**Name:**

**Advanced Programming in C++**

**Lab Exercise 4/21/2023**

**Introduction to the BigInt Data Type**

In this activity, you will explore the BigInt data type. One of the major shortcomings with regard to high level programming languages is the Integer data type. In 16-bit systems, the Integer data type could only hold numbers from −32,768 to 32,767. 32-bit systems were better but were still limited to the range of  −2,147,483,648 to 2,147,483,647. The BigInt data type will allow us to work with integer values limited only by the system memory.

In our C++ compiler, we have 3 types of integers; int, long, and long long.

Type in the following program to see what size these integers are.

#include <iostream>

using namespace std;

int main()

{

int number;

long number2;

long long number3;

cout << "Size of int = " << sizeof(number) << " bytes" << endl;

cout << "Size of long = " << sizeof(number2) << " bytes" << endl;

cout << "Size of long long = " << sizeof(number3) << " bytes" << endl;

return 0;

}

Two problems we have done it the past that was severely limited were the factorial and Fibonacci problems. You will write four functions and a driver. When you have completed the functions, submit your source code.

1. Rewrite the Fibonacci function using an iterative technique using the BigInt class.
2. Rewrite the Fibonacci function using a recursive technique using the BigInt class.
3. Rewrite the factorial function using an iterative technique using the BigInt class.
4. Rewrite the factorial function using a recursive technique using the BigInt class.