**First and last occurrences of x: -**

Medium Accuracy: 37.36% Submissions: 212K+ Points: 4

Given a sorted array **arr** containing **n** elements with possibly duplicate is to find indexes of first elements, the task is to find the first and last occurrences of an element **x** in the given array.  
**Note:** If the number **x** is not found in the array then return both the indices as -1.

**Example 1:**

**Input:**

n=9, x=5

arr[] = { 1, 3, 5, 5, 5, 5, 67, 123, 125 }

**Output:**   
2 5

**Explanation**:   
First occurrence of 5 is at index 2 and last occurrence of 5 is at index 5.

**Example 2:**

**Input:**

n=9, x=7

arr[] = { 1, 3, 5, 5, 5, 5, 7, 123, 125 }

**Output:**   
6 6  
**Explanation:**   
First and last occurrence of 7 is at index 6.

**Your Task:**  
Since, this is a function problem. You don't need to take any input, as it is already accomplished by the driver code. You just need to complete the function **find**() that takes **array arr, integer n and integer x** as parameters and returns the required answer.

**Expected Time Complexity:** O(logN)  
**Expected Auxiliary Space:** O(1).

**Constraints:**1 ≤ N ≤ 106  
1 ≤ arr[i],x ≤ 109

**Code: -**

//{ Driver Code Starts

#include<bits/stdc++.h>

using namespace std;

// } Driver Code Ends

class Solution

{

public:

int lower(int arr[], int n, int target){

int low = 0, mid, high = n-1, ans = -1;

while(low <= high){

mid = low + (high - low) / 2;

if(arr[mid] >= target){

if(arr[mid] == target)

ans = mid;

high = mid - 1;

}

else

low = mid + 1;

}

return ans;

}

int upper(int arr[], int n, int target){

int low = 0, mid, high = n-1, ans = -1;

while(low <= high){

mid = low + (high - low) / 2;

if(arr[mid] <= target){

if(arr[mid] == target)

ans = mid;

low = mid + 1;

}

else

high = mid - 1;

}

return ans;

}

vector<int> find(int arr[], int n , int x ){

int first = lower(arr, n, x);

if(first == -1)

return {-1, -1};

int last = upper(arr, n, x);

return {first, last};

}

};

//{ Driver Code Starts.

int main()

{

int t;

cin>>t;

while(t--)

{

int n,x;

cin>>n>>x;

int arr[n],i;

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

cin>>arr[i];

vector<int> ans;

Solution ob;

ans=ob.find(arr,n,x);

cout<<ans[0]<<" "<<ans[1]<<endl;

}

return 0;

}

// } Driver Code Ends

**T.C: - O(log N)**

**S.C: - O(1)**