

main.cpp

```
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 #include "MyHeader.h"
11
12 /*****
13 * ASSESSING RECURSION PERFORMANCE
14 *
15 * This program allows a user to compare the performance of
16 * a recursion based algorithm against to an iteration based
17 * (loop) algorithm. the program gets an option from the user
18 * that he wants to get the factorial or fibbenacci result
19 * of a number and returns the result of the position of
20 * that number in whichever function the user chooses
21 * (factorial or fibbenacci)
22 *
23 * INPUT:
24 *   userInp: user's choice of number.
25 *   option : user's choice in the main menu.
26 *
27 * OUTPUT:
28 *   -
29 *****/
30
31 int main()
32 {
33
34     /*****
35     * CONSTANTS
36     * -----
37     * USED FOR CLASS HEADING - ALL WILL BE OUTPUT
38     * -----
39     * Type      : Program Type
40     * LAB_NUM    : Lab Number (specific to this lab)
41     * LAB_NAME   : Title of the Lab
42     *****/
43
44     const int LAB_NUM      = 0;
45     const char TYPE        = 'A';
46     const char LAB_NAME[50] = "Assignment 4 - Recursive";
47
48
49     /*****
50     * VARIABLE *
51     *****/
52
53     int option;          //IN & PROCESS - user menu choice
54     int userInp;         //IN & PROCESS & OUT - user input number
55     long factResult;     //OUT - factorial of the user input number
```

main.cpp

```
56  long fibResult; //OUT - fibonacci 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 fibonacci 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
```

main.cpp

```
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
```

main.cpp

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