https://course.acciojob.com/idle?question=9cde0b6f-2f07-4283-b698-237ea7b82b69

- EASY
- Max Score: 30 Points

Remove Element

Given an integer array `nums` and an integer `val`, remove all occurrences of `val` in `nums` in-place. The relative order of the elements may be changed.

Since it is impossible to change the length of the array in some languages, you must instead have the result be placed in the first part of the array `nums`. More formally, if there are `k` elements after removing the duplicates, then the first `k` elements of `nums` should hold the final result. It does not matter what you leave beyond the first `k` elements.

Return `k` after placing the final result in the first `k` slots of `nums`.

Do not allocate extra space for another array. You must do this by modifying the input array in-place with O(1) extra memory.

Input Format

The first line of input contains integers `n` and `val` representing the size of array `nums` and `value` integer respectively.

The second line of input contains `n` space-separated integers representing the elements of array `nums`.

Output Format

The only line of output contains a single integer `k` representing the size of array after removing all occurrences of `val` in `nums`.

Example 1

Input

```
4 3
3 2 2 3
```

Output

```
2
```

Explanation

Your function should return k = 2, with the first two elements of nums being 2. It does not matter what you leave beyond the returned k (hence they are underscores).

Example 2

Input

```
8 2
0 1 2 2 3 0 4 2
```

Output

```
5
```

Explanation

Your function should return k = 5, with the first five elements of nums containing 0, 0, 1, 3, and 4. Note that the five elements can be returned in any order. It does not matter what you leave beyond the returned k (hence they are underscores).

Constraints:

`0 <= nums.length <= 100`

`0 <= nums[i] <= 50`

`0 <= val <= 100`

Topic Tags

- 2-Pointers
- Arrays

My code

```
// in java
import java.util.*;

class Solution {
    public int removeElement(int[] nums, int val) {
        // Write your code here
                int n=nums.length;
                int c=0;
                for(int i=0;i<n;i++)
                        if(nums[i]==val)
                                c++;
                return n-c;
    }
}
```

```
public class Main{
    public static void main(String[] args) throws Exception {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int val=sc.nextInt();
        int[] arr = new int[n];
        for (int i = 0; i < n; i++) {
            arr[i] = sc.nextInt();
        }
        sc.close();
        Solution Obj = new Solution();
        System.out.println(Obj.removeElement(arr, val));
    }
}
```