



**Green University of Bangladesh**  
**Department of Computer Science and Engineering (CSE)**  
**Faculty of Sciences and Engineering**  
**Semester: (Summer, Year:2022), B.Sc. in CSE (Day)**

**Course Title** : Data Structure  
**Course Code** : CSE 106  
**Section** : D2

**Student Details**

	Name	ID
1.	Jahidul Islam	221002504
2.		
3.		

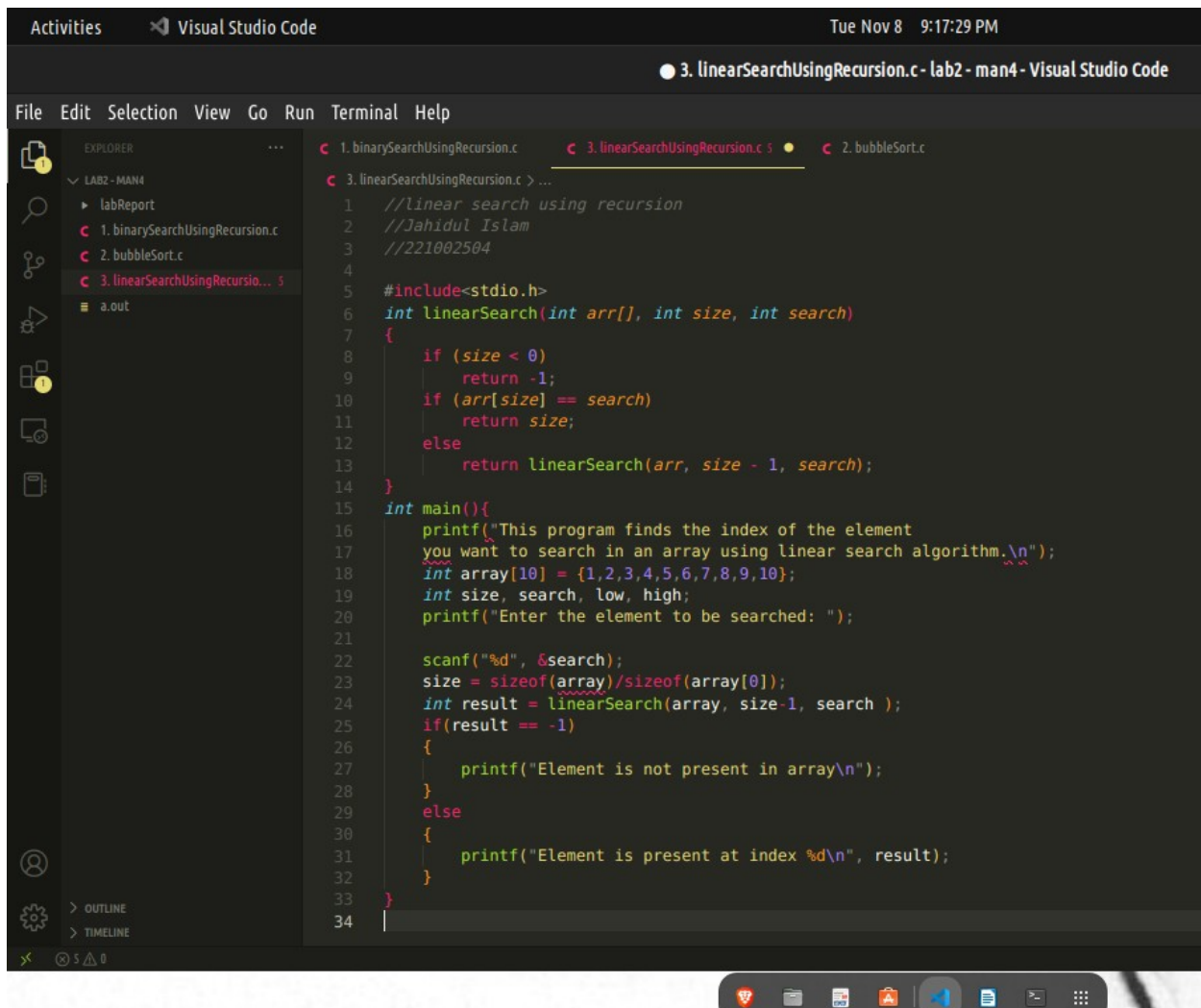
**Submission Date** : 8 November, 2022  
**Course Teacher's Name** : Umme Habiba

[For Teachers use only: **Don't Write Anything inside this box**]

**Marks:** .....  
**Comments:** .....

**Signature:** .....  
**Date:** .....

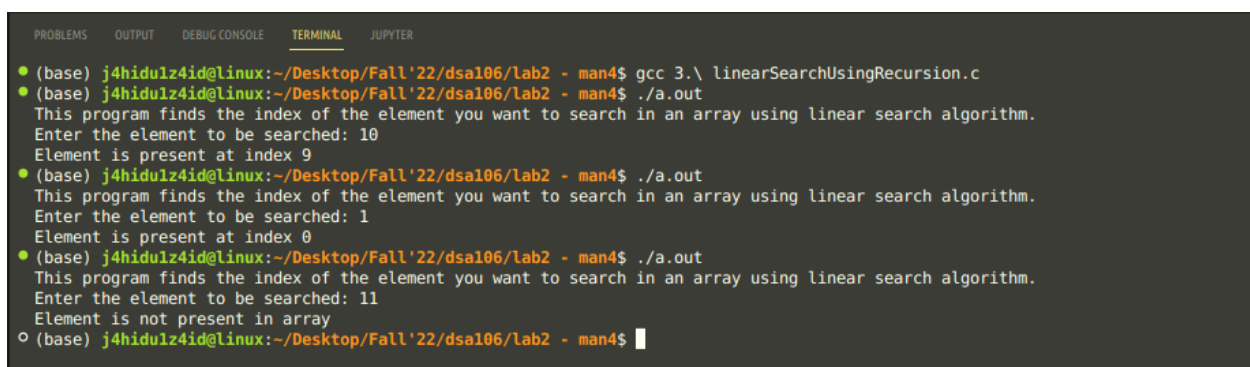
## 1. Implement linear search algorithms using recursion.



The screenshot shows the Visual Studio Code interface with a C program for linear search using recursion. The Explorer panel on the left shows the project structure: LAB2 - MAN4, labReport, 1. binarySearchUsingRecursion.c, 2. bubbleSort.c, 3. linearSearchUsingRecursion.c, and a.out. The main editor displays the code for 3. linearSearchUsingRecursion.c, which includes a recursive function linearSearch and a main function that tests it with an array of 10 elements.

```
1 //linear search using recursion
2 //Jahidul Islam
3 //221002504
4
5 #include<stdio.h>
6 int linearSearch(int arr[], int size, int search)
7 {
8     if (size < 0)
9         return -1;
10    if (arr[size] == search)
11        return size;
12    else
13        return linearSearch(arr, size - 1, search);
14 }
15 int main(){
16     printf("This program finds the index of the element
17     you want to search in an array using linear search algorithm.\n");
18     int array[10] = {1,2,3,4,5,6,7,8,9,10};
19     int size, search, low, high;
20     printf("Enter the element to be searched: ");
21
22     scanf("%d", &search);
23     size = sizeof(array)/sizeof(array[0]);
24     int result = linearSearch(array, size-1, search );
25     if(result == -1)
26     {
27         printf("Element is not present in array\n");
28     }
29     else
30     {
31         printf("Element is present at index %d\n", result);
32     }
33 }
34
```

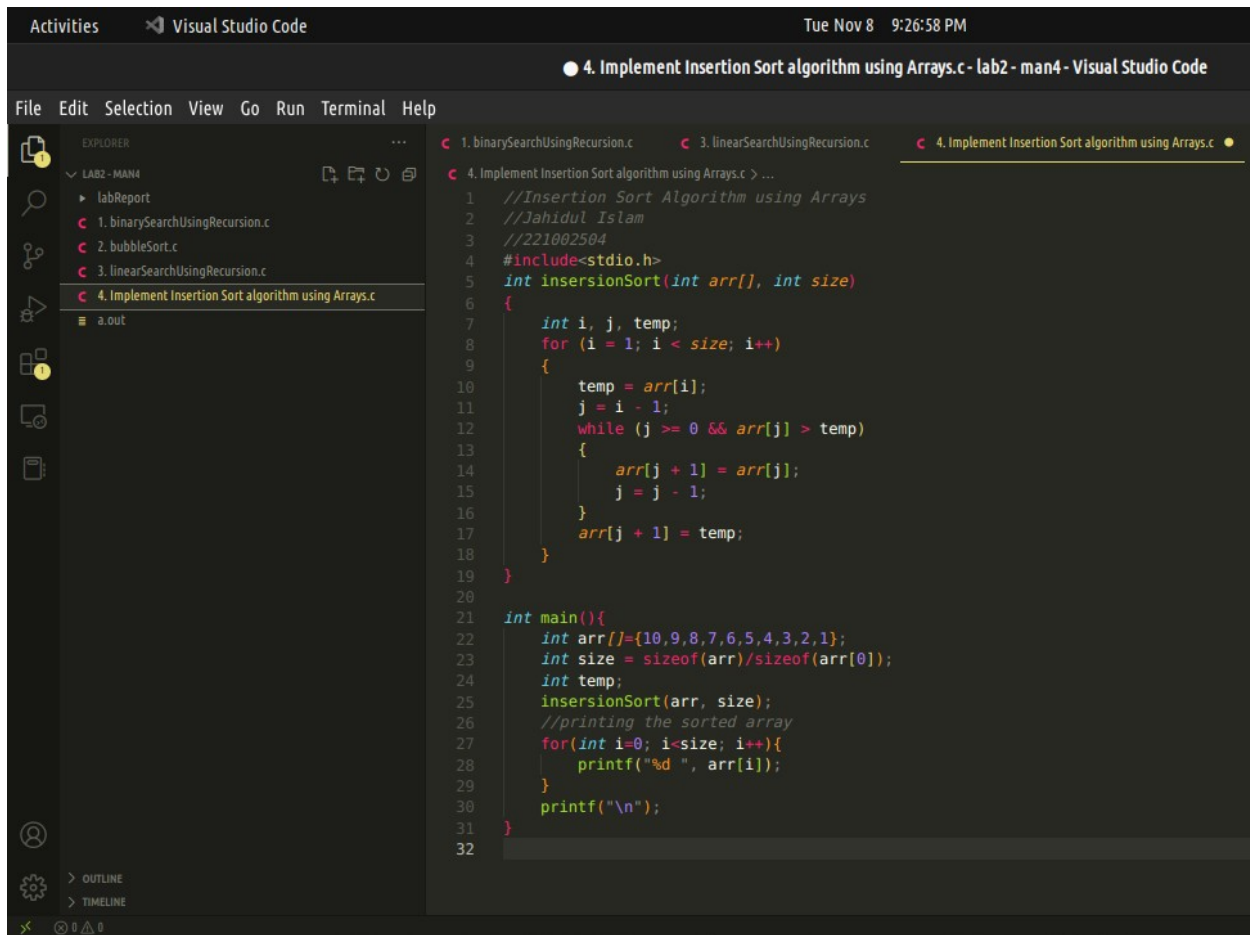
### Output of solution 01:



The screenshot shows the terminal output of the program. It displays three test cases where the user enters an element to be searched, and the program outputs the index of the element or a message indicating it is not present in the array.

```
(base) jahidulz4id@linux:~/Desktop/Fall'22/dsa106/lab2 - man4$ gcc 3.\ linearSearchUsingRecursion.c
(base) jahidulz4id@linux:~/Desktop/Fall'22/dsa106/lab2 - man4$ ./a.out
This program finds the index of the element you want to search in an array using linear search algorithm.
Enter the element to be searched: 10
Element is present at index 9
(base) jahidulz4id@linux:~/Desktop/Fall'22/dsa106/lab2 - man4$ ./a.out
This program finds the index of the element you want to search in an array using linear search algorithm.
Enter the element to be searched: 1
Element is present at index 0
(base) jahidulz4id@linux:~/Desktop/Fall'22/dsa106/lab2 - man4$ ./a.out
This program finds the index of the element you want to search in an array using linear search algorithm.
Enter the element to be searched: 11
Element is not present in array
(base) jahidulz4id@linux:~/Desktop/Fall'22/dsa106/lab2 - man4$
```

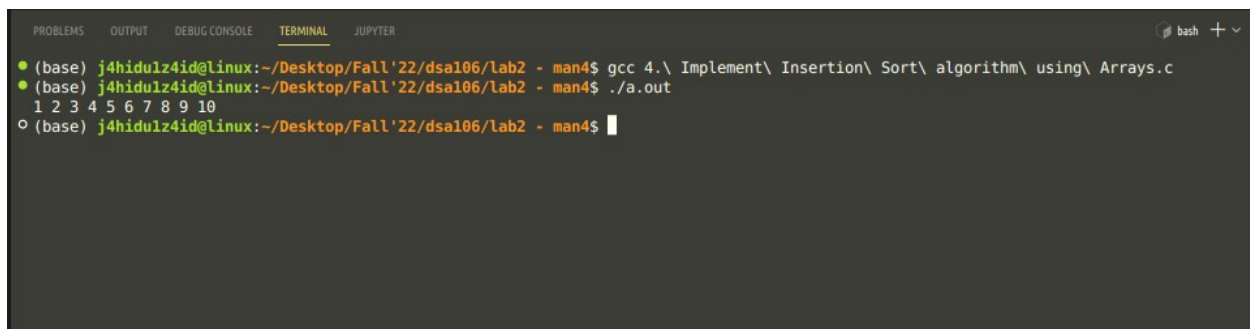
## 2. Implement Insertion Sort algorithm using Arrays



The screenshot shows the Visual Studio Code interface with the file explorer on the left and the code editor in the center. The file explorer shows a project named 'LAB2 - MAN4' with a 'labReport' folder and several C files. The code editor displays the implementation of the Insertion Sort algorithm using Arrays. The code is as follows:

```
1 //Insertion Sort Algorithm using Arrays
2 //Jahidul Islam
3 //221002504
4 #include<stdio.h>
5 int insersionSort(int arr[], int size)
6 {
7     int i, j, temp;
8     for (i = 1; i < size; i++)
9     {
10         temp = arr[i];
11         j = i - 1;
12         while (j >= 0 && arr[j] > temp)
13         {
14             arr[j + 1] = arr[j];
15             j = j - 1;
16         }
17         arr[j + 1] = temp;
18     }
19 }
20
21 int main(){
22     int arr[]={10,9,8,7,6,5,4,3,2,1};
23     int size = sizeof(arr)/sizeof(arr[0]);
24     int temp;
25     insersionSort(arr, size);
26     //printing the sorted array
27     for(int i=0; i<size; i++){
28         printf("%d ", arr[i]);
29     }
30     printf("\n");
31 }
32
```

## Output of solution 02:



The screenshot shows the terminal output of the program. The commands and their outputs are as follows:

```
(base) j4hidulz4id@linux:~/Desktop/Fall'22/dsa106/lab2 - man4$ gcc 4.\ Implement\ Insertion\ Sort\ algorithm\ using\ Arrays.c
(base) j4hidulz4id@linux:~/Desktop/Fall'22/dsa106/lab2 - man4$ ./a.out
1 2 3 4 5 6 7 8 9 10
(base) j4hidulz4id@linux:~/Desktop/Fall'22/dsa106/lab2 - man4$
```