Part 1: Setting Up Your Environment

Due Date: 5:00 p.m., September 2, 2016

All programs will be tested on the machines in the Q22 lab. If your code does not run on the system in this lab, it is considered non-functioning EVEN IF IT RUNS ON YOUR PERSONAL COMPUTER. You do not have to write your program on the lab machines, but always check that your code runs on the lab machines before submitting.

For the first part of program we are going to log into our accounts and write a 'Hello World' C program. This will ensure everyone has access to their accounts and all of the tools needed for the course.

We will be using the Q22 CS LDAP accounts for this course.

Connecting to CS LDAP account:
If you do not know what your password is, go to the following web page, and request a new password.
https://www.cs.binghamton.edu/~sysadmin/
Once you have reset your password, you should be able to log into the system.

If you want to access the program machines remotely, you can do so with SSH by connecting to remote.cs.binghamton.edu with your PODS id and password.

- If you are on OS X or Linux, you already have SSH, and can connect as shown in class.
- If you are on a windows machine, you can download the putty program (select the putty.exe link) to have ssh on your machine.

- ## Part A: Writing the code
    - If you haven't already, create a 'cs580u' folder in your home folder. Inside CS580, create a 'programs' folder. Next create a 'program1' folder inside the 'programs' folder.
    - As shown in class, create a file in the text editor named "program1.c", and save it to the 'cs580/programs/program1' folder you just created.
    - Open Terminal. Using the 'cd ~/cs580u/programs/program1' command (the '~' is a stand-in for your home directory), change the directory to your program directory you just created
    - Create the program file using the following command: 'touch program1.c'
    - You can edit the text file you created in 2 different ways:
        - Using command 'nano program1.c'. (nano tutorial)
            - *You may use any command line editor you like, nano is just what I recommend for a beginner*
        - Open the file in the GUI text editor (gEdit).

        - from the CL, type
            - gedit <filename>&
    - In the main() of the driver code provided, write a single line that prints out hello world.
    - Save and exit.

Part B: Compiling and Executing

- Like you did in the previous step, navigate to your program folder in the shell.
- Type in 'ls' to list the directory's files. Ensure that 'program1.c' is in the directory.
- To compile, we will be using the program called ‘gcc’. As described in class, compile your code to an executable using gcc.
    - You will need to set a flag to use the math library, -lm
        - gcc program1.c -o program1 -lm
- The source should compile to an executable. You can run the executable with the command, './<executable>. In this case, './program1'

Part 2 - Using Variables and Arithmetic

For the second part of the program we will be using variables and math. Helpful link: printf

Part A

- Uncomment the remainder of the code, and add the following expressions expressions shown here in the main:
    - $3x^3 - 5x^2 + 6$ for $x = 2.5$.
    - The result of $(3 \times 10^{8} + 2 \times 10^{-7}) / (7 \times 10^{-6} + 2 \times 10^{8})$ **rounded to the nearest integer**
        - read the comments in the code to know where to write your code
        - The standard C library does not have a round function. You can instead use casting to round a number up or down.
- To round off an integer i to the next largest even multiple of another integer j, the following formula can be used:
    - int next_multiple = i + j - i % j
        - For example, to round off 256 days to the next largest number of days evenly divisible by a week, values of i = 256 and j = 7 can be substituted into the preceding formula as follows:
          int next_multiple = 256 + 7 - 256 % 7

        - Write a function called findNextMultiple(int number1, int number2) to find the next largest even multiple for the following values of i and j:

i	j
365	7
12258	28
996	4

    - Write a function, float convertFtoC(float fahrenheit), that converts 27° from degrees Fahrenheit (F) to degrees Celsius (C) using the following formula and returns the result:
        - C = (F - 32) / 1.8
- Part B
    - In the next part of the program we are going to see how choosing the wrong data types and careless casting can result in data loss. You should see inaccurate results.
        - Write a function to typecast a long integer to the following datatypes
            - int
            - double
            - char
- Part C
    - 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 a function that generates the first 20 fibonacci numbers (including 0 and 1) using a loop.
        - You should return the final resulting value

Part 3 - Submission

- Required code organization:
    - program1.c
- While inside your program1 folder, create a zip archive with the following command
    - zip -r program1 *
        - This creates an archive of all file and folders in the current directory called program1.zip
        - **Do not zip the folder itself, only the files required for the program**
- Upload the archive to Blackboard under Program 1.

Driver Code and Test Input Files

- ### Driver Code
  - program1.c
    - *All of the function interfaces are provided for you. You just need to implement them.* ***You must use the driver code as I have given you and only make alterations where stated. DO NOT change it to take user input, etc.***
  - Any additional files should be included in your project1.zip submission

- ### Test Input Files
  - N/A

Grading Guidelines

- **Part 1 (1 point):**
  - Compiles and outputs hello world when run: 1 point
- **Part 2 (8 points):**
  - Part A: 2 points
  - Part B: 3 points
  - Part C: 3 points
- **Style Guidelines (1 point)**
  - Follows requested program structure and submission format
  - Follows formatting guidelines