Module-1

Study / Learning

MCT

Coding

Mock Interview

1 : 1

1 hr

3 coding

1 question - 20 min

- explain question
- Approach
- coding part

Module-2

Java

Take an **array** as input. Given condition is that the array contains all the **unique** elements. Then take **k** as an integer input and print all the combinations of the pairs whose **absolute difference is k**.

Input Format

First line contains an integer number **n** representing size of array.

Second line contains **n** integer numbers representing elements of the array.

Third line contains an integer number **k**

Constraints

```
1 <= n <= 100000
0 <= array[index] <= 100000
0 <= k <= 1000000
```

Output Format

Print the required combinations in different line.

Sample Input 0

```
5
1 2 3 4 5
3
```

Sample Output 0

```
1 4
2 5
```

Explanation 0

|arr[0] - arr[3]| = |1 - 4| = 3

|arr[1] - arr[4]| = |2 - 5| = 3

Submitted Code

Language: Java 15

```
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
       Scanner sc = new Scanner(System.in);
        int n =sc.nextInt();
        int arr[] = new int[n];
        for(int i=0;i<arr.length;i++)arr[i]=sc.nextInt();

        int k=sc.nextInt();
        findDiffer(arr,k);
    }
    public static void findDiffer(int arr[] , int k){
        for(int i=0;i<arr.length;i++){
            for(int j=i;j<arr.length;j++){
                if(Math.abs(arr[i]-arr[j])==k)System.out.println(arr[i]+" "+arr[j]);
            }
        }
    }
}
```

10 - Marks

① explain — 20%.

② q/Approach — 40%

③ q/Approach/code test — 90%.

q/A/code/TC — 100%.

⑦ easy

⑦ Midium

⑥ Advance

Given the array **nums**, for each **nums[i]** find out how many numbers in the array are **smaller** than it. That is, for each **nums[i]** you have to **count** the number of valid j's such that **j != i** and **nums[j] < nums[i]**

NOTE :-

After answering the question, attempt the related question in the linked resource to improve your understanding of the question . Click here

Input Format

First line contains an integer **N** representing the size of array.

Second line contains **N** number of integers representing the elements of array.

Constraints

```
0 <= N <= 10000

0 <= arr[i] <= 10000
```

Output Format

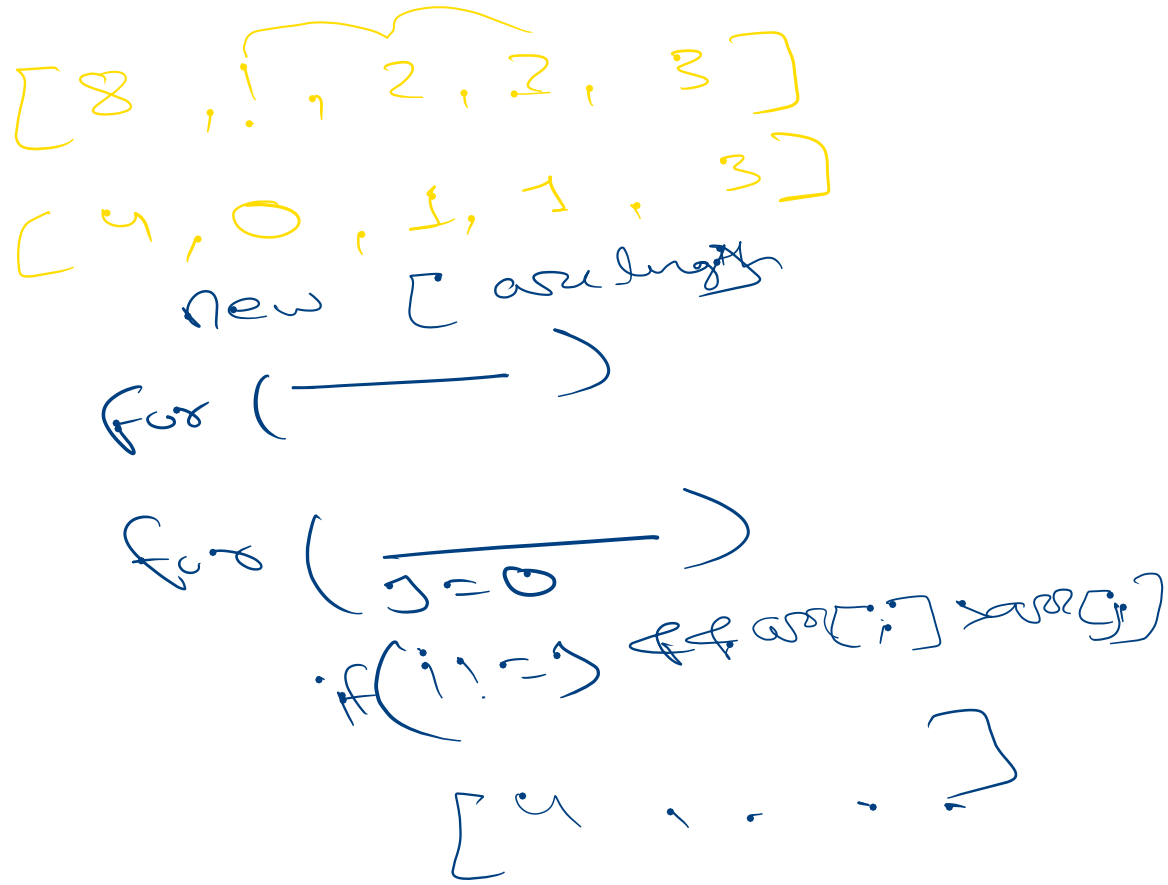
Print the required answer.

Sample Input 0

```
5
8 1 2 2 3
```

Sample Output 0

```
4 0 1 1 3
```



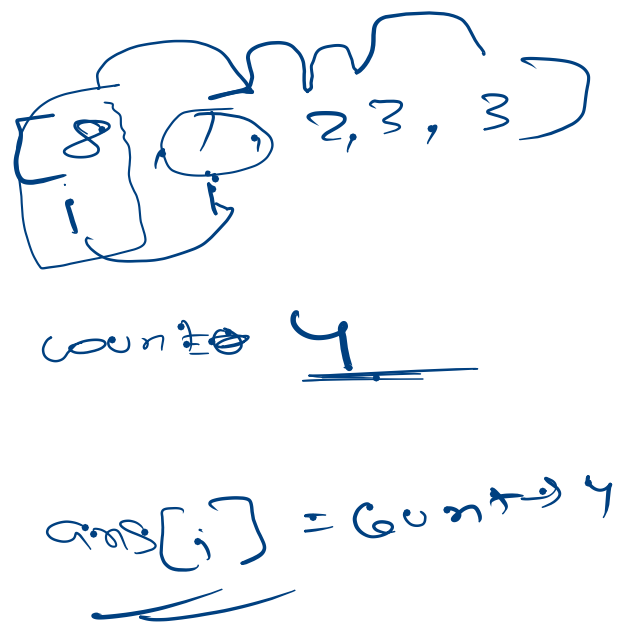
Submitted Code

Language: Java 15

```java
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int arr[]= new int[n];
        for(int i=0;i<n;i++){
            arr[i]=sc.nextInt();
        }
        int ans []= countSmaller(arr);
         for(int i=0;i<ans.length;i++){
            System.out.print(ans[i]+" ");
        }
    }
    public static int[] countSmaller(int [] arr){
        int ans []= new int[arr.length];
        for(int i=0;i<arr.length;i++){
            int count=0;
            for(int j=0;j<arr.length;j++){
                if(i!=j && arr[i]>arr[j])count++;
            }
            ans[i]=count;
        }
        return ans;
    }
}
```



Given an array **arr[]** of size N of positive integers(1 - N). One number 'A' from set **{1, 2, ...N}** is **missing** and one number 'B' **occurs twice** in array. Write a program to print the **repeating element** and the **missing element** in array.

Input Format

First line contains integer input N (size of array).

Second line contains n integers of arr[] elements.

Constraints

```
1 ≤ N ≤ 10^4

1 ≤ arr[i] ≤ N
```

Output Format

In the first line print repeating element.

In the second line contains print missing element.

Sample Input 0

```
4
1 2 4 4
```

Sample Output 0

```
4
3
```

Explanation 0

Repeating number is 4 and smallest positive missing number is 3 in given array.

Submitted Code

Language: Java 15

```java
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int [] arr = new int[n];
        for(int i=0;i<n;i++)arr[i]=sc.nextInt();
        repatMissing(arr);

    }
    public static void repatMissing(int [] arr){
        int rept = 0;
        int missing = 0;
        int arrElemetSum=0;
        int naturalNumSum=0;
        for(int i =0;i<arr.length;i++){
            arrElemetSum+=arr[i];
            naturalNumSum+=i+1;
            int count = 0;
            for(int j=0;j<arr.length;j++){
                if(arr[i]==arr[j])count++;
            }
            if(count==2){
                rept=arr[i];
            }
        }

        System.out.println(rept);
        int differ=Math.abs(naturalNumSum-arrElemetSum);
        if(arrElemetSum>naturalNumSum){
            System.out.println(rept-differ);
        }else{
        System.out.println(differ+rept);
    }
    }
}
```

Given an array of size **n** with **unique** integer elements. And then take **m** as an integer input. Declare the **second array** of size **m** that stores values of int data-type. Then take **m** integer inputs and store them in the array one by one.

Then **print all the elements** of the first array whose **absolute values** are present in the second array. You shouldn't use any extra space.

Input Format

First line contains integer number **n** representing size of first array.

Second line contains **n** integer inputs representing elements of first array.

Third line contains integer number **m** representing size of second array.

Fourth line contains **m** integer inputs representing elements of second array.

Constraints

```
1 <= n,m <= 100000

0 <= arr1[i],arr2[i] <= 100000
```

Output Format

An series on integer element in single line.

Sample Input 0

```
5
1 2 -2 4 -1
5
1 2 3 -2 5
```

Sample Output 0

```
1 2 -2 -1
```

Explanation 0

Second array contains only absolute value of 1 , 2 ,-2 and -1.