CS 100 Lab Three – Spring 2016

Create a directory called `lab3` on your machine. Move into that directory. Complete the four tasks shown below.

1. We took `stats2.c` from **lab2** and broke it into functions. Retrieve **stats2.c** from **troll.cs.ua.edu** using:

   ```
   Windows:  wget troll.cs.ua.edu/cs100/labs/stats2.c
   Mac:      curl -O troll.cs.ua.edu/cs100/labs/stats2.c
   ```

   The main routine already exists; you just need to implement the following functions in **stats2.c**
   - **void readData(int *)** – takes one argument (your array) and reads five values into the array.
   - **double findMean(int *)** – returns the mean of the five values in the arrays
   - **double findVariance(int *, double)** – returns the variance, when passed the data array and mean
   - **void printResults(double, double, double)** – prints the median, variance and standard deviation

   When testing on **cs-intro.ua.edu**, remember to compile with `gcc stats2.c -lm`

2. Name this program `avg.c` – Retrieve the file **avg.c** from **troll.cs.ua.edu** using `wget` or `curl`. For this, we are simply calculating the average of three numbers that the user enters. Our goal with this program is to be as polite as possible to the user when asking for the input. In **getNumber**, you should prompt the user as follows

   ```
   Please enter the first number :
   Please enter the second number :
   Please enter the third number :
   ```

   We are only using one input function, **getNumber**, so it has to work for all three prompts. As you can tell from the code, **getNumber** takes a single argument, a string that contains "**first**" or "**second**" or "**third**" and returns the integer that the user entered.

3. Name this program `shift.c` – Retrieve the file **shift.c** from **troll.cs.ua.edu** using `wget` or `curl`. For this program, you need to write two functions, **shiftLeft** and **shiftRight**. These functions both take three integer values and then shift these values to the left (or right). This is a variation on **swap**. Instead of swapping two values, you shift the three values to the left (or right). The main routine has already been written for you, you simply need to write the two functions **shiftLeft** and **shiftRight**.

4. Name this program `song.c` – When you were little, they made you sing songs in school. We're not going to do that, but we will print the lyrics for one Christmas song – the twelve days of Christmas. Retrieve the file **song.c** from **troll.cs.ua.edu** using `wget` or `curl`. The skeleton of the program is written for you – a main routine and twelve different functions. You need to fill in the details on what takes place in functions **day02()** through **day12()**. For each of these functions, you must adhere to the rules of:
   - Print only a single line of output in that function (what you got on that day of Christmas)
   - Call only one other function (that prints the other stuff you got from all the other days)

   The purpose of this exercise is to make you realize that functions can call functions that can call other functions that can call other functions (and on and on and on).
   If you don't know this song at all, check out http://www.carols.org.uk/the_twe1ve_days_of_christmas.htm

Submit your lab

First, on your local machine, bundle the files in your **lab3** directory into a single (compressed) file. To do this:
- PC: Using Windows Explorer, right click on the **lab3** directory and select "Send To" and then "Compressed (zipped) folder"
- Mac: Using Finder, use a secondary click on the **lab3** directory and then select "Compress *foldername*"

Once you have a compressed file that contains your four lab3 programs, submit that file to Blackboard.

Attendance: We will circulate a roster sheet shortly after lab starts and again about half-way through the lab. Not being present to sign the roster sheet will result in a deduction of 25 points for each missed signature.