Day-5

NAME-Anish

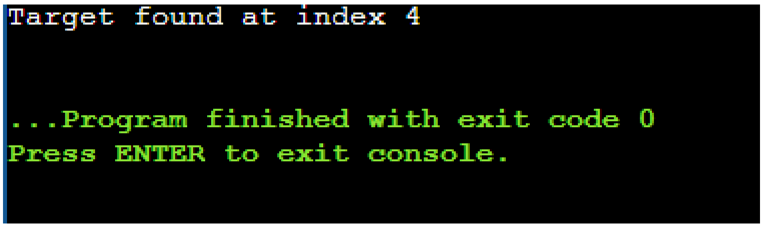
UID-22BCS15292

DATE-26/12/2024

Question1

```
#include <iostream>
#include <vector>
usingnamespacestd;

intlinearSearch(constvector<int>&arr,inttarget){ for (int i = 0;

    i < arr.size(); i++) {
        if(arr[i]==target){
            returni;
        }
    }
    return-1;
}

intmain(){
    vector<int>array={10,20,30,40,50}; int
    targetValue = 50;

    intresult=linearSearch(array,targetValue); if (result

    != -1) {
        cout<<"Targetfoundatindex"<<result<<endl;
    }else{
        cout<<"Targetnotfoundinthearray"<<endl;
    }

    return0;
}
```

Output:

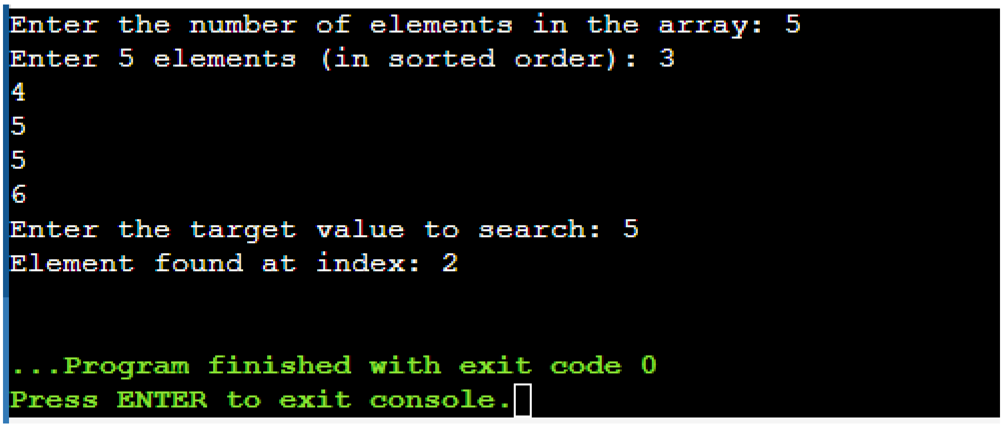


```
Target found at index 4


...Program finished with exit code 0
Press ENTER to exit console.
```

Question2

```
#include <iostream>
#include <algorithm>
usingnamespacestd;

intbinarySearch(intarr[],intsize,inttarget){ int left = 0;
    intright=size-1;

    while(left<=right){
        intmid=left+(right-left)/2; if
        (arr[mid] == target) {
            returnmid;
        }
        if(arr[mid]<target){ left =
            mid + 1;
        }
        else{
            right=mid-1;
        }
    }

    return-1;
}
intmain(){
    intsize, target;
    cout<<"Enterthenumberofelementsinthearray:"; cin >> size;
    intarr[size];
    cout<<"Enter"<<size<<"elements(insortedorder):"; for (int i = 0; i <
    size; i++) {
        cin>>arr[i];
    }
    cout<<"Enterthetargetvaluetosearch:"; cin >>
    target;
    sort(arr,arr+size);
    intresult=binarySearch(arr,size,target); if (result
    != -1) {
        cout<<"Elementfoundatindex:"<<result<<endl;
    }else{
        cout<<"Elementnotfound"<<endl;
    }

    return0;
}
```

Output:

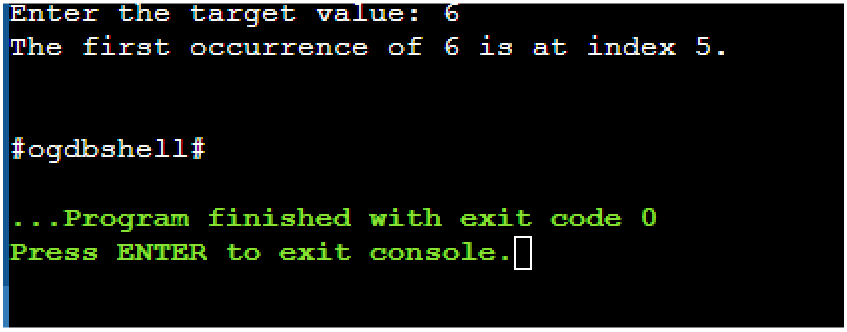


```
Enter the number of elements in the array: 5
Enter 5 elements (in sorted order): 3
4
5
5
6
Enter the target value to search: 5
Element found at index: 2


...Program finished with exit code 0
Press ENTER to exit console.
```

Question3

```
#include <iostream>
#include <vector>
usingnamespacestd;


intfindFirstOccurrence(constvector<int>&arr,inttarget){ int left = 0,
    right = arr.size() - 1;
    intresult=-1;

    while(left<=right){
        intmid=left+(right- left)/2;

        if(arr[mid]==target){ result
            = mid;
            right=mid-1;
        }elseif(arr[mid]<target){ left =
            mid + 1;
        }else{
            right=mid-1;
        }
    }

    returnresult;
}

intmain(){
    vector<int>arr={1,2,4,4,4,6,7}; int target;

    cout<<"Enterthetargetvalue:"; cin >>
    target;

    intindex=findFirstOccurrence(arr,target); if (index


    != -1) {
        cout<<"Thefirstoccurrenceof"<<target<<"isatindex
"<<index<<"."<< endl;
    }else{
        cout<<target<<"isnotpresentinthearray."<<endl;
    }

    return0;
}
```

Output:

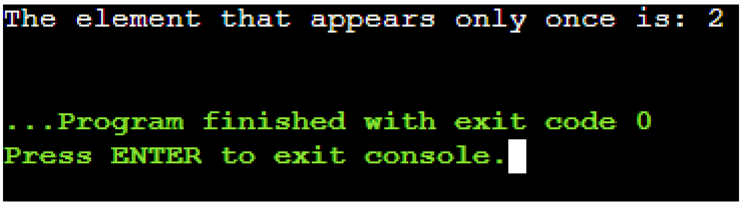


```
Enter the target value: 6
The first occurrence of 6 is at index 5.


#ogdbshell#

...Program finished with exit code 0
Press ENTER to exit console.
```

Question4

```
#include <iostream>
#include <vector>
usingnamespacestd;


intfindSingleElement(constvector<int>&arr){ int left =
    0, right = arr.size() - 1;

    while(left<right){
        intmid=left+(right- left)/2;


        if(mid%2==1){ mid--
            ;
        }



        if(arr[mid]==arr[mid+1]){ left = mid


            + 2;
        }else{

            right= mid;
        }
    }



    returnarr[left];
}

intmain(){
    vector<int> arr={1,1,2,3,3,4,4,8,8}; int result =

    findSingleElement(arr);
    cout<<"Theelementthatappears onlyonceis:"<<result<<endl;

    return0;
}
```

Output:



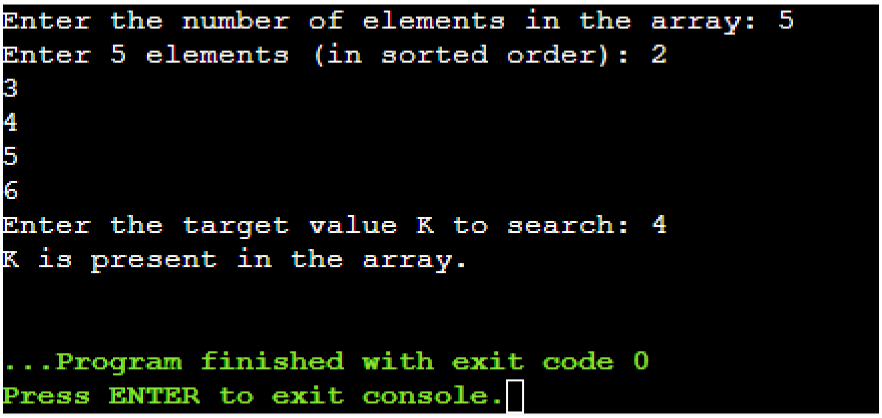
```
The element that appears only once is: 2


...Program finished with exit code 0
Press ENTER to exit console.
```

Question5

```
#include <iostream>
usingnamespacestd;


boolbinarySearch(intarr[],intsize,intK){ int left =
    0;
    intright=size-1;


    while(left<=right){
        intmid=left+(right- left)/2;

        if(arr[mid]==K){ return
            true;
        }
        elseif(arr[mid]<K){ left =
            mid + 1;
        }
        else{
            right=mid-1;
        }
    }

}

int main() {
    intsize,K;


    cout<<"Enterthenumberofelementsinthearray:"; cin >> size;


    intarr[size];


    cout<<"Enter"<<size<<"elements(insortedorder):"; for (int i = 0; i <
    size; i++) {
        cin>>arr[i];
    }


    cout<<"EnterthetargetvalueKto search:"; cin >> K;


    if(binarySearch(arr,size,K)) {
        cout<<"Kispresentinthearray."<<endl;
    }else{
        cout<<"Kisnotpresentinthearray."<<endl;
    }

    return0;
```

```
}
```

Output:

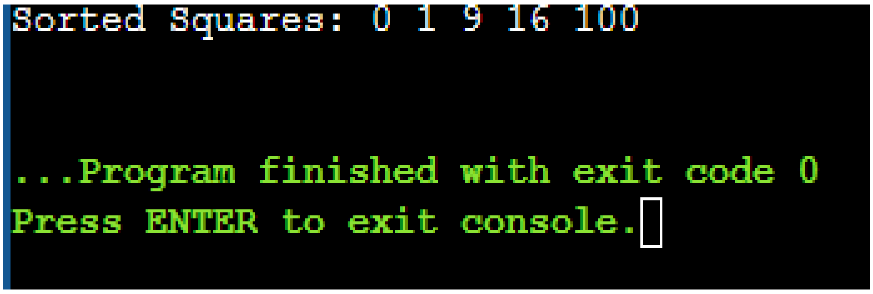


```
Enter the number of elements in the array: 5
Enter 5 elements (in sorted order): 2
3
4
5
6
Enter the target value K to search: 4
K is present in the array.


...Program finished with exit code 0
Press ENTER to exit console.
```

Question6

```
#include <iostream>
#include <vector>
#include<algorithm>

std::vector<int>sortedSquares(conststd::vector<int>&nums){ int n =
    nums.size();
    std::vector<int>result(n); int
    left = 0, right = n - 1; int pos =
    n - 1;

    while(left<=right){
        intleftSquare=nums[left]*nums[left];
        intrightSquare=nums[right]*nums[right]; if
        (leftSquare > rightSquare) {
            result[pos]=leftSqua

            result[pos]=rightSquare;
            right--;
        }
        pos--;
    }

    returnresult;
}

intmain(){
    std::vector<int> nums = {-4, -1, 0, 3, 10};
    std::vector<int>result=sortedSquares(nums);

    std::cout<<"SortedSquares:"; for (int
    num : result) {
        std::cout<<num<<"";
    }
    std::cout<<std::endl;

    return0;
}
```

Output:

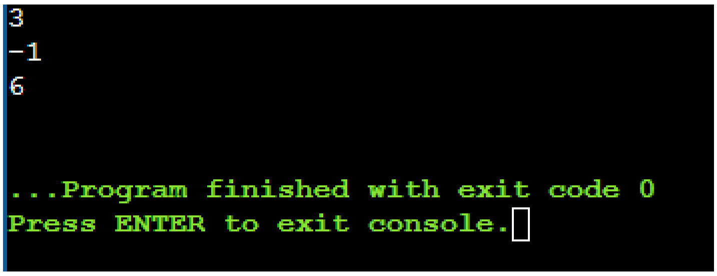


```
Sorted Squares: 0 1 9 16 100


...Program finished with exit code 0
Press ENTER to exit console.
```

Question7

```
#include <iostream>
#include <vector>
usingnamespacestd;

intfindFirstOccurrence(intk,constvector<int>&arr){ for (int i =

    0; i < arr.size(); i++) {
        if(arr[i]==k){
            returni+ 1;
        }
    }
    return-1;
}

int main() {

    intk1=16;
    vector<int>arr1={9,7,16,16,4};
    cout<<findFirstOccurrence(k1,arr1)<<endl;


    intk2=98;
    vector<int> arr2 = {1, 22, 57, 47, 34, 18, 66};
    cout<<findFirstOccurrence(k2,arr2)<<endl;


    intk3=9;
    vector<int> arr3 = {1, 22, 57, 47, 34, 9, 66};
    cout<<findFirstOccurrence(k3,arr3)<<endl;

    return0;
}
```

Output:

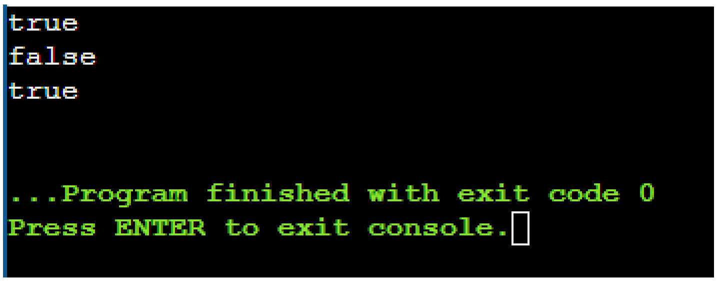


```
3
-1
6


...Program finished with exit code 0
Press ENTER to exit console.
```

Question8

```
#include <iostream>
#include <vector>
usingnamespacestd;

boolisPresent(constvector<int>&arr,intk){ int left = 0,
    right = arr.size() - 1;


    while(left<=right){
        intmid=left+(right-left)/2; if
        (arr[mid] == k) {
            returntrue;
        }elseif(arr[mid]<k){ left =
            mid + 1;
        }else{
            right=mid-1;
        }
    }

    returnfalse;
}

intmain(){
    vector<int>arr1={1,2,3,4,6}; int k1 =
    6;
    cout<<(isPresent(arr1,k1)?"true" :"false")<<endl;

    vector<int>arr2={1,2,4,5,6}; int k2 =
    3;
    cout<<(isPresent(arr2,k2)?"true" :"false")<<endl;

    vector<int>arr3={1,2,4,5,6}; int k3 =
    6;
    cout<<(isPresent(arr3,k3)?"true" :"false")<<endl;

    return0;
}
```

Output:

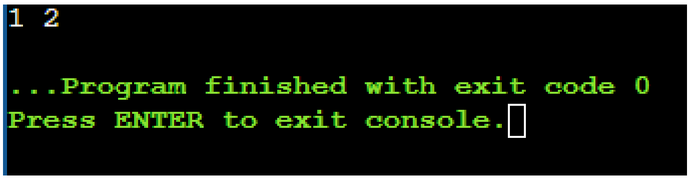


```
true
false
true


...Program finished with exit code 0
Press ENTER to exit console.
```

Question9

```
#include <vector>
#include<iostream>

std::vector<int>targetIndices(conststd::vector<int>&nums,int target) {
    intlessCount=0,targetCount=0;


        for(intnum:nums){ if
            (num < target) {
               ++lessCount;
        }elseif(num==target){
            ++targetCount;
        }
    }


    std::vector<int>result;
    for(inti=0;i<targetCount;++i){
        result.push_back(lessCount+i);
    }

    returnresult;
}

intmain(){
    std::vector<int>nums={1,2,5,2,3}; int target
    = 2;

    std::vector<int>result=targetIndices(nums,target);


    for (int index : result) {
        std::cout<<index<<"";
    }

    return0;
}
```

Output:

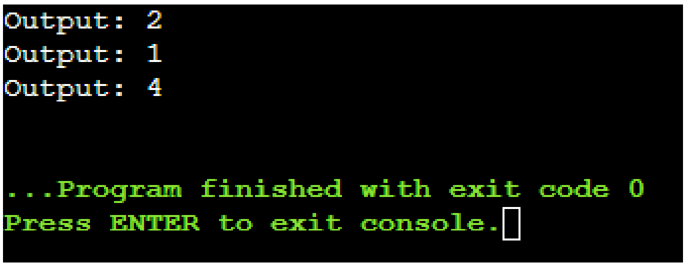


```
1 2

...Program finished with exit code 0
Press ENTER to exit console.
```

Question 10

```
#include <vector>
#include<iostream>

intsearchInsert(std::vector<int>&nums,inttarget){ int left = 0,
    right = nums.size() - 1;


    while(left<=right){
        intmid=left+(right- left)/2;

        if(nums[mid]==target){ return mid;
        }elseif(nums[mid]<target){ left =
            mid + 1;
        }else{
            right=mid-1;
        }
    }


    returnleft;
}

intmain(){
    std::vector<int>nums1={1,3,5,6}; int target1
    = 5;
    std::cout<<"Output:"<<searchInsert(nums1,target1)<<"\n";

    std::vector<int>nums2={1,3,5,6}; int target2
    = 2;
    std::cout<<"Output:"<<searchInsert(nums2,target2)<<"\n";

    std::vector<int>nums3={1,3,5,6}; int target3
    = 7;
    std::cout<<"Output:"<<searchInsert(nums3,target3)<<"\n";


    return0;
}
```

Output:



```
Output: 2
Output: 1
Output: 4


...Program finished with exit code 0
Press ENTER to exit console.
```