Classwork

First and Last Position of an Element In Sorted Array [Coding Ninjas]

Question Link:-
https://www.codingninjas.com/studio/problems/first-and-last-position-of-an-element-in-sorted-array_1082549?leftPanelTab=0

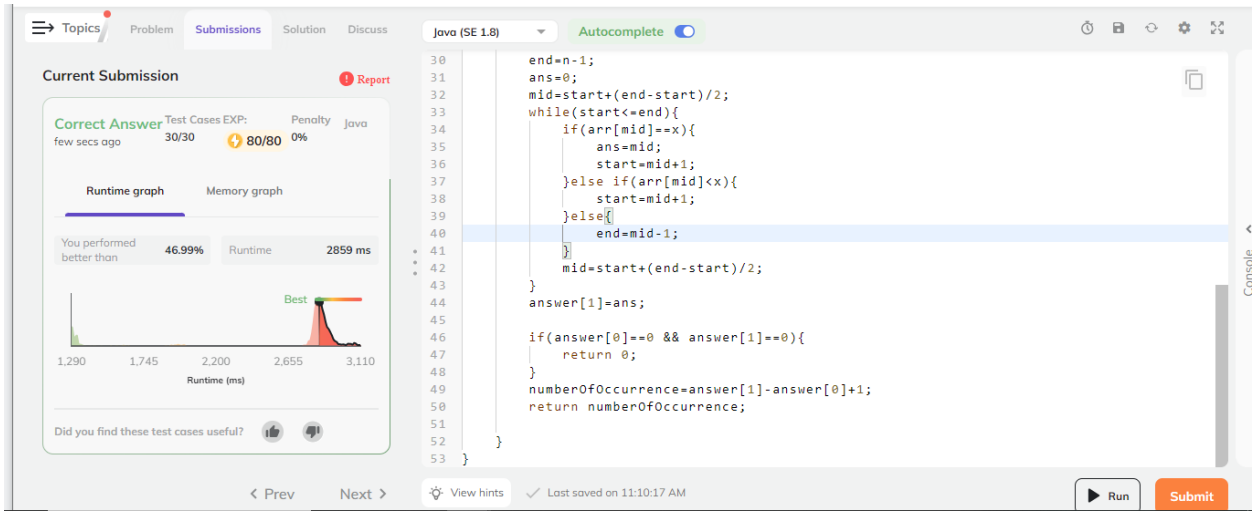
Java Code

```
import java.io.*;
import java.util.*;
public class Solution {

    public static int[] firstAndLastPosition(ArrayList<Integer> arr, int n, int k) {
        // Write your code here.
        int[] answer=new int[2];

        //First occurrence
        int start=0;
        int end=n-1;
        int ans=-1;
        int mid=start+(end-start)/2;
        while(start<=end){
            if(arr.get(mid)==k){
                ans=mid;
                end=mid-1;
            }else if(arr.get(mid)<k){
```

```
                start=mid+1;
            }else{
                end=mid-1;
            }
            mid=start+(end-start)/2;
        }
        answer[0]=ans;

        //Last occurrence
        start=0;
        end=n-1;
        ans=-1;
        mid=start+(end-start)/2;
        while(start<=end){
            if(arr.get(mid)==k){
                ans=mid;
                start=mid+1;
            }else if(arr.get(mid)<k){
                start=mid+1;
            }else{
                end=mid-1;
            }
            mid=start+(end-start)/2;
        }
        answer[1]=ans;

        return answer;
    }

};
```

Number of occurrence [Coding Ninjas]

Question link:-
https://www.codingninjas.com/studio/problems/occurrence-of-x-in-a-sorted-array_630456?leftPanelTab=0



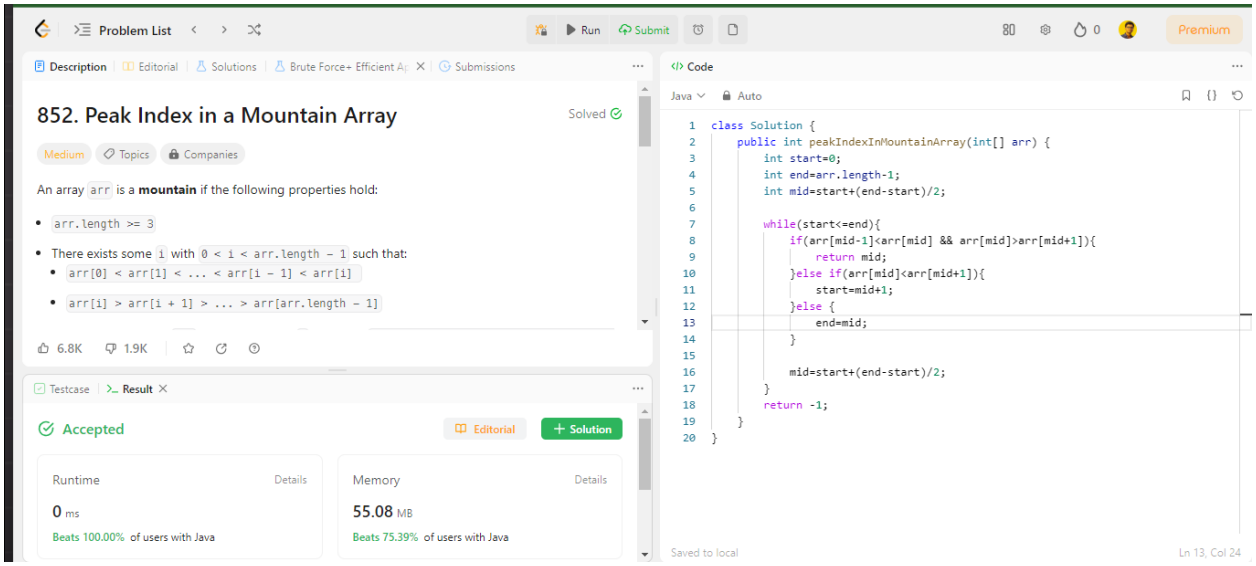
Java Code

```
public class Solution {
    public static int count(int arr[], int n, int x) {
        //Your code goes here
         int[] answer=new int[2];
         answer[0]=0;
         answer[1]=0;
         int numberOfOccurrence=0;

        //First occurrence
        int start=0;
        int end=n-1;
        int ans=0;
        int mid=start+(end-start)/2;
        while(start<=end){
            if(arr[mid]==x){
                ans=mid;
                end=mid-1;
            }else if(arr[mid]<x){
                start=mid+1;
            }else{
                end=mid-1;
            }
            mid=start+(end-start)/2;
        }
        answer[0]=ans;
```

```
        //Last occurrence
        start=0;
        end=n-1;
        ans=0;
        mid=start+(end-start)/2;
        while(start<=end){
            if(arr[mid]==x){
                ans=mid;
                start=mid+1;
            }else if(arr[mid]<x){
                start=mid+1;
            }else{
                end=mid-1;
            }
            mid=start+(end-start)/2;
        }
        answer[1]=ans;

        if(answer[0]==0 && answer[1]==0){
            return 0;
        }
        numberOfOccurrence=answer[1]-answer[0]+1;
        return numberOfOccurrence;

    }
}
```

Peak index in a Mountain Array [LeetCode]

Question Link:-
https://leetcode.com/problems/peak-index-in-a-mountain-array/description/



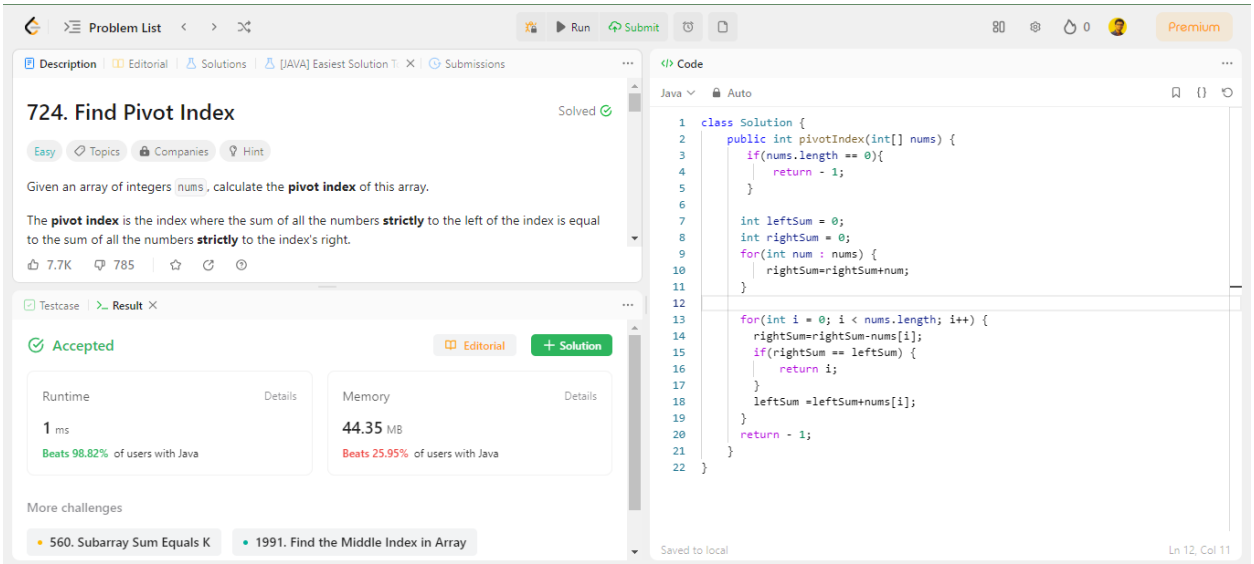
Java Code

```java
class Solution {
    public int peakIndexInMountainArray(int[] arr) {
        int start=0;
        int end=arr.length-1;
        int mid=start+(end-start)/2;

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

            mid=start+(end-start)/2;
        }
        return -1;
    }
}
```

Homework

Find Pivot Index [LeetCode]

Question Link:-
https://leetcode.com/problems/find-pivot-index/description/



Java Code

```java
class Solution {
    public int pivotIndex(int[] nums) {
        if(nums.length == 0){
            return - 1;
        }

        int leftSum = 0;
        int rightSum = 0;
        for(int num : nums) {
            rightSum=rightSum+num;
        }

        for(int i = 0; i < nums.length; i++) {
          rightSum=rightSum-nums[i];
          if(rightSum == leftSum) {
              return i;
          }
          leftSum =leftSum+nums[i];
        }
        return - 1;
    }
}
```