CHITTAGONG UNIVERSITY OF ENGINEERING AND TECHNOLOGY

Department of Computer Science & Engineering

Course No: CSE- 244

Course Title:
Algorithm Desing and Analysis (Sessional)

Experiment No: 01

Name Of the Experiment: Introduction to algorithm

Identity Details

Name of the Student:Md.Safayat Bin Nasir

Student Id Number: 2004099

Level: 2 Term: 2 Section: B

Group: B1

Date of performance: 20-08-2023

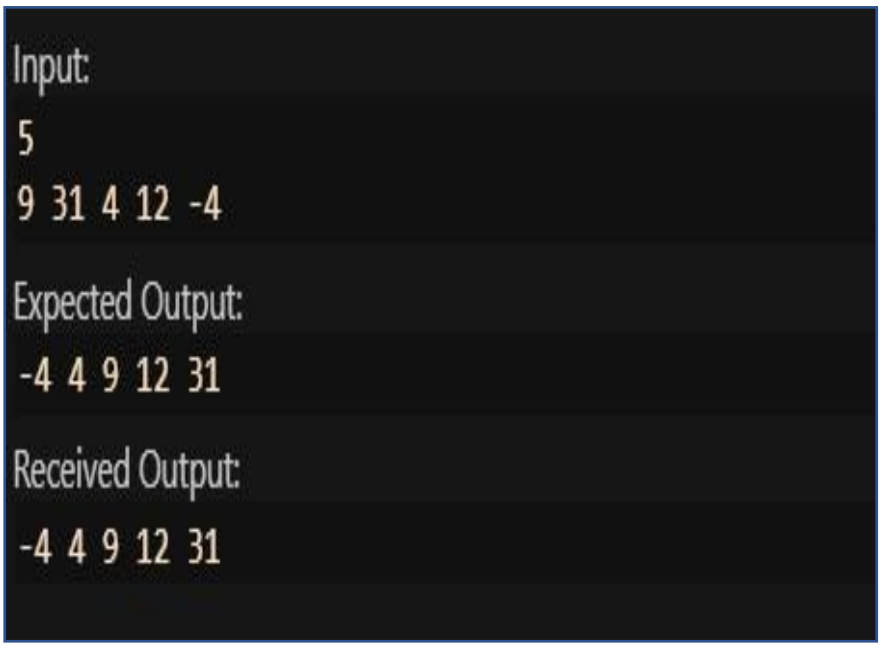
Date of Submission: 03-09-2023

Remarks

Program 1: Given a list of numbers , sort them in ascending order using bubble sort algorithm. After implementing the program analyze its complexity.

```cpp
#include <bits/stdc++.h>
using namespace std;
vector<int> bubble(vector<int>&v){
    for(int i = 0 ; i < v.size() ; i++){
        for(int j = 0 ; j < v.size(); j++){
            if(v[i] <= v[j]){
                int temp = v[j];
                v[j] = v[i];
                v[i] = temp;
            }
        }
    }
    return v;
}
int main()
{
    int n;
    cin >> n;
    vector<int>v(n);
    for(auto &x:v) cin >> x;
    v = bubble(v);
    for(int i = 0 ; i < n ;i++){
        cout << v[i] << " ";
    }
    cout << endl;
}
```

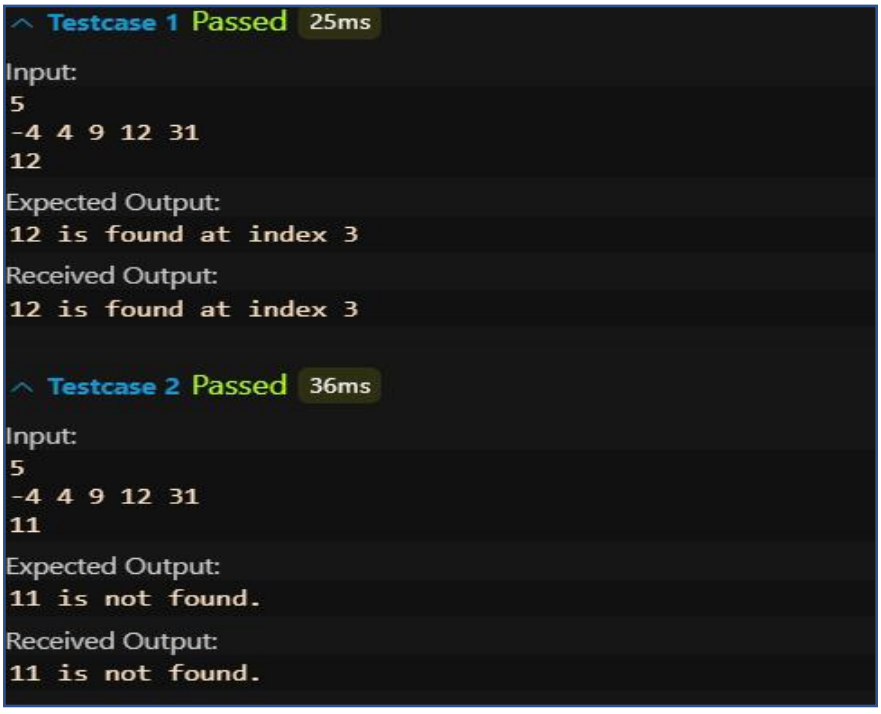
Output:



Program 2: Given a list of sorted numbers in ascending order, find a specific element from that list using binary search.After implementing the program analyze its complexity.

```cpp
#include<bits/stdc++.h>
using namespace std;
int main(){
    int n ;
    cin >> n;
    vector<int>v(n);
    for(auto &x:v) cin >> x;
    int x;
    cin >> x;
    int l = 0, r = n , ind =- 1;
    while(l <= r){
        int mid = l + (r-l) /2;
        if(v[mid] == x){
            ind = mid;
            break;
        }
        if(v[mid] > x) r = mid-1;
        else l = mid+1;
    }
    if(ind == -1){
        cout << x << " is not found." << endl;
    }
    else{
        cout << x << " is found at index " << ind << endl;
    }
}
```

Output:



Program 3: Given a list of sorted numbers in ascending order, find how many times a specific element is presented in that list.Complexity of the code must be O(logn).

```cpp
#include<bits/stdc++.h>
using namespace std;
int main(){
    int n ;
    cin >> n;
    vector<int>v(n);
    for(auto &x:v) cin >> x;
    int x;
    cin >> x;
    int l = 0, r = n , ind =- 1;
    while(l <= r){
        int mid = l + (r-l) /2;
        if(v[mid] == x){
            ind = mid;
            break;
        }
    }
    else{
        int l = ind-1, r = ind+1, cnt = 1;
        while(l >= 0 ){
            if(v[l] == x){
                l--;
                cnt++;
            }
            else break;
        }
        while(r < n ){
            if(v[r] == x){
                r++;
                cnt++;
            }
            else break;
        }
        cout << x << " is presented " << cnt << " times" << endl;
    }
}
```

Program 3: Given a list of sorted numbers in ascending order, find how many times a specific element is presented in that list.Complexity of the code must be O(logn).

Output: