**Project 2 - Loops and Bit Operations**

Part 1 - Using Loops and Decisions

In Part 1 of the lab we will be using variables and math. All of your code should go into the main function (do not use separate functions).

In the Fibonacci sequence, the first two Fibonacci numbers, called f0 and f1, are defined to be 0 and 1, respectively. Thereafter, each successive Fibonacci number fi is defined to be the sum of the two preceding Fibonacci numbers fi-1 and fi-2. So f*2* is calculated by adding together the values of f0 and f1.

* + - Write code that generates the first 20 Fibonacci numbers using a loop.
    - Print the values inside the loop.

Part 2 - Bits and Bytes

In Part 2 you are going to write a looping binary printer that prints out the string representation of the binary value. All of your code can go into the main function (you may use a separate function call if you choose).

# Part A - Binary Printer

* + You will need to create a code block that uses bit shifting to print the binary representation of an integer to the screen. To do so, you will need to use a bit mask and bitwise right shift. We will only be working with 32 bit integers, so you can hardcode the loop that prints the values for 32 iterations.
  + You should have 5 separate loops that test the following values:
    - 2
    - 255
    - 32
    - -1
    - -255
  + You can use the following website to check your results: <http://www.binaryhexconverter.com/binary-to-decimal-converter>

# Part B - Printing a Random Binary Value

* + Using the library [rand()](http://www.cplusplus.com/reference/cstdlib/rand/) function, generate a random number and print the binary representation of that number to the console.
    - Include the library <limits.h> at the top of your main source code file so you can use the global constant INT\_MAX and INT\_MIN to make your random number fall between the minimum and maximum values for an integer.
    - Using the following expression, (rand() % INT\_MAX) + (rand() % INT\_MIN), you can get a number between the largest and smallest integers on your machine.

Part 3 - Submission

Create a tar archive with the command “tar -czvf lab2.tar.gz .”, and then email your archive to bu580u2017@gmail.com and cc your TA dmu1@binghamton.edu before the submission deadline. Make sure you do not include the executable in your archive (make clean before creating the archive). Late assignments will not be accepted under any circumstances. Plan to turn in your assignments early.

Demo your lab before the demo deadline (after the submission deadline) by downloading your submission from class Gmail and extracting your archive with the command “tar -xvf lab2.tar.gz”. Then compile (with your makefile), and run your code, show your source to your TA, and answer any questions your TA may have.

Grading Guidelines

## Part 1:

* + Compiles and prints first 20 Fibonacci numbers: 4 points

## Part 2:

* + Part A: 3 points
  + Part B: 2 points

## Style Guidelines - 1 point

* + Uses whitespace in source to clearly identify code blocks
  + Clear variable names
  + No single letter names (except 'i')
  + Clearly formatted output

Submission Deadline: 11:59pm 9/14/2017 EDT

Demo Deadline: 12:00pm 9/27/2017 EDT