CS 12 - Assignment 1: Noise Signals

Collaboration Policy

You may not use code from any source (another student, a book, online, etc.) within your solution to this assignment. In fact, you may not even look at another student's solution or partial solution to this assignment. You also may not allow another student to look at any part of your solution to this exercise. You should get help on this assignment by coming to the instructor's or TA's office hours or by posting questions on Piazza (you still must not post assignment code publically on Piazza.) See the full Course Collaboration Policy here: Collaboration Policy

---

Assignment specs:

In this assignment, you will demonstrate your knowledge of **arrays**. You are not allowed to use vectors anywhere in your code. In fact, do not use the word vector anywhere, including comments.

In engineering simulations, we often want to generate a floating-point sequence of values with a specified mean and variance. The randFloat function below allows us to generate a random sequence between limits a and b, but it does not allow us to specify the mean and variance. By using results from probability, the following relationships can be derived between the limits of a uniform random sequence and its theoretical mean μ and variance σ^2:

$$\mu = \frac{(a+b)}{2} \qquad \sigma^2 = \frac{(b-a)^2}{12}$$

```
/*This function generates a random double value between a and b*/
double randFloat (double a, double b)
{
   return a + (static_cast<double>(rand()) / RAND_MAX) * (b - a);
}
```

Part 1:

Write a program that uses the randFloat function given above to generate sequences of random floating-point values between 4 and 10. You should now use two specific sized sequences; sequences of 100 and 10,000. Then compare the computed

mean and variance to the theoretical values computed (using formulas above).

There should be no user input for this section.

The expected output should be seen for both sequences: theoretical mean, practical mean, theoretical variance, practical variance. The values should be separated by spaces.

Example:
Given: theoretical mean = 2.0, practical mean = 2.1, theoretical variance = .75, and practical variance = .76

The output would be: 2.0 2.1 0.75 0.76

Output should be seen with this output style for both sequences, each started on a newline.

Part 2:

Write a program that uses the randFloat function given above to generate two sequences of 500 points. Each sequence should have a theoretical mean of 4, but one sequence should have a variance of 0.5 and the other should have a variance of 2. Check the computed means and variances and compare to the theoretical means and variances. (Hint: Use the two given equations to write two equations with two unknowns. Then solve for the unknowns by hand.)

The practical and theoretical values for each sequence should be output as in Exercise 1.

Part 3:

Write a program that uses the randFloat function to generate two sequences of 500 points. Each sequence should have the same variance of 3.0 but one sequence should have a mean of 0.0 and the other should have a mean of -4.0. Compare the computed and theoretical values of mean and variance.
(Hint: Use the two given equations to write two equations with two unknowns. Then solve for the unknowns by hand.)

There should be no user input for this section

The output should contain the theoretical and practical values as in Exercise 1 for both sequences.

Part 4:

Write a function named rand_mv that generates a random floating-point value. rand_mv should take in a user specified mean and variance. Assume the corresponding function prototype is

```
double rand_mv(double mean, double var);
```

rand_mv() should then calculate A and B, and generate a random value using `randFloat(a,b)`

The main function will provide two prompts, the first asking for the desired mean from the user.

Note: The values denoted as x.xx represent a value of type float in all cases.
It is not a number formatted specifically as X.XX So if the output is 4.99999 it should remain 4.99999 and not be formatted to look like 4.99

Enter Mean: x.xx

The second prompt will ask for the variance.
Enter Variance x.xx

Finally, in order for the test harness in R'Sub to know if you have correctly implemented this function your randMV function must output to the terminal the A and B values computed by randMV.

Note: Normally you would not have randMV output these values. This is just so we can test your solution. Even though we are not testing the random value that your function returns, you should still have randMV return this value, since that really is the purpose of this function.

Example: if your randMV function computes the following: A = 2.0 and B = 4.0
We should have output to the terminal as follows:
2.0 4.0