4, 2, 1.

The final value of *k* is called the ***stopping time*** of the sequence. In the example above, the stopping time is 9.

The ***Collatz Conjecture*** says that all positive integers have a finite stopping time. It was first posed in 1937 and it has never been proven. However, sequences have been calculated up to very large *n* and no counterexample has been found.

Write a C program **collatz.c** that takes as a parameter a value of *N,* and computes the stopping time for each *n <= N.* Your output should look like this:

```
n = 1       : 0
n = 2       : 1
n = 3       : 7
…
```

Your program should include a function written in assembly language that computes the stopping time for a given *n*. Its C prototype is as follows:

```
int collatz(int n);
// computes the Collatz Sequence for n
// returns the stopping time
// performs no input/output
```

Extra Credit Supplement

Only attempt this version of the program if you have successfully completed the basic one. This one, longcollatz.c, computes the stopping time for a single integer *n.* However, this version of the program should support the use of `long` integers. This will require you to implement 64-bit arithmetic in 32-bit assembly language, where both the input number and the result may both be `long`.

Special Policies for EC Supplement

This is not a big coding exercise, it's more about figuring it out. As a result, this is more than usually an individual project. Specifically:

1) TA's will not discuss the problem with you, and I will only answer questions about the requirements.
2) You are not to discuss the problem at all with classmates.
3) Your solution should be accompanied by a paragraph or two explaining your approach. (Your assembly code should still be commented, but may refer to this document.)
4) You may utilize Web resources, but you must cite them in your solution description.