https://course.acciojob.com/idle?question=fad89b8e-c73f-4a75-9de6-a5b045e60394

- EASY
- Max Score: 30 Points

Array11

Given an array of integers. Compute recursively the number of times that the value `11` appears in the array. We'll use the convention of considering only the part of the array that begins at the given index. In this way, a recursive call can pass `index+1` to move down the array. The initial call will pass in the index as `0`.

array11([1, 2, 11], 0) → 1

array11([11, 11], 0) → 2

array11([1, 2, 3, 4], 0) → 0

Input Format

The first line contains the number n, the size of the array

The second line contains `N` integers

You need to complete the `array11` function, which contains the `nums` array of size `N` and an integer `index` and finally returns the answer.

Output Format

Print the number of `11` in the array.

Example 1

Input

```
5
```

```
1 3 11 11 2
```

Output

```
2
```

Explanation

11 occur twice

Example 2

Input

```
6
1 8 9 12 11
```

Output

```
1
```

Explanation

11 occur once

Constraints

`2 <= N <= 3000`

`0 <= A[i] <= 5000`

Topic Tags

- **Recursion**
- **Arrays**

My code

```
// n java
import java.util.*;
import java.lang.*;
import java.io.*;

public class Main
{
static int fun(int arr[],int n)
      {
        if(n==0) return 0;
        if(arr[n-1]==11) return (1+fun(arr,n-1));
        else return fun(arr,n-1);
      }
        public static void main (String[] args) throws
java.lang.Exception
        {
                //your code here

      Scanner s=new Scanner(System.in);
      int n=s.nextInt();
      int arr[]=new int[n];
      for(int i=0;i<n;i++)
        arr[i]=s.nextInt();
          int a=fun(arr,n);
      System.out.print(a);
```

```
        }
    }
```