```cpp
#include "header.h"
/*******************************************************************************
 * This function will find the factorial of an int num
 *------------------------------------------------------------------------------
 * INPUT:
 *      num - long long integer
 * OUTPUT:
 *      factorial of num
 *******************************************************************************/
long long factorial(long long num)
{
    long long factorial; //OUT - holds factorial of num

    factorial = num;

    if (num <= 1)
    {
       factorial = 1;
    }
    while(num-1 > 0)
    {
        factorial = factorial * (--num);
    }
    return factorial;
}

/*******************************************************************************
 * This function will output the fibonachi series, up to (num) numbers
 *------------------------------------------------------------------------------
 * INPUT:
 *      num - number of numbers in series to calculate
 * OUTPUT:
 *      fibonachi series
 *******************************************************************************/
string fib(long num)
{
    int i;          //CALC - int used in for loop
    long fib;       //CALC - stores result of calculation
                    //       for next number in series
    long series[50];//CALC - array of fib series
    string output;  //OUT  - string of series to output

    if(num <= 1)
    {
        series[0] = 1;
    }
    else
    {
        fib = 0;
        for(i = 0;i < num;i++)
        {
```

```
52            if(i == 0)
53            {
54                series[i] = 0;
55            }
56            else if (i == 1)
57            {
58                series[i] = 1;
59            }
60            else
61            {
62                series[i] = series[i-1] + series[i-2];
63            }
64        }
65    }
66    return outputArray(series, num);
67}
68
69/*****************************************************************************
70 * This function will get a menu infut from the user and error check the input
71 *----------------------------------------------------------------------------
72 * INPUT:
73 *      NA
74 * OUTPUT:
75 *      menuOpt
76 *****************************************************************************/
77int menuInput()
78{
79    int menuOpt;
80    bool invalid = false;
81    do
82    {
83        outputMenu();
84        if(!(cin >> menuOpt))
85        {
86            cout << "\n**** Please input a number between 0 and 4 ****\n";
87            cin.clear();
88            cin.ignore(numeric_limits<streamsize>::max(), '\n');
89            invalid = true;
90        }
91        else if(menuOpt < 0 || menuOpt > 4)
92        {
93            cout << "\n**** The number " <<  menuOpt << " is an invalid entry ****\n";
94            cout << "**** Please input a number between 0 and 4 ****\n";
95            invalid =  true;
96        }
97        else
98        {
99            cin.ignore(1000, '\n');
100           invalid = false;
101       }
102   }while(invalid);
```

```
103
104     return menuOpt;
105 }
106 /*****************************************************************************
107  * This function will output a menu of options for the user to pick from
108  *----------------------------------------------------------------------------
109  * INPUT:
110  *      NA
111  * OUTPUT:
112  *      NA
113  *****************************************************************************/
114 void outputMenu()
115 {
116     cout << "\n1 - Calculate and Display Factorial of a Number\n";
117     cout << "2 - Calculate and Display Fibonachi Series of a Number\n";
118     cout << "3 - Compare Performance for Factorial Implementations\n";
119     cout << "4 - Compare Performance for Fibonachi Implementations\n";
120     cout << "0 - Exit\n";
121     cout << "enter a command (0 to exit): ";
122 }
123
124 /*****************************************************************************
125  * This function will output a series of numbers from an array
126  *----------------------------------------------------------------------------
127  * INPUT:
128  *      arr[]   - array of numbers to output
129  *      num     - number of numbers to output from array
130  * OUTPUT:
131  *      NA
132  *****************************************************************************/
133 string outputArray(long arr[], int num)
134 {
135     string output;
136     int i;
137     for(i=0;i<num;i++)
138     {
139         output += to_string(arr[i]);
140         if(i < num-1)
141         {
142             output += ",";
143         }
144     }
145     return output;
146 }
147
148 /*****************************************************************************
149  * This function will output the class header using ostream
150  *----------------------------------------------------------------------------
151  * INPUT:
152  *      output      - output file variable
153  *      exersize    - Lab or Assignment
```

```
154 *      exersizeName- name of exersize
155 *      num         - number of Lab/Assignment
156 *      names       - names of programmers
157 * OUTPUT:
158 *      header
159 ********************************************************************************/
160 void PrintHeader(ostream &output, char exersize, string exersizeName,  int num, string
    names)
161 {
162
163     int colWidth;  //CALC - changes based on exersize
164     string asType; //CALC - changes based on exersize
165
166     if(exersize == 'L')
167     {
168         asType = "Lab";
169         colWidth = 9;
170     }
171     else
172     {
173         asType = "Assignment";
174         colWidth = 2;
175     }
176
177     output << left;
178     output <<"************************************************\n";
179     output <<"* PROGRAMMED BY : "  << names << endl;
180     output <<"* "<< setw(14) << "CLASS"   << ": " << "CS1A"    << endl;
181     output <<"* "<< setw(14) << "SECTION" << ": " << "MW: 7:30P"  << endl;
182     output <<"* "<< asType << " #" << setw(colWidth)  << num   << ": " << exersizeName
    << endl;
183     output <<"************************************************\n";
184     output << right;
185 }
186
187
```