VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

FALL Semester 2022-23

ITA5002 - Problem Solving with Data Structuresand Algorithms Lab

LAB DIGITAL ASSESSMENT – II

Due Date: 05-12-2022

Submitted By: Rajat Singh
Reg No.: 22MCA0139

SOLVE THE PROBLEMS GIVEN BELOW

1. Try to create an array of N elements as input which consists of Positive, Negative numbers and also the duplicate values. Perform any sorting procedure to get output as two sorted array one with positive numbers and one with negative numbers and print the number of comparisons made for each sorted array and aslo the say count for the each the duplicate values.
   Exapmle : input[15]= { 10,4,-3,-1,0,4,3,-15,-8,4,-1,9,3,1,7}

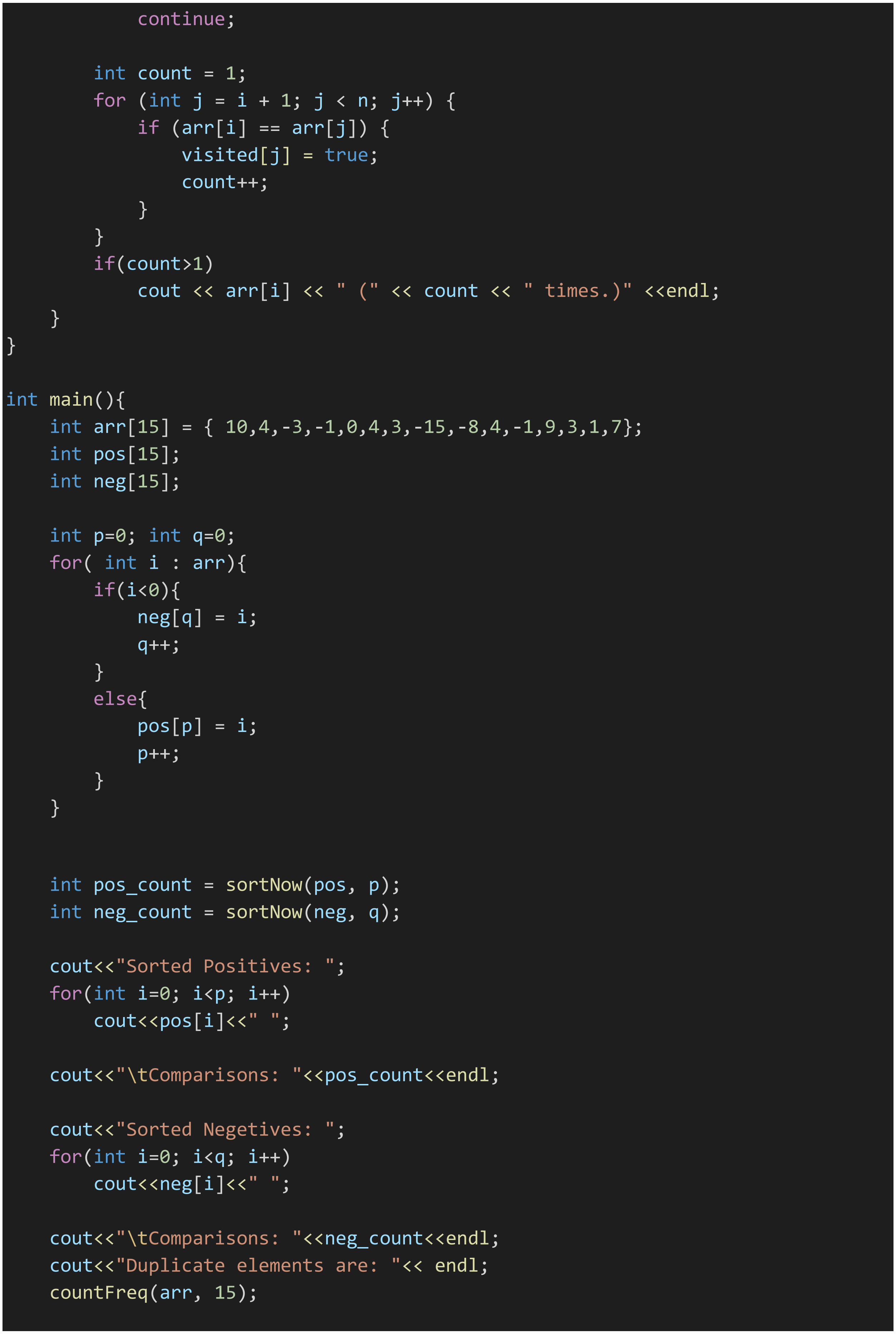
   Output1[15]={-1,-1,-3,-8,-15} and print Number of Comparisons

   Output2[15]={0,1,3,3,4,4,4,7,9,10} and print Number of Comparisons

   -1 has been present twice
   + 3 has been present twice
   + 4 has been present thrice

Code:

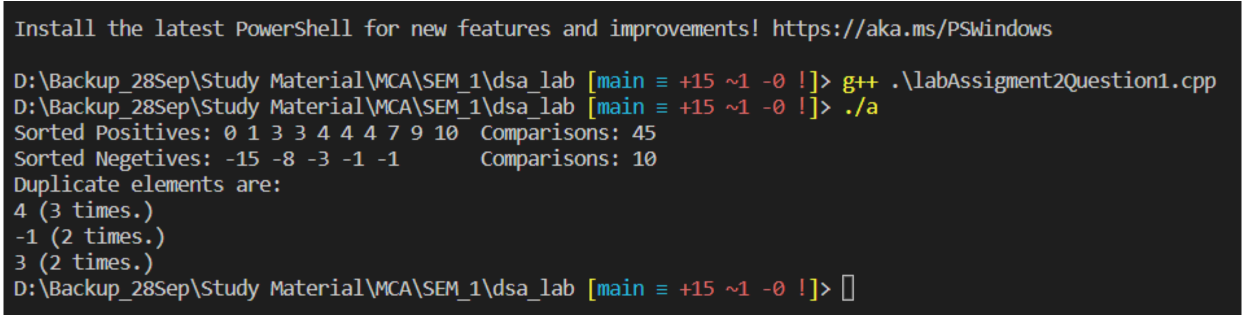
```
#include<iostream>
#include<vector>
using namespace std;

int sortNow(int arr[], int n){
    int i, j;
    int c=0;
    for (i = 0; i < n - 1; i++){
        for (j = 0; j < n - i - 1; j++){
            c++;
            if (arr[j] > arr[j + 1]){
                swap(arr[j], arr[j + 1]);
            }
        }
    }
    return c;
}

void countFreq(int arr[], int n)
{
    vector<bool> visited(n, false);


    for (int i = 0; i < n; i++) {

        if (visited[i] == true)
```



```
            continue;

        int count = 1;
        for (int j = i + 1; j < n; j++) {
            if (arr[i] == arr[j]) {
                visited[j] = true;
                count++;
            }
        }
        if(count>1)
            cout << arr[i] << " (" << count << " times.)" <<endl;
    }
}

int main(){
    int arr[15] = { 10,4,-3,-1,0,4,3,-15,-8,4,-1,9,3,1,7};
    int pos[15];
    int neg[15];

    int p=0; int q=0;
    for( int i : arr){
        if(i<0){
            neg[q] = i;
            q++;
        }
        else{
            pos[p] = i;
            p++;
        }
    }


    int pos_count = sortNow(pos, p);
    int neg_count = sortNow(neg, q);

    cout<<"Sorted Positives: ";
    for(int i=0; i<p; i++)
        cout<<pos[i]<<" ";

    cout<<"\tComparisons: "<<pos_count<<endl;

    cout<<"Sorted Negetives: ";
    for(int i=0; i<q; i++)
        cout<<neg[i]<<" ";

    cout<<"\tComparisons: "<<neg_count<<endl;
    cout<<"Duplicate elements are: "<< endl;
    countFreq(arr, 15);
```

```
    return 0;
}
```

Output:



```
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

D:\Backup_28Sep\Study Material\MCA\SEM_1\dsa_lab [main ≡ +15 ~1 -0 !]> g++ .\labAssigment2Question1.cpp
D:\Backup_28Sep\Study Material\MCA\SEM_1\dsa_lab [main ≡ +15 ~1 -0 !]> ./a
Sorted Positives: 0 1 3 3 4 4 4 7 9 10  Comparisons: 45
Sorted Negetives: -15 -8 -3 -1 -1       Comparisons: 10
Duplicate elements are:
4 (3 times.)
-1 (2 times.)
3 (2 times.)
D:\Backup_28Sep\Study Material\MCA\SEM_1\dsa_lab [main ≡ +15 ~1 -0 !]> ▯
```

2. Perform Binary Search for the given input

   A[14]={11,22,33,44,55,66,77,88,99,111,222,333,444}

   And try to print the position of the number present in the upper half starting from 1 to n N is the middle element.

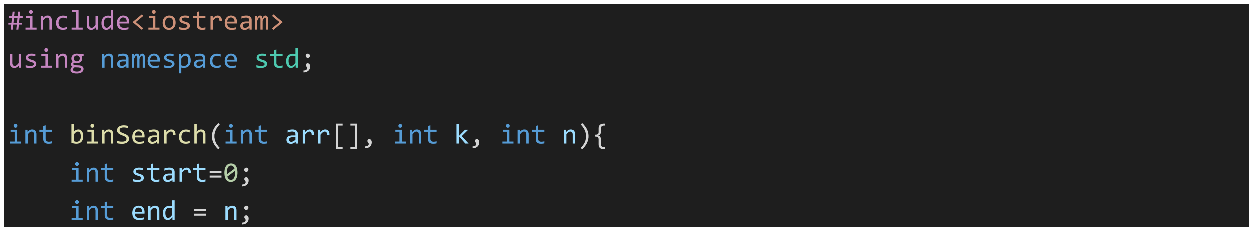
   After middle element again the position should start from 1 to n. Here N is the last element.

   Example mid = (0+14)/2 ==7 middle element is 88

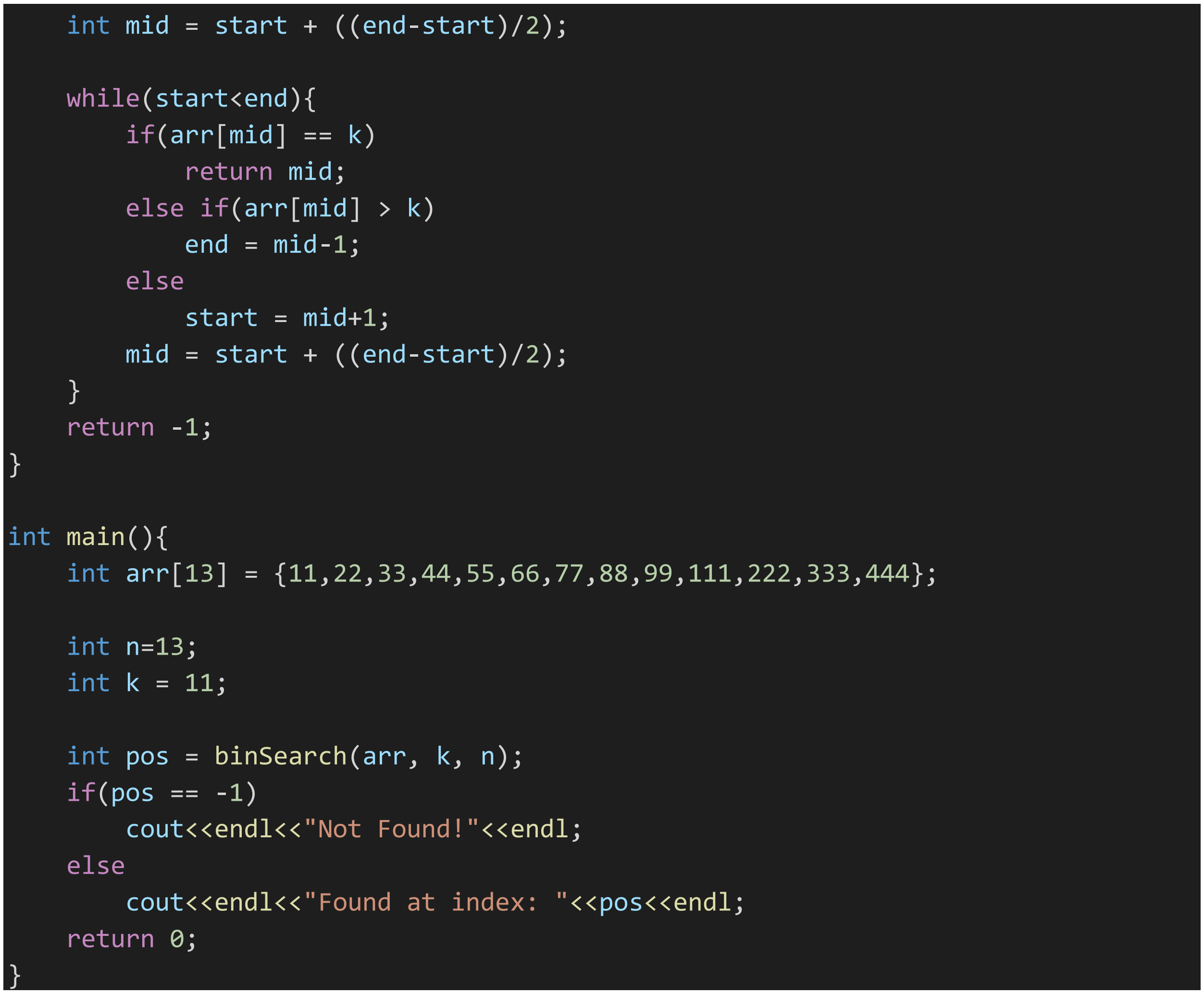
   Numbers 33 present in the third position
   88 is present in the ninth positon

   99 is present in first position
   333 is present in fourth position

Code:

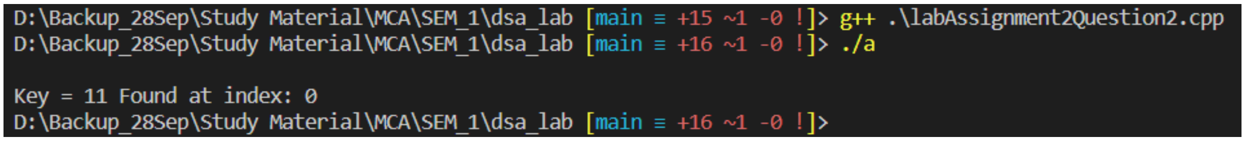


```
#include<iostream>
using namespace std;

int binSearch(int arr[], int k, int n){
    int start=0;
    int end = n;
```



```
    int mid = start + ((end-start)/2);

    while(start<end){
        if(arr[mid] == k)
            return mid;
        else if(arr[mid] > k)
            end = mid-1;
        else
            start = mid+1;
        mid = start + ((end-start)/2);
    }
    return -1;
}

int main(){
    int arr[13] = {11,22,33,44,55,66,77,88,99,111,222,333,444};

    int n=13;
    int k = 11;

    int pos = binSearch(arr, k, n);
    if(pos == -1)
        cout<<endl<<"Not Found!"<<endl;
    else
        cout<<endl<<"Found at index: "<<pos<<endl;
    return 0;
}
```

Output:



```
D:\Backup_28Sep\Study Material\MCA\SEM_1\dsa_lab [main ≡ +15 ~1 -0 !]> g++ .\labAssignment2Question2.cpp
D:\Backup_28Sep\Study Material\MCA\SEM_1\dsa_lab [main ≡ +16 ~1 -0 !]> ./a

Key = 11 Found at index: 0
D:\Backup_28Sep\Study Material\MCA\SEM_1\dsa_lab [main ≡ +16 ~1 -0 !]>
```