Started on	Saturday, 18 November 2023, 8:51 AM
State	Finished
Completed on	Saturday, 18 November 2023, 9:33 AM
Time taken	41 mins 52 secs
Marks	3.00/3.00
Grade	**15.00** out of 15.00 (**100**%)
Name	BALAJI S CSD

Question **1**

Correct

Mark 1.00 out of 1.00

In some array arr, the values were in arithmetic progression: the values arr[i+1] - arr[i] are all equal for every 0 <= i < arr.length - 1.

Then, a value from arr was removed that **was not the first or last value in the array**.

Return the removed value.

Example 1:

Input:

4

5 7 11 13

Output:

9

Explanation:

The previous array was [5,7,**9**,11,13].

Example 2:

Input:

3

15 13 12

Output:

14

Explanation:

The previous array was [15,**14**,13,12].

Constraints:

- 3 <= arr.length <= 1000
- 0 <= arr[i] <= 10^5

Answer: (penalty regime: 0 %)

Ace editor not ready. Perhaps reload page?
Falling back to raw text area.

```
import java.util.*;

public class Duplicate{
    public static void main(String []args){
        Scanner scan =new Scanner (System.in);
        int size =scan.nextInt();
        int [] x = new int[size];
        for(int i=0; i<size;i++){
            x[i] = scan.nextInt();

        }
        Arrays.sort(x);
        int diff =x[1]-x[0];
        int j;
        for(j=0;j<size;j++){
            if(x[j]+diff !=x[j+1]){
                System.out.println(x[j]+diff);
                break;
            }
        }
    }
}
```

	Input	Expected	Got	
✔	4 5 7 11 13	9	9	✔
✔	3 15 13 12	14	14	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Question **2**

Correct

Mark 1.00 out of 1.00

Given an array A of distinct integers sorted in ascending order, return the smallest index i that satisfies A[i] == i. Return -1 if no such i exists.

Example 1:

Input:

5

-10 -5 0 3 7

Output:

3

Explanation:

For the given array, A[0] = -10, A[1] = -5, A[2] = 0, A[3] = 3, thus the output is 3.

Example 2:

Input:

5

0 2 5 8 17

Output:

0

Explanation:

A[0] = 0, thus the output is 0.

Example 3:

Input:

6

-10 -5 3 4 7 9

Output:

-1

Explanation:

There is no such i that A[i] = i, thus the output is -1.

Note:

1. 1 <= A.length < 10^4
2. -10^9 <= A[i] <= 10^9

Answer: (penalty regime: 0 %)

Ace editor not ready. Perhaps reload page?
Falling back to raw text area.

```
import java.util.*;
public class Main{
    public static void main(String [] args){
        Scanner scan= new Scanner(System.in);
        int size =scan.nextInt();
        int []arr =new int [size];
        for (int j=0;j<size;j++){
            arr[j]=scan.nextInt();

        }
        int i;
        for(i=0;i<size;i++){
            if(arr[i]==i){
                break;
            }
        }
        if(i<size){
            System.out.println(i);
        }
        else
        System.out.println("-1");
    }
}
```

	Input	Expected	Got	
✔	5 -10 -5 0 3 7	3	3	✔
✔	6 -10 -5 3 4 7 9	-1	-1	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Question **3**

Correct

Mark 1.00 out of 1.00

Write a program to print all the locations at which a particular element (taken as input) is found in a list and also print the total number of times it occurs in the list. The location starts from 1.

For example, if there are 4 elements in the array:

5
6
5
7

If the element to search is 5 then the output will be:

5 is present at location 1
5 is present at location 3
5 is present 2 times in the array.

Sample Test Cases

Test Case 1

Input

4
5
6
5
7
5

Output

5 is present at location 1.
5 is present at location 3.
5 is present 2 times in the array.

Test Case 2

Input

5
67
80
45
97
100
50

Output

50 is not present in the array.

Answer: (penalty regime: 0 %)

Ace editor not ready. Perhaps reload page?
Falling back to raw text area.

```
import java.util.*;
public class Daa
{
    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 element=sc.nextInt();
        int count=0;
        for(int i=0;i<n;i++)
        {
            if(arr[i]==element)
            {
                System.out.println(element +" is present at location "+(i+1)+".");
                count++;
            }
        }
        if (count==0){
        System.out.println(element +" is not present in the array.");

           }
        else{
        System.out.println(element +" is present "+count+" times in the array. ");
            }
    }
}
```

	Input	Expected	Got	
✔	4 5 6 5 7 5	5 is present at location 1. 5 is present at location 3. 5 is present 2 times in the array.	5 is present at location 1. 5 is present at location 3. 5 is present 2 times in the array.	✔
✔	5 67 80 45 97 100 50	50 is not present in the array.	50 is not present in the array.	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.