CS 100 Lab Five – Fall 2017

Create a directory called **`lab5`** on your machine using **`mkdir lab5`** and move into that directory with **`cd lab5`**
Complete the following problems. Make sure to prompt the user for any input needed by the program.

1. Name this program **`stats.c`** – This program below performs some basic statistics on an array of integers. It must use the five functions indicated. You need to complete this program (write the five functions). If you are using the math functions sqrt and pow, your compiler might require the compile option **–lm**, as in **`gcc -Wall -lm stats.c`**

```
#include <stdio.h>
#include <math.h>
void getData(int, int[]);
double calcMean(int, int[]);
double calcVariance(int, int[], double);
double calcStdDev(double);
void printResults(double, double, double);
int main(void) {
        int size;
        double mean, variance, stddev;
        printf("Enter the array size : ");
        scanf("%d", &size);
        int array[size];
        getData(size, array);
        mean = calcMean(size, array);
        variance = calcVariance(size, array, mean);
        stddev = calcStdDev(variance);
        printResults(mean, variance, stddev);
        return 0;
}
```

   Hint: the math involved is not hard and can be found at https://www.mathsisfun.com/data/standard-deviation.html

2. Name this program **`rpsls.c`** – The game of rock-paper-scissors was expanded so that it now includes rock-paper-scissors-lizard-spock. The expansion was created by Sam Kass and Karen Bryla and made famous on the Big Bang Theory. See http://bigbangtheory.wikia.com/wiki/Rock_Paper_Scissors_Lizard_Spock for the game rules.
   This program asks the user for the number of games to play, then simulates playing **that number of** games by repeatedly generating two random numbers (**0=rock, 1=paper, 2=scissors, 3=lizard,** and **4=spock**) and then determining the winner. The main program, shown below, counts the number of times each player won and then prints the results.

```
int main(void) {
    srand(0);
    int games, p1, p2, ans, ties=0, p1wins=0, p2wins=0;
    games = getNumGames();
    for (int a=0; a<games; a++) {
        p1 = rand() % 5;
        p2 = rand() % 5;
        ans = findWinner(p1, p2);
        if (ans == 0) ties++;
        if (ans == 1) p1wins++;
        if (ans == 2) p2wins++;
    }
    printResults(games, ties, p1wins, p2wins);
    return 0;
}
```

   You must use at least three functions in your program to accomplish the three tasks shown below

   - **`int getNumGames(void);  // get number of games to play`**
     - Prompt the user for the number of games to play, read that number, return it
   - **`int findWinner(int, int);  // determine a single game winner`**
     - Returns a 0 for a tie, 1 if player 1 won that game, and a 2 if player 2 won that game
     - The two arguments passed to the function are player 1's move (0-4) and player 2's move (0-4)
   - **`void printResults(int, int, int, int);  // print the final results`**
     - Prints the results after playing the specified number of games
     - The four arguments passed are number of games, ties, player 1 wins, and player 2 wins

Submit your lab

First, on your local machine, compress your lab5 directory into a single (compressed) file.
Second, once you have a compressed file that contains your two lab5 programs, submit that file to Blackboard.