Question **1**

Correct

Mark 1.00 out of 1.00

Given an array `nums` of size `n`, return *the majority element*.

The majority element is the element that appears more than `⌊n / 2⌋` times. You may assume that the majority element always exists in the array.

Example 1:

```
Input: nums = [3,2,3]
Output: 3
```

Example 2:

```
Input: nums = [2,2,1,1,1,2,2]
Output: 2
```

Constraints:

- `n == nums.length`
- `1 <= n <= 5 * 10^4`
- `-2^31 <= nums[i] <= 2^31 - 1`

For example:

Input	Result
3 3 2 3	3
7 2 2 1 1 1 2 2	2

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
#include<stdlib.h>
int count(int* arr,int ele,int left,int right){
    int c = 0;
    for(int i=left;i<right;i++){
        if(arr[i]==ele) c++;
    }
    return c;
}
int majorityElement(int *arr,int left,int right){
    if(left== right) return arr[left];
    int mid = (int)(left+right)/2;
    int leftmajority = majorityElement(arr,left,mid);
    int rightmajority = majorityElement(arr,mid+1,right);
    if(leftmajority == rightmajority) return leftmajority;
    else{
        int leftcount =count(arr,leftmajority,left,right);
        int rightcount = count(arr,rightmajority,left,right);
        return leftcount> rightcount? leftmajority:rightmajority;
    }
}

int main(){
    int n;
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++) scanf("%d",&arr[i]);
    printf("%d",majorityElement(arr,0,n-1));
    return 0;
}
```

	Input	Expected	Got	
✔	3 3 2 3	3	3	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

◄ 1-Number of Zeros in a Given Array

Jump to...

3-Finding Floor Value ►