Project 2 - Loops and Bit Operations

Due Date: 5:00 p.m., September 25, 2015

Part 1 - Using Loops, and Decisions

In Part 1 of the lab we will be using variables and math. All of your code can go into the main function, you do not have to use separate functions.

- ## Part A
    - 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 f*i* is defined to be the sum of the two preceding Fibonacci numbers f*i*2 and f*i*1. So f*i*2 is calculated by adding together the values of f*i*0 and f*i*1.
        - 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 seperate loops that test the following values:
        - 2
        - 255
        - 32
        - -1
        - INT_MAX
        - INT_MIN
            - 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

    - 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() function, generate a random number and print the binary representation of that number to the console.
        - Use INT_MAX and INT_MIN to make your random number fall between the minimum and maximum values for an integer.
        - Come up with an expression that generates a random number anywhere between the largest and smallest signed integers on your machine.

Part 3 - Submission

- Create a tar archive with the command ”tar -cvf lab2.tar.gz .”, and then upload the archive to Blackboard before the deadline. Make sure you do not include the executable in your archive (make clean before creating the archive).
- Demo your lab by the beginning of next lab by downloading from blackboard 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 the TA, and answer any questions he or she may have.

Grading Guidelines

- **Part 1:**
    - Compiles and prints first 20 fibonacci numbers: 3 points
- **Part 2:**
    - Part A: 3 points
    - Part B: 3 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