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>

int count(int arr[], int low, int high, int x)
{
    if ((low > high)|| (low == high && arr[low] != x))
        {return 0;}

    if (low == high && arr[low] == x)
        {return 1;}

    return count(arr, low,(low + high) / 2, x)+ count(arr, 1 + (low + high) / 2,high, x);
}


int main()
{
   int n;
    scanf("%d",&n);
    int arr[n] , a[n];
    for(int i=0;i<n;i++){
        scanf("%d",&arr[i]);
    }

    for(int i=0;i<n;i++){
    a[i] =count(arr, 0, n - 1, arr[i]);
    }

    int m;
    int max = a[0];

    for (int i = 0; i < n; i++){
        if (a[i] > max)
            max =arr[i];
             m=i;
```

```
    }

    printf("%d",arr[m]);
}
```

	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 ►