**Number of occurrence :-**

Medium Accuracy: 59.34% Submissions: 135K+ Points: 4

Given a sorted array **Arr**of size **N**and a number **X**, you need to find the number of occurrences of**X** in **Arr**.

**Example 1:**

**Input:**

N = 7, X = 2

Arr[] = {1, 1, 2, 2, 2, 2, 3}

**Output:** 4

**Explanation:** 2 occurs 4 times in the

given array.

**Example 2:**

**Input:**

N = 7, X = 4

Arr[] = {1, 1, 2, 2, 2, 2, 3}

**Output:** 0

**Explanation:** 4 is not present in the

given array.

**Your Task:**  
You don't need to read input or print anything. Your task is to complete the function **count()** which takes the array of integers **arr,** **n,**and**x**as parameters and returns an integer denoting the answer.  
If x is not present in the array (arr) then return 0.

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

**Constraints:**  
1 ≤ N ≤ 105  
1 ≤ Arr[i] ≤ 106  
1 ≤ X ≤ 106

**Code :-**

//{ Driver Code Starts

#include<bits/stdc++.h>

using namespace std;

// } Driver Code Ends

//User function template for C++

class Solution{

public:

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

int low=start, high=n-1, ans=-1;

while(low <= high){

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

if(arr[mid] < target)

low = mid + 1;

else{

if(arr[mid] == target)

ans = mid;

high = mid - 1;

}

}

return ans;

}

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

int low=start, high=n-1, ans=-1;

while(low <= high){

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

if(arr[mid] > target)

high = mid - 1;

else{

if(arr[mid] == target)

ans = mid;

low = mid + 1;

}

}

return ans;

}

/\* if x is present in arr[] then returns the count

of occurrences of x, otherwise returns 0. \*/

int count(int arr[], int n, int x) {

int start = lower(arr, n, x, 0);

if(start != -1){

int end = upper(arr, n, x, start);

return end-start+1;

}

return 0;

}

};

//{ Driver Code Starts.

int main() {

int t;

cin >> t;

while (t--) {

int n, x;

cin >> n >> x;

int arr[n];

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

cin >> arr[i];

}

Solution ob;

auto ans = ob.count(arr, n, x);

cout << ans << "\n";

}

return 0;

}

// } Driver Code Ends

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

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