```cpp
/************************************************************
* AUTHOR    : Ali Eshghi
* STUDENT ID    : 1112261
* ASSIGNMENT #4 : Assessing Recursion Performance
* CLASS         : CS 1B
* SECTION       : MW - 7:30 pm - 9:50 pm
* DUE DATE      : 11/26/2019
*************************************************************/

#include "MyHeader.h"

/************************************************************
 * ASSESSING RECURSION PERFORMANCE
 *_________________________________________________________
 * This program allows a user to compare the performance of
 *  a recursion based algorithm against to an iteration based
 *  (loop) algorithm. the program gets an option from the user
 *  that he wants to get the factorial or fibbenacci result
 *  of a number and returns the result of the position of
 *  that number in whichever function the user chooses
 *  (factorial or fibbenacci)
 *_________________________________________________________
 * INPUT:
 *   userInp: user's choice of number.
 *   option : user's choice in the main menu.
 *
 * OUTPUT:
 *   -
 ************************************************************/

int main()
{

    /********************************************************
    * ----------------------------------------------------
    * USED FOR CLASS HEADING - ALL WILL BE OUTPUT
    * ----------------------------------------------------
    * Type     : Program Type
    * LAB_NUM  : Lab Number (specific to this lab)
    * LAB_NAME : Title of the Lab
    *********************************************************/

    const int  LAB_NUM      = 0;
    const char TYPE         = 'A';
    const char LAB_NAME[50] = "Assignment 4 - Recursive";


    /*************
     *  VARIABLE *
     *************/

    int option;      //IN & PROCESS - user menu choice
    int userInp; //IN & PROCESS & OUT - user input number
    long factResult; //OUT - factorial of the user input number
    long fibResult;  //OUT - fibbonacci result of the user input number




    //This function will print the header
    PrintHeader(LAB_NAME, TYPE, LAB_NUM);


    //this function will get the user input for the menu option
    GetCheckInp(option);



    //while loop for user's choice of the menu
    while(option != 0)
    {
        switch(option)
        {
            //this option gets the factorial of the number
            case FACTLOOP: userInp = CheckInpFact();
                           cout << endl << userInp << "! = ";
                           factResult = Fact(userInp);
                           cout << " = " << factResult;
                           cout << endl << endl;
                           break;

            //This function gets the fibbonacci result using recursive function
            case FIBLOOP: userInp = CheckInpFact();
                          cout << endl;
                          fibResult = SeriesFib(userInp);
                          cout << "fib(" << userInp << ") = " << fibResult;
                          cout << endl << endl;
                          break;


            //FACTORIAL RECORD
            case FACTREC:
                        {
                         userInp = CheckInpFact();
                          //RECURSIVE
                          high_resolution_clock::time_point
                          t1 = high_resolution_clock::now();
                          Fact(userInp);
                          high_resolution_clock::time_point
                          t2 = high_resolution_clock::now();
                          auto duration = duration_cast<microseconds>
                          ( t2 - t1 ).count();
                          cout << "It took the program " << duration
                               << " microseconds to execute with recursive.";
                          cout << "\n";

                          //LOOP

                          high_resolution_clock::time_point
                          t3 = high_resolution_clock::now();
                          FactLoop(userInp);
                          high_resolution_clock::time_point
                          t4 = high_resolution_clock::now();
                          auto duration1 = duration_cast<microseconds>
                          ( t4 - t3 ).count();
                          cout << "It took the program " << duration1
                               << " microseconds to execute with loops.";
                          cout << "\n";

                          break;
                    }

            //FIBONACCI RECORD

            case FIBREC:
                       {
                             userInp = CheckInpFact();

                             //RECURSIVE

                             high_resolution_clock::time_point
                             t5 = high_resolution_clock::now();
                             SeriesFib(userInp);
                             high_resolution_clock::time_point
                             t6 = high_resolution_clock::now();
                             auto duration2 = duration_cast<microseconds>
                             ( t6 - t5 ).count();
                             cout << "It took the program " << duration2
                                  << " microseconds to execute with recursive.";
                             cout << "\n";

                             //LOOP

                             high_resolution_clock::time_point
                             t7 = high_resolution_clock::now();
                             SeriesFibLoop(userInp);
                             high_resolution_clock::time_point
                             t8 = high_resolution_clock::now();
                             auto duration3 = duration_cast<microseconds>
                             ( t8 - t7 ).count();
                             cout << "It took the program " << duration3
                                  << " microseconds to execute with loops.";
                             cout << "\n";

                             break;
                       }

        } //END of switch statement

        //this function will get the user input for the menu option
        GetCheckInp(option);

    } //END of ehile loop

```

```
 return 0;
}

```