ETEC2110-01 **Lab5: Control Structures** **Points: 35**

Background: Collatz Conjecture

In 1937 a mathematician named Lothar Collatz, introduced the following conjecture:

Start with any positive integer:

Repeat the following until the number 1 is reached.

If the number is even, divide it by two.

If the number is odd, triple it and add one.

Collatz claimed that the sequence would eventually terminate for any positive integer.

Some numbers terminate quickly:

(8: 8→4→2→1)

…and other numbers may continue for some time

(7: 7→ 22→ 11→ 34→ 17→ 52→ 26→ 13→ 40→ 20→ 10→ 5→ 16→ 8→ 4→ 2→ 1)

His conjecture is very difficult to prove and is an open problem in mathematics. It is simple and can be interesting to play with.

Tasks:

1. (**20 points**) Write a C program:
   - make two functions (in collatz.h / collatz.c) [you can define other functions if you wish]

```
int collatz_length(long long int number)
{
    //This function should return the count of how many Collatz
    // iterations it takes before number reaches 1.  For example:
    // collatz_length(4) would return a value of 3.

    return count;
}

void collatz_print_sequence(long long int number)
{
    // This function should print the Collatz sequence for the number
    //  provided.  For example: collatz_print_sequence(4) would print
    //  the following: (4: 4 -> 2 -> 1)
    return;
}
```

   - make the program iterate over numbers from 1 to 1,000,000,000, calling the collatz_length function for each and determining the number with the longest Collatz sequence (if there's a tie, just pick one of them arbitrarily). For the longest one, print out the Collatz sequence.
2. (**15 points**) Write a python, Java (or MathPiper?) version of this same program, including the use of functions (and modules) – if you're not sure how to correlate what you did in C to one of these other languages, ask. Calculate the time it takes (in seconds) to complete both the C and other program.
   - In C, I used the clock and difftime functions defined in <time.h>
   - Put the time taken (on your computer) to run the program in a comment in your main program file.
   - Tip: don't wait until the last minute. It took…a while…for my computer to complete this
   - Tip2: don't immediately test the 1,000,000 values – start with something small (like 10) to work out the bugs.