```
 1 ************************************************
 2 *  Programmer    : Ali Eshghi
 3 *  STUDENT ID    : 1112261
 4 *  CLASS         : MW - 7:30-9:50
 5 *  ASSIGNMENT #0 : Assignment 4 - Recursive
 6 ************************************************
 7
 8
 9
10
11 MENU OPTIONS
12
13 1 – Calculate and Display Factorial of a Number
14 2 – Calculate and Display Fibonacci Series of a Number
15 3 – Compare Performance for Factorial Implementations
16 4 – Compare Performance for Fibonacci Series Implementations
17 0 –EXIT
18 Enter an userOption (0 to exit): 1
19 Enter the target number: 8
20
21 8! =  = 40320
22
23
24
25 MENU OPTIONS
26
27 1 – Calculate and Display Factorial of a Number
28 2 – Calculate and Display Fibonacci Series of a Number
29 3 – Compare Performance for Factorial Implementations
30 4 – Compare Performance for Fibonacci Series Implementations
31 0 –EXIT
32 Enter an userOption (0 to exit): 2
33 Enter the target number: 11
34
35 fib(11) = 89
36
37
38
39 MENU OPTIONS
40
41 1 – Calculate and Display Factorial of a Number
42 2 – Calculate and Display Fibonacci Series of a Number
43 3 – Compare Performance for Factorial Implementations
44 4 – Compare Performance for Fibonacci Series Implementations
45 0 –EXIT
46 Enter an userOption (0 to exit): 3
47 Enter the target number: 7
48 It took the program 0 microseconds to execute with recursive.
49 It took the program 0 microseconds to execute with loops.
50
51
52 MENU OPTIONS
53
54 1 – Calculate and Display Factorial of a Number
55 2 – Calculate and Display Fibonacci Series of a Number
```

```
3 – Compare Performance for Factorial Implementations
4 – Compare Performance for Fibonacci Series Implementations
0 –EXIT
Enter an userOption (0 to exit): 3
Enter the target number: 12
It took the program 0 microseconds to execute with recursive.
It took the program 0 microseconds to execute with loops.


MENU OPTIONS

1 – Calculate and Display Factorial of a Number
2 – Calculate and Display Fibonacci Series of a Number
3 – Compare Performance for Factorial Implementations
4 – Compare Performance for Fibonacci Series Implementations
0 –EXIT
Enter an userOption (0 to exit): 4
Enter the target number: 12
It took the program 1 microseconds to execute with recursive.
It took the program 0 microseconds to execute with loops.


MENU OPTIONS

1 – Calculate and Display Factorial of a Number
2 – Calculate and Display Fibonacci Series of a Number
3 – Compare Performance for Factorial Implementations
4 – Compare Performance for Fibonacci Series Implementations
0 –EXIT
Enter an userOption (0 to exit): 4
Enter the target number: 45


It took the program 8646052 microseconds to execute with recursive.
It took the program 0 microseconds to execute with loops.


MENU OPTIONS

1 – Calculate and Display Factorial of a Number
2 – Calculate and Display Fibonacci Series of a Number
3 – Compare Performance for Factorial Implementations
4 – Compare Performance for Fibonacci Series Implementations
0 –EXIT
Enter an userOption (0 to exit):
```

```
 1 /*************************************************************
 2 * AUTHOR    : Ali Eshghi
 3 * STUDENT ID    : 1112261
 4 * ASSIGNMENT #4 : Assessing Recursion Performance
 5 * CLASS         : CS 1B
 6 * SECTION       : MW - 7:30 pm - 9:50 pm
 7 * DUE DATE      : 11/26/2019
 8 *************************************************************/
 9
10 #ifndef MYHEADER_H_
11 #define MYHEADER_H_
12
13
14
15 #include <iostream>
16 #include <iomanip>
17 #include <string>
18 #include <cstdlib>
19 #include <chrono>
20 #include <ctime>
21 using namespace std;
22 using namespace std::chrono;// using namespace std::chrono
23
24 /*************
25  *  VARIABLE *
26  *************/
27
28 enum MenuOption //PROCESS - used for the menu
29 {
30     EXIT,
31     FACTLOOP,
32     FIBLOOP,
33     FACTREC,
34     FIBREC
35 };
36
37
38 /*****************************************************************
39  * PrintHeader
40  *    This function receives receives an assignment name, type
41  *     and number then outputs the appropriate header
42  *
43  *    returns  nothing -> This will output the class heading.
44  *****************************************************************/
45 void PrintHeader(string asName,  // IN - assignment Name
46                  char   asType,  // IN - assignment type
47                  int    asNum);  // IN - assignment number
48
49
50 /*****************************************************************
51  * Menu
52  *    This function will print the program menu into the console.
53  *
54  *    returns  nothing -> This will output the menu.
55  *****************************************************************/
```

```
56 void Menu();
57
58
59 /*****************************************************************
60  * GetCheckInput
61  *    This function will get the user's option from the menu and
62  *     returns it into the main.
63  *
64  *    returns  nothing -> This will output the menu.
65  *****************************************************************/
66 void GetCheckInp(int &userOption);  //IN & PROCESS - user's input
67
68
69 /*****************************************************************
70  * CheckInpFact
71  *    This function gets the user's input and pass
72  *     it back to the main.
73  *
74  *    returns  integer -> target number for fib/fact
75  *****************************************************************/
76 int CheckInpFact(); //IN & PROCESSS - user's input
77
78 /*****************************************************************
79  * Fact
80  *    This function will calculate the factorial of user's number.
81  *
82  *    returns  integer -> factorial of the target number
83  *****************************************************************/
84 int Fact(int userInt);  //IN & PROCESS - user's input
85
86
87 /*****************************************************************
88  * SeriesFib
89  *    This function will calculate the fibbonacci number of user's number
90  *    input
91  *
92  *    returns  integer -> fibbonacci result of the target number
93  *****************************************************************/
94 int SeriesFib(int userInp); //IN & PROCESS - user's input
95
96 /*****************************************************************
97  * FactLoop
98  *    This function will calculate the factorial number of user's number
99  *    input using a loop
100 *
101 *    returns  integer -> factorial result of the target number
102 *****************************************************************/
103 int FactLoop(int userInp);  //IN & PROCESS - user's input
104
105 /*****************************************************************
106  * SeriesFibLoop
107  *    This function will calculate the fibbinacci result number of
108  *    user's number input using a loop
109  *
110  *    returns  integer -> fibbonacci result of the target number
```

```
111  ***************************************************************/
112 int SeriesFibLoop(int userInp); //IN & PROCESS - user's input
113
114 #endif /* MYHEADER_H_ */
115
```

```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()
{

    /*************************************************************
    * CONSTANTS
    * ----------------------------------------------------------
    * 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
```

```
56      long fibResult;  //OUT - fibbonacci result of the user input number
57
58
59
60
61      //This function will print the header
62      PrintHeader(LAB_NAME, TYPE, LAB_NUM);
63
64
65      //this function will get the user input for the menu option
66      GetCheckInp(option);
67
68
69
70      //while loop for user's choice of the menu
71      while(option != 0)
72      {
73          switch(option)
74          {
75              //this option gets the factorial of the number
76              case FACTLOOP: userInp = CheckInpFact();
77                             cout << endl << userInp << "! = ";
78                             factResult = Fact(userInp);
79                             cout << " = " << factResult;
80                             cout << endl << endl;
81                             break;
82
83              //This function gets the fibbonacci result using recursive function
84              case FIBLOOP: userInp = CheckInpFact();
85                            cout << endl;
86                            fibResult = SeriesFib(userInp);
87                            cout << "fib(" << userInp << ") = " << fibResult;
88                            cout << endl << endl;
89                            break;
90
91
92              //FACTORIAL RECORD
93              case FACTREC:
94                          {
95                           userInp = CheckInpFact();
96                            //RECURSIVE
97                            high_resolution_clock::time_point
98                            t1 = high_resolution_clock::now();
99                            Fact(userInp);
100                           high_resolution_clock::time_point
101                           t2 = high_resolution_clock::now();
102                           auto duration = duration_cast<microseconds>
103                           ( t2 - t1 ).count();
104                           cout << "It took the program " << duration
105                                << " microseconds to execute with recursive.";
106                           cout << "\n";
107
108                           //LOOP
109
110                           high_resolution_clock::time_point
```

```
111                         t3 = high_resolution_clock::now();
112                         FactLoop(userInp);
113                         high_resolution_clock::time_point
114                         t4 = high_resolution_clock::now();
115                         auto duration1 = duration_cast<microseconds>
116                         ( t4 - t3 ).count();
117                         cout << "It took the program " << duration1
118                              << " microseconds to execute with loops.";
119                         cout << "\n";
120
121                         break;
122                 }
123
124             //FIBONACCI RECORD
125
126             case FIBREC:
127                         {
128                               userInp = CheckInpFact();
129
130                               //RECURSIVE
131
132                               high_resolution_clock::time_point
133                               t5 = high_resolution_clock::now();
134                               SeriesFib(userInp);
135                               high_resolution_clock::time_point
136                               t6 = high_resolution_clock::now();
137                               auto duration2 = duration_cast<microseconds>
138                               ( t6 - t5 ).count();
139                               cout << "It took the program " << duration2
140                                    << " microseconds to execute with recursive.";
141                               cout << "\n";
142
143                               //LOOP
144
145                               high_resolution_clock::time_point
146                               t7 = high_resolution_clock::now();
147                               SeriesFibLoop(userInp);
148                               high_resolution_clock::time_point
149                               t8 = high_resolution_clock::now();
150                               auto duration3 = duration_cast<microseconds>
151                               ( t8 - t7 ).count();
152                               cout << "It took the program " << duration3
153                                    << " microseconds to execute with loops.";
154                               cout << "\n";
155
156                               break;
157                         }
158
159         } //END of switch statement
160
161         //this function will get the user input for the menu option
162         GetCheckInp(option);
163
164     } //END of ehile loop
165
```

```
166     return 0;
167    }
168
```

```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"

/*****************************************************************
 * CheckInpFact
 *    This function gets the user's input and pass
 *     it back to the main.
 *
 *    returns  integer -> target number for fib/fact
 *****************************************************************/

int CheckInpFact()
{
    /*************
     *  VARIABLE *
     *************/

    int factInp;   //IN - user's target number

    bool checkInput;   //PROCESS - input checking

    /****************
     *  INITIALIZE  *
     ****************/

    checkInput = false;


    /*************
     *   INPUT   *
     *************/

    //do while loop for error checking
    do
    {

        cout << "Enter the target number: ";

        //CHECK INPUT

        if (!(cin >> factInp))
        {
            cin.clear(); cin.ignore(numeric_limits<streamsize>::max(), '\n');

            cout << "\n**** Please input a NUMBER ****\n";

            checkInput = false;
```

```
56
57          }
58
59          //CHECK FOR RANGE
60          else if (factInp < 0)
61          {
62              cout << "\n****   The number " << factInp
63                   << " is an invalid entry   ****\n"
64                   << "**** Please input a positive ****\n";
65
66              checkInput = false;
67          }
68
69          else
70          {
71              cin.ignore(numeric_limits<streamsize>::max(), '\n');
72
73              checkInput = true;
74          }
75
76      } while (!checkInput);
77
78      //returns integer to the main
79      return factInp;
80  }
81
82
```

```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"

/*****************************************************************
 * factNum
 *    This function will calculate the factNumorial of user's number.
 *
 *    returns  integer -> factNumorial of the target number
 *****************************************************************/

int Fact(int userInp)   //IN & PROCESS - user's input
{

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

    long long factNum;      //PROCESS & OUT - factNumorial of user input number

    /***************
     *  INITIALIZE *
     ***************/

    factNum = userInp;

    //if statement for user input
    if(userInp == 0 || userInp == 1)
    {
        factNum = 1;
        return factNum;
    }

    else
    {
        factNum = factNum * Fact(userInp - 1);
        return factNum;
    }
}

```

```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"

/*****************************************************************
 * FactLoop
 *    This function will calculate the factorial number of user's number
 *    input using a loop
 *
 *    returns  integer -> factorial result of the target number
 *****************************************************************/

int FactLoop(int userInp)   //IN & PROCESS - user's input
{

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


    long long factTot;  //PROCESS & OUT - factorial of the user input number
    int i;          //PROCESS - loop LCV

    //if statement for 1 or 0 input
    if(userInp == 0 || userInp == 1)
    {
        factTot = 1;
        return factTot;
    }

    else
    {
        for(i = 2; i <= userInp ; i++)
        {
            factTot = factTot * i;
        }

        //returns an integer to the main
        return factTot;
    }
}


```

```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"

/*****************************************************************
 * GetCheckInput
 *    This function will get the user's option from the menu and
 *     returns it into the main.
 *
 *    - returns  nothing -> This will output the menu.
 *****************************************************************/

void GetCheckInp(int &option)  //IN & PROCESS - user's input
{
    /*************
     *  VARIABLE *
     *************/

    bool checkInput;   //PROCESS - input checking

    /***************
     *  INITIALIZE *
     ***************/

    checkInput = false;


    //do while loop for error checking
    do
    {
        //this function prints menu
        Menu();
        cout << "Enter an userOption (0 to exit): ";

        //CHECK INPUT
        if (!(cin >> option))
        {
            cin.clear(); cin.ignore(numeric_limits<streamsize>::max(), '\n');
            cout << "\n**** Please input a NUMBER between 0 and 5 ****\n";
            checkInput = false;
        }

        //CHECK FOR RANGE
        else if (option > 4 || option < 0)
        {
            cout << "\n****   The number " << option
                 << " is an invalid entry   ****\n"
                 << "**** Please input a number between 0 and 5 ****\n";
```

```
56
57              checkInput = false;
58          }
59
60          else
61          {
62              cin.ignore(numeric_limits<streamsize>::max(), '\n');
63              checkInput = true;
64          }
65
66      } while (!checkInput);
67
68  }
69
70
71
```

```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"

/*********************************************************************
 * Menu
 *    This function will print the program menu into the console.
 *
 *    returns  nothing -> This will output the menu.
 *********************************************************************/

void Menu()
{

    cout << endl << endl;
    cout << left;
    cout << "MENU OPTIONS\n\n";
    cout << "1 - Calculate and Display Factorial of a Number\n";
    cout << "2 - Calculate and Display Fibonacci Series of a Number\n";
    cout << "3 - Compare Performance for Factorial Implementations\n";
    cout << "4 - Compare Performance for Fibonacci Series Implementations\n";
    cout << "0 -EXIT\n";
    cout << right;

}

```

```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"

/*****************************************************************
 * PrintHeader
 *    This function receives receives an assignment name, type
 *     and number then outputs the appropriate header
 *
 *    returns  nothing -> This will output the class heading.
 *****************************************************************/

void PrintHeader(string asName,  // IN - assignment Name
                 char   asType,  // IN - assignment type
                 int    asNum)   // IN - assignment number
{

    cout << left;
    cout << "*****************************************************\n";
    cout << "*  Programmer : Ali Eshghi\n";
    cout << "*  " << setw(14) << "STUDENT ID" << ": 1112261\n";
    cout << "*  " << setw(14) << "CLASS"      << ": MW - 7:30-9:50\n";
    cout << "*  ";

    if (toupper(asType) == 'L')
    {
        cout << "LAB #" << setw(9);
    }

    else
    {
        cout << "ASSIGNMENT #" << setw(2);
    }

    cout << asNum << ": " << asName << endl;
    cout << "*****************************************************\n\n";
    cout << right;

}
```

```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"

/*****************************************************************
 * SeriesFib
 *    This function will calculate the fibbonacci number of user's number
 *    input
 *
 *    returns  integer -> fibbonacci result of the target number
 *****************************************************************/

int SeriesFib(int userInp)  //IN & PROCESS - user's input
{
    /*************
     *  VARIABLE *
     *************/

    long long fib;      //PROCESS & OUT- fibonacci series of user input num


    /***************
     *  INITIALIZE *
     ***************/

    fib = userInp;

    //if statement for 0 or 1 input
    if(userInp == 0 || userInp == 1)
    {
        return fib;
    }

    else
    {
        fib = SeriesFib(fib - 1) + SeriesFib(fib - 2);
        return fib;
    }
}
```

```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"

/*****************************************************************
 * SeriesFibLoop
 *    This function will calculate the fibbinacci result number of
 *    user's number input using a loop
 *
 *    returns  integer -> fibbonacci result of the target number
 *****************************************************************/

int SeriesFibLoop(int userInp)  //IN & PROCESS - user's input
{
    /*************
     *  VARIABLE *
     *************/

    //INTEGERS

    long long i;        //PROCESS & OUT - fibonacci series of user input num
    long long x;        //PROCESS - number for n - 1
    long long y;        //PROCESS - number for n - 2
    int  z;

    /***************
     *  INITIALIZE *
     ***************/

    i = 1;
    x = 0;
    y = 1;
    z = 2;

    //While loop for number of times
    while (z <= userInp)
    {
        x = y;
        y = i;

        i = x + y;

        z++;
    }

    //returns an integer to the main
    return i;

```

```
56 }
57
```