# CPSC 121 Assignment 5 - Loops

Your task is to finish the starter code given, performing various mathematical tasks.

**You are not allowed to use any math libraries (built-in math functions) from C++, you must do all the calculations yourself.**

## Case 1: Calculate the nth Fibonacci number

Fibonacci numbers are found everywhere in nature.

They are a series of numbers such that the next number in the sequence is the sum of the previous two.

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, …

It can be expressed by the following:

$fib(0) =$
$fib(1) = 1$
$fib(n) = (n - 1) + (n - 2)$

Case 1 Details:
Prompt the user for an integer n, then calculate the nth fibonacci number and print to the console.

E.g.
If the user enters 0, print 0
If the user enters 1, print 1
If the user enters 2, print 1
If the user enters 3, print 2
If the user enters 4, print 3
If the user enters 10, print 55

## Case 2: Prime number printer

Prompt the user for a number, then print all the prime numbers up to and including that number.

E.g.
If the user enters 10, the program prints 2, 3, 5, 7
If the user enters 23, the program prints 2, 3, 5, 7, 11, 13, 17, 19, 23
And so on…

## Case 3: Random number tester

Count how long it takes for the computer to generate two of the same random numbers.

Prompt the user for an upper bound, then calculate a random number [0, upper bound].
The program will then repeatedly generate another random number within the same range, and count how many generations it takes to produce the same random number again.

This case will then output the count to the user.

E.g.

1. The user enters 100
2. The program then generates a random number between 0 and 100 inclusive, for example the number 73.
3. The program will then repeatedly generate random numbers between 0 and 100 inclusive, until it generates the same random number as in step 2, in our example 73.
4. The program will then display how many numbers were generated before repeating the same number twice (73 and 73)

srand has been provided already, seeding the random number generator with time(NULL). You must use rand() for this case

## Case 4: Exponent Calculator

This case will have you implement the pow function by hand. Prompt the user for a base and an exponent, then output the result to the console. Inputs will be integers.

E.g.
Enter a base: 2
Enter an exponent: 3
Answer: 8

## Case 5: Approximate pi

This case will have you approximate pi with varying accuracy. You can approximate pi with the following equation:

$$\pi \approx \frac{4}{1} - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \ \ldots$$

The more terms you calculate the closer you will get to the value pi.

Prompt the user for a value n, then calculate the approximate value of pi to the n$^{th}$ term using the formula above. Output the approximate value of pi to the console.

**Case 6: Exit the program**

This case exits the program gracefully.

**Additional Notes:**

Write your code between the comments for each case, do not modify the code other than adding to the cases provided.

Rubric: 60 Points total

| | |
|---|---|
| Code compiles without errors or warnings | 0 Points max score if program doesn't compile<br>-5 points if there are any warnings |
| Case 1 | 10 Points |
| Case 2 | 10 Points |
| Case 3 | 10 Points |
| Case 4 | 10 Points |
| Case 5 | 10 Points |
| Case 6 | 5 Points |
| Comments | 5 points - Use appropriate comments throughout the code. Every 5-10 lines of code should have at least one comment explaining what the code is doing. |

Submission:

Submit the completed LoopPractice.cpp file and Makefile in a .zip folder to blackboard by the due date. Name your .zip folder "lastname_firstname_assignment5.zip"