55 pts

Name1: Amirarsalan Valipour

Name2: Ali Eshghi

Class Day / Time: MW - 7:30 pm

Due Date: 09 / 24 / 2019

Lab #5: Binary Search

In this lab, you will perform the tasks bellow. DO NOT USE GLOBAL CONSTANTS!

1. Create a header file that contains the following.
   - All necessary pre-processor directives
   - The prototype for a function that sorts an array using an insertion sort.
   - The prototype for a function that searches an array using a sequential search and returns the appropriate index in the array.
   - The prototype for a function that searches an array using a binary search and returns the appropriate index in the array.
   - The prototype for a function that outputs an array.

2. Create your source files as follows:
   - Create a source file that contains the code for the search functions.
   - Create a source file that contains the code for the sort function.
   - Create a source file that contains the code for the output function.

3. Create a file that contains the main function which should perform the following tasks in order.
   - Call the output function.
   - Allow the user to input a key
   - Call the function that performs a sequential search 4 times.
     Output the index # that represents where the item was found.
   - Call the function that performs the insertion sort.
   - Call the output function.
   - Call the function that performs the binary search 4 times.
     Output the index # that represents where the item was found.

Use the following Array:
int intArray[8] = {4, 1, 7, 12, 8, 13, 9, 21};

Turn in as a single PDF file (IN THIS ORDER)

1 – The first page of this lab (fill in the information on the top right)
2 – Program output (cut and pasted into a text file within eclipse)
3 – Header file
4 – Main.cpp
5 – Search functions source file, sort function source file and output source file

Screen Input/Output

```
Index #0:  4
Index #1:  1
Index #2:  7
Index #3:  12
Index #4:  8
Index #5:  13
Index #6:  9
Index #7:  21

Enter an integer to search for:  9
The integer 9 was found in index #6.

Enter an integer to search for:  6
6 was not found!

Enter an integer to search for:  21
The integer 21 was found in index #7.

Enter an integer to search for:  4
The integer 4 was found in index #0.


Performing Insertion Sort!

Index #0:  1
Index #1:  4
Index #2:  7
Index #3:  8
Index #4:  9
Index #5:  12
Index #6:  13
Index #7:  21

Enter an integer to search for:  12
The integer 12 was found in index #5.

Enter an integer to search for:  21
The integer 1 was found in index #7.

Enter an integer to search for:  2
2 was not Found!

Enter an integer to search for:  1
The integer 1 was found in index #0.
```

```
 1 ************************************************************
 2 *  PROGRAMMED BY : Ali Eshghi & Amirarsalan Valipour
 3 *  CLASS         : CS1B
 4 *  SECTION       : MW: 7:30p - 9:50p
 5 *  LAB #5        : Finary Search
 6 ************************************************************
 7
 8 index #0: 4
 9 index #1: 1
10 index #2: 7
11 index #3: 12
12 index #4: 8
13 index #5: 13
14 index #6: 9
15 index #7: 21
16
17 Enter an integer to search for: 9
18 The integer 9 was found in index #6.
19
20 Enter an integer to search for: 6
21 6 was not found!
22
23 Enter an integer to search for: 21
24 The integer 21 was found in index #7.
25
26 Enter an integer to search for: 4
27 The integer 4 was found in index #0.
28
29 Performing Insertion sort
30
31 index #0: 1
32 index #1: 4
33 index #2: 7
34 index #3: 8
35 index #4: 9
36 index #5: 12
37 index #6: 13
38 index #7: 21
39
40 Enter an integer to search for: 12
41 The integer 12 was found in index #5.
42
43 Enter an integer to search for: 21
44 The integer 21 was found in index #7.
45
```

```
Enter an integer to search for: 2
2 was not found!

Enter an integer to search for: 1
The integer 1 was found in index #0.


```

```
1 /
  ********************************************************************************
  *********
2  * PROGRAMMER : Ali Eshghi & Amirarsalan Valipour
3  * STUDENT ID : 1112261 / 1103126
4  * CLASS      : CS1B
5  * SECTION    : MW 7:30pm
6  * Lab #5     : Binary Search
7  * DUE DATE   : 24 September 2019
8
  ********************************************************************************
  *********/
9
10 #ifndef MYHEADER_H_
11 #define MYHEADER_H_
12
13 #include <iostream>
14 #include <iomanip>
15 #include <string>
16 using namespace std;
17
18 //FUNCTIONS
19
20 void PrintHeader(const string PROGRAMMER,
21                  const string CLASS,
22                  const string SECTION,
23                  const int    ASSIGN_NUM,
24                  const string ASSIGN_NAME);
25
26 void ArrayOutput(int numAr[], const int AR_SIZE, int searchNum);
27
28 void ArraySort(int numAr[], const int AR_SIZE);
29
30 int ArraySequentialSearch(int numAr[], const int AR_SIZE, int
   searchNum);
31
32 int ArrayBinarySearch(int numAr[], const int AR_SIZE, int
   searchNum);
33
34 void ArrayOutput(int numAr[], const int AR_SIZE);
35
36 #endif /* MYHEADER_H_ */
37
```

```cpp
/*****************************************************************************
 * PROGRAMMER : Ali Eshghi & Amirarsalan Valipour
 * STUDENT ID : 1112261 / 1103126
 * CLASS      : CS1B
 * SECTION    : MW 7:30pm
 * Lab #5     : Binary Search
 * DUE DATE   : 24 September 2019

*****************************************************************************
 * BINARY SEARCH
 *
-----------------------------------------------------------------------------
 * In this program we will assign an array and will do the following through
 *  different functions: SORTING, BINARY SEARCH, SEQUENTIAL SEARCH, OUTPUT

*****************************************************************************/

#include "MyHeader.h"

int main()
{
    /*************************************************************************
     * CONSTANTS
     *
     -------------------------------------------------------------------------
     * OUTPUT - USED FOR CLASS HEADING
     *
     -------------------------------------------------------------------------
     * PROGRAMMER  : Programmer's Name
     * CLASS   : Student's Course
     * SECTION     : Class Days and Time
     * LAB_NUM     : Lab Number (specific to this lab)
     * LAB_NAME    : Title of the Assignment

```

```
**********************************************************************
******/
30
31      const string PROGRAMMER = "Ali Eshghi & Amirarsalan Valipour";
32      const string CLASS      = "CS1B";
33      const string SECTION= "MW: 7:30p - 9:50p";
34      const int    LAB_NUM=  5;
35      const string LAB_NAME= "Finary Search";
36
37      /
**********************************************************************
******
38       * OUTPUT - HEADER
39
**********************************************************************
******/
40
41      PrintHeader(PROGRAMMER, CLASS, SECTION, LAB_NUM, LAB_NAME);
42
43      /
**********************************************************************
******
44       * CONSTANTS
45       *
----------------------------------------------------------------
----
46       * ESSENTIAL CONSTANTS
47       *
----------------------------------------------------------------
----
48       * AR_SIZE    : Used for the size of array
49
**********************************************************************
******/
50
51      const int AR_SIZE = 8;
52
53      /
**********************************************************************
******
54       * VARIABLES
55
**********************************************************************
******/
56
```

```
57      int numAr[AR_SIZE]= {4,1,7,12,8,13,9,21}; // Calc & Out - given
    array
58
59      int i;                  // Calc - index for the for loop
60      int seqIndex;           // Calc - index for the sequential
    search
61      int binSearch;          // Calc - index for binary search
62      int searchNum;          // In, Calc & Out - User's choice of
    number
63
64      /
    ************************************************************************
    ******
65       * OUTPUT ARRAY
66
    ************************************************************************
    ******/
67
68      ArrayOutput(numAr, AR_SIZE);
69
70      /
    ************************************************************************
    ******
71       * INPUT / PROCESSING
72
    ************************************************************************
    ******/
73
74      //FOR loop runs 4 times and asks user to input the number they
    want to
75      //search for
76
77      for(i = 1; i <= 4; i++)
78      {
79          //asks for users number of choice
80
81          cout << "Enter an integer to search for: ";
82          cin  >> searchNum;
83
84          //Search for the number through this function
85          seqIndex = ArraySequentialSearch(numAr, AR_SIZE,
    searchNum);
86
87          //OUTPUT
88          if(seqIndex != -1)
```

```
 89         {
 90             cout << "The integer " << searchNum
 91                  << " was found in index #"
 92                  << seqIndex  << "."
 93                  << endl << endl;
 94         } //END - IF
 95
 96         else
 97         {
 98             cout << searchNum << " was not found!"
 99                  << endl << endl;
100         } //END - ELSE
101
102     } //END - FOR
103
104     cout << "Performing Insertion sort" << endl << endl;
105
106     //Sorts Array
107     ArraySort(numAr, AR_SIZE);
108
109     //Outputs Array
110     ArrayOutput(numAr, AR_SIZE);
111
112     //FOR loop runs 4 times and asks user to input the number they
    want to
113     //search for
114
115     for(i = 1; i <= 4; i++)
116     {
117         //asks for users number of choice
118         cout << "Enter an integer to search for: ";
119         cin  >> searchNum;
120
121         //Search for the number through this function
122         binSearch = ArrayBinarySearch(numAr, AR_SIZE, searchNum);
123
124         //OUTPUT
125         if(binSearch != -1)
126         {
127             cout << "The integer " << searchNum
128                  << " was found in index #"
129                  << binSearch  << "."
130                  << endl << endl;
131         } //END - IF
132
```

```
133         else
134         {
135             cout << searchNum << " was not found!"
136                  << endl << endl;
137         } //END - ELSE
138 
139     } //END - FOR
140 
141     return 0;
142 }
143 
144 
145 
146 
147 
```

```
1 /
  ********************************************************************************
  *********
2  * PROGRAMMER : Ali Eshghi & Amirarsalan Valipour
3  * STUDENT ID : 1112261 / 1103126
4  * CLASS      : CS1B
5  * SECTION    : MW 7:30pm
6  * Lab #5     : Binary Search
7  * DUE DATE   : 24 September 2019
8
  ********************************************************************************
  *********
9  * Function : PrintHeader
10 *
  ------------------------------------------------------------------------
  -------
11 * This function will print the program header onto the console.
12
  ********************************************************************************
  *********/
13
14 #include "MyHeader.h"
15
16 void PrintHeader ( const string MY_NAME,
17                    const string CLASS,
18                    const string CLASS_TIME,
19                    const int    ASSIGN_NUM,
20                    const string ASSIGN_NAME )
21 {
22
23     cout  << left;
24     cout  <<
  "************************************************************\n"      ;
25     cout  << "*  PROGRAMMED BY : "    << MY_NAME
           ;
26     cout  << "\n*  "    << setw(14) << "CLASS"     << ": " <<
  CLASS      ;
27     cout  << "\n*  "    << setw(14) << "SECTION"   << ": " <<
  CLASS_TIME ;
28     cout  << "\n*  LAB #"   << setw(9)  << ASSIGN_NUM  << ": " <<
  ASSIGN_NAME;
29     cout  <<
  "\n************************************************************\n\n"  ;
30     cout  << right;
31
```

```
32 }
33
34
```

```cpp
/*****************************************************************************
 * PROGRAMMER : Ali Eshghi & Amirarsalan Valipour
 * STUDENT ID : 1112261 / 1103126
 * CLASS      : CS1B
 * SECTION    : MW 7:30pm
 * Lab #5     : Binary Search
 * DUE DATE   : 24 September 2019

*****************************************************************************
 * ArrayBinarySearch
 *
 ----------------------------------------------------------------------------
 * This function will look for the user's given number in our array through
 *  binary search and checks to see if it exits, then it will return the
 *  location of that number as an index. if it does not exists it will return
 *   -1 as an sensitive value.

*****************************************************************************/

#include "MyHeader.h"

int ArrayBinarySearch(int numAr[], const int AR_SIZE, int searchNum)
{
    int index;          //Calc & Output - index to go through the loop
    int low;        //Calc - to store the smaller number
    int high;           //Calc - to store the bigger number
    int     mid;        //Calc - middle of the array address
    bool    searchStat;     //Calc - bool to check if find the value or no

    //INITIALIZATION

    low = 0;
    high = AR_SIZE - 1;
    searchStat = false;

```

```
33      //BINARY SEARCH
34
35      while(!searchStat && low <= high)
36      {
37          mid = (low + high) / 2;
38
39          if(numAr[mid] == searchNum)
40          {
41              searchStat = true;
42              index = mid;
43          }
44
45          else if(numAr[mid] < searchNum)
46          {
47              low = mid + 1;
48          }
49
50          else
51          {
52              high = mid - 1;
53          }
54      } //END - WHILE
55
56      //IF NOT FOUND IN ARRAY
57
58      if(!searchStat)
59      {
60          index = -1;
61      }
62
63      return index;
64  }
65
```

```cpp
/*****************************************************************************
 * PROGRAMMER : Ali Eshghi & Amirarsalan Valipour
 * STUDENT ID : 1112261 / 1103126
 * CLASS      : CS1B
 * SECTION    : MW 7:30pm
 * Lab #5     : Binary Search
 * DUE DATE   : 24 September 2019

*****************************************************************************
 * ArraySequentialSearch
 *
 ----------------------------------------------------------------------------
 * This function will look for the user's given number in our array through
 *  sequential search and checks to see if it exits, then it will return the
 *  location of that number as an index. if it does not exists it will return
 *   -1 as an sensitive value.

*****************************************************************************/

#include "MyHeader.h"

int ArraySequentialSearch(int numAr[], const int AR_SIZE, int searchNum)
{

    int index;          //Calc & Output - index to go through the loop
    bool searchStat;//Calc - bool to check if find the value or no

    //INITIALIZATION

    index = 0;
    searchStat = false;

    //SEQUENTIAL SEARCH

    while(!searchStat && index < AR_SIZE)
```

```
33      {
34
35          if(searchNum == numAr[index])
36          {
37              searchStat = true;
38          }
39
40          else
41          {
42              index++;
43          }
44
45      } //END - WHILE
46
47      //IF NOT FOUND IN ARRAY
48
49      if(!searchStat)
50      {
51          index = -1;
52      }
53
54
55      return index;
56  }
57
58
59
60
```

```cpp
/*****************************************************************************
 * PROGRAMMER : Ali Eshghi & Amirarsalan Valipour
 * STUDENT ID : 1112261 / 1103126
 * CLASS      : CS1B
 * SECTION    : MW 7:30pm
 * Lab #5     : Binary Search
 * DUE DATE   : 24 September 2019

*****************************************************************************
 * ArraySort
 *
 ----------------------------------------------------------------------------
 * This function will sort the array from the smaller number to the bigger
 *  numbers.

*****************************************************************************/

#include "MyHeader.h"

void ArraySort(int numAr[], const int AR_SIZE)
{
    int i;          //Calc - index to go through array
    int j;          //Calc - index to go through array
    int tempNum;//Calc - store value to replace

    //SORTS THE ARRAY
    for(i = 1; i < AR_SIZE; i++)
    {
        tempNum = numAr[i];

        j = i - 1;

        while (j >= 0 && numAr[j] > tempNum)
        {
            numAr[j + 1] = numAr[j];

            j = j - 1;
        } // END - While

```

```
37          numAr[j + 1] = tempNum;
38
39      } // End - For
40
41  }
42
```

```cpp
/*****************************************************************************
 * PROGRAMMER : Ali Eshghi & Amirarsalan Valipour
 * STUDENT ID : 1112261 / 1103126
 * CLASS      : CS1B
 * SECTION    : MW 7:30pm
 * Lab #5     : Binary Search
 * DUE DATE   : 24 September 2019

*****************************************************************************
 * ArrayOutput
 *
 ----------------------------------------------------------------------------
 * This function will output the sorted array.

*****************************************************************************/

#include "MyHeader.h"

void ArrayOutput(int numAr[], const int AR_SIZE)
{
    int index;  //Calc - index to go through array

    for(index = 0; index < AR_SIZE; index++)
    {

        cout << "index #" << index
             << ": " << numAr[index]
             << endl;

    } //END - FOR

    cout << endl;

}
```