CSci 1113: Introduction to C/C++ Programming for Scientists and Engineers Homework 03: Loops and Random Numbers Spring 2017 Due Date: Friday, Feb. 24, 2015; 6:00pm.

Purpose:
This homework involves some further basics of C++ programming, specifically loops. You will also get to work with random numbers, which are important in scientific simulations.

Instructions:
The instructions are similar to the previous homework: this is an individual, **not collaborative assignment**; remember to be careful with naming rules; etc. See the HW 0 description for more details.

Problem Introduction:
Probabilistic simulations are an important computational tool used in a variety of areas. Suppose we have a random process and take N random samples for some positive integer N. Then we take a N more random samples of the same process. How close do we expect the average of the first sets of samples to be to the second set of samples? For example, if the random numbers are from a uniform distribution between 0 and 100, what is that chance that the average of 100 random samples differs by more than 3.0 from the average of the next 100 random samples?

Researchers who study probability have derived some powerful theoretical results about questions like these. But another (less rigorous but still useful) way to study such questions is though computational simulations. This homework asks you to write some C++ programs that simulate sets of random samples.

Problem A: Flipping a Fair Coin
Write a C++ program that simulates flipping a fair coin. Your program should do the following:

1. Ask the user to input a random seed (of type int), and call `srand` with that input number as its argument. That is, right after the variable declarations in your program you should have the lines:
   ```
   cout << "Input the random seed: ";
   cin >> seed;
   srand(seed);
   ```

2. Ask the user to input a difference value d, which should be an int.

3. Uses the cmath function rand() to generate a random number that is either 0 or 1. A 0 corresponds to a tail, and 1 to a head, and both possibilities should be equally likely. Repeat this process 100 times, counting the number of heads. Note that rand() generates double-type random numbers from 0 to 1 (inclusive), with all numbers being equally probable. Your task is to **convert the generated number to 0 or 1, maintaining the equally probable property**.

4. Repeat Step 3 (i.e., generate another 100 random numbers) and count the number of heads in those 100 flips.

5. If the differences in the counts from Steps 3 and 4 is greater than or equal to the user input difference d, then stop. Otherwise repeat Steps 3 and 4 and check if the difference of the counts of these new samples is greater than or equal to d. If it is, stop. Continue this process until the difference in the counts is greater than or equal to d, or Steps 3 and 4 have been repeated 8 times.

Here is an example run:

```
Input the random seed: 346
Input the difference: 8

       Set 1  Set 2

 1     54     56

 2     46     49

 3     51     52

 4     45     48

 5     55     41
```

Test your program a variety of times to make sure its answers are correct, as best you can tell. When you have written, tested, and corrected your solution, save your file as <username>_3A.cpp, where you replace <username> with your U of M username. Submission information is given after Problem B. As usual, be diligent in following this naming convention.

Problem B: Variation on the Coin Flipping

Repeat the process in Problem A except make the following changes:

1. The user input difference d should be a double rather than an int.
2. Instead of generating a 0 or a 1 each time, generate a random number that is a double in the range [40.0, 100.0]. Each number in this range should be equally likely.
3. Print the averages with 2 digits to the right of the decimal point.
4. Instead of the stopping criterion in Problem A, always run 10 repetitions. However, after all 10 are completed, print out how many of them had their averages differ by more than the user input difference d.
5. The entire process (after setting the random seed) should be in a do - while continuation loop. So the program should always run the 10 repetitions mentioned in Step 4. Then the program should ask the user if they wish to repeat the process. If the user inputs a ‘y’ or ‘Y’, then the program should repeat this process. The program should continue in this vein until the user enters a character other than ’y’ or ’Y’.

Other than these changes, your Problem B program should be the same as Problem A. (Hint: copy and modify your Problem A solution rather than writing this part from scratch. However, make sure you save your Problem A solution and in this part work with the copy.)

(Continues next page)

Here is an example of running the program:

```
Input the random seed: 346
Input the difference: 3.5

       Set 1  Set 2
 1     70.76  66.28
 2     69.93  69.29
 3     71.05  70.03
 4     74.08  71.96
 5     68.21  66.04
 6     73.45  67.48
 7     68.62  69.44
 8     70.21  67.22
 9     67.82  67.51
 10    72.87  70.97

Number that exceeded 3.50: 2
Repeat the process (y/n)? y
Input the difference: 4.2

       Set 1  Set 2
 1     70.73  65.68
 2     69.80  71.54
 3     70.65  68.06
 4     68.30  70.36
 5     68.83  70.35
 6     70.94  68.92
 7     71.29  73.50
 8     66.92  68.72
 9     69.97  70.17
 10    67.99  68.70

Number that exceeded 4.20: 1
Repeat the process (y/n)? n
```

Test your program a variety of times to make sure its answers are correct, as best you can tell. When you have written, tested, and corrected your solution, save your file as <username>_3B.cpp, where you replace <username> with your U of M username, and submit it together with the Problem A source file using the Homework 3 link in Moodle. As usual, be diligent in following this naming convention. Please note, both files must be uploaded separately in Moodle, and you can only upload at most two files.