School of Applied Sciences And Technology
Department of IST
Program: CNT

CMPE1666 – ICA#01 – Linear Search

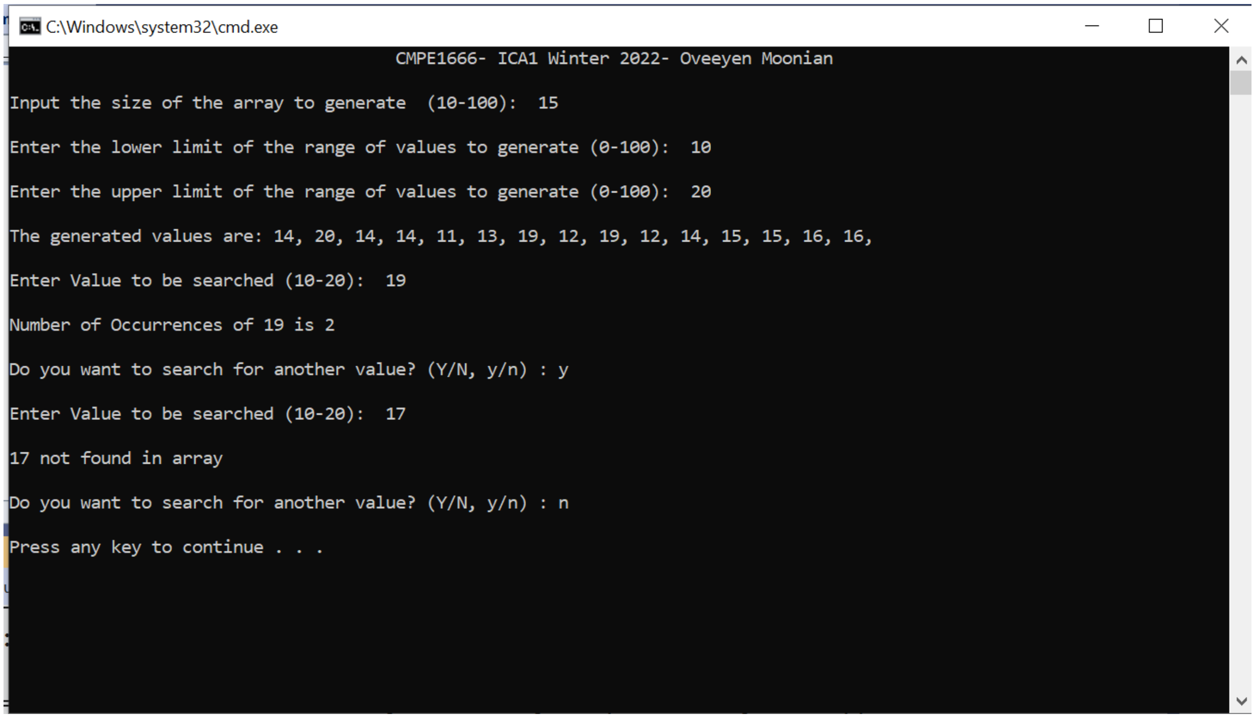
1. Write a Console C# program containing the following methods:
    - A method **GetValue()** that has as parameters a prompt and an **out** variable of type int as well as 2 more int parameters to represent the minimum and maximum acceptable values. It displays the prompt and allows the user to input an integer value, which is assigned to the **out** parameter. The method must ensure that the value given is a valid integer within the given range, otherwise it must be trapped in a loop.
    - A method **GetRange()** that uses **GetValue()** to obtain the lower and upper limits of a range from the user. It must validate that the upper limit is higher than the lower limit.
    - **GenerateArray()**. This method will generate an array of random int values. It has 3 parameters- the first one is the number of values to be generated, the second and third parameters represent the lower limit and upper limit of the range of values to be generated. It returns the generated array.
    - A method **DisplayArray()** that has as parameter an array and it displays the values from the array onto the console. Note that the method requires that you iterate through the elements of the array and display them. Do not use built-in methods like string.join()..etc or Console.Write(array).
    - **CountOccurrences()**. This method must have as parameters an array of int and an int variable. It counts the number of occurrences of the value of the int parameter in the array by using a linear search. It returns this value. Use a **foreach** loop for counting.
      Eg. **CountOccurrences(Arr, val)** will count and return the number of occurrences of **val** in **Arr**. Do not use any array built-in methods or properties and do not use predicates and/or lambda expressions for counting.

Your **Main()** method must declare an array of int. it must then use **GetValue()** to allow the user to input a number of random values to be generated and use **GetRange()** to allow the user to input the lower and upper limits of the range of values. It then uses **GenerateArray()** to create the array with the number of values and range specified by the user.
It displays the array (using **DisplayArray()**), then continuously allows the user to input a value. It validates that the value given is an integer, within the range of values in the array. It must be trapped in a loop as long as a valid in-range value is not given. For a valid value, it then calls **CountOccurrences()** to count the number of occurrences of the given value and displays it on the console. If the value is not found in the array, it must display “Value <value> not found”. The

message for a value in the correct range but not found must be different from that for a value outside the range of contained values. After each run it asks the user whether to have another run. The user has to answer by **y** or **n** (upper or lower case). If the user answers by y (or Y), another run is executed. If the answer is n (or N) the program exits. Any other answer will ask the user the question again.

To test your program, use a small range (say 10-20) and generate a relatively large number of values so as to force more than one occurrence of at least some of the values.

A sample run of the program is as below:



```
                                      CMPE1666- ICA1 Winter 2022- Oveeyen Moonian

Input the size of the array to generate  (10-100):  15

Enter the lower limit of the range of values to generate (0-100):  10

Enter the upper limit of the range of values to generate (0-100):  20

The generated values are: 14, 20, 14, 14, 11, 13, 19, 12, 19, 12, 14, 15, 15, 16, 16,

Enter Value to be searched (10-20):  19

Number of Occurrences of 19 is 2

Do you want to search for another value? (Y/N, y/n) : y

Enter Value to be searched (10-20):  17

17 not found in array

Do you want to search for another value? (Y/N, y/n) : n

Press any key to continue . . .
```

Rubric [30 Marks]

Item	Marks	Penalty
GetValue() performing input, validating and passing to out parameter as required	4	Validation not performed:-2 Not using out parameter: -2
GetRange() working as required	4	Not using **GetValue():-**2 Not using out parameters: -2 Validation for lower limit< upper limit not performed: -2
GenerateArray() generates and return an array of random values within the range specified by the user	4	Not returning array: -2 Not using parameters as required: -3
DisplayArray() working as required	3	Array Parameter not properly used: -2 Not properly iterating through array: -3
CountOccurrence() method working properly as required	4	Parameters not properly used: -4 Used built-in Count() method: -4 Value not returned: -2 Foreach loop not used: -2 Validation not performed as required: -2
Main() working as required	5	Not using the different methods: -4 Does not contain loop to ask user whether to continue: -4 Not validating answer in loop: -2 Message for search values outside range not different from values within range but not found.
Proper documentation (appropriate variable names, spacing between blocks, comments, programmer block)	6	Penalties: -1 to -6 based on instructor's judgement