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 element(int nums[], int low, int high)
{
    if (low == high)
    {
        return nums[low];
    }
    int mid = (low + high) / 2;
    int left = element(nums, low, mid);
    int right= element(nums, mid + 1, high);
    if (left== right) {
        return left;
    }

    int lcount = 0, rcount = 0;
    for (int i = low; i <= high; i++) {
        if (nums[i] == left)
        {
            lcount++;
        }
        if (nums[i] == right)
        {
            rcount++;
        }
    }
    if(lcount > rcount)
    {
```

```
        return left;
    }
    else
    {
        return right;
    }
}
int majorityelement(int nums[], int n) {
    return element(nums, 0, n - 1);
}
int main() {
    int n;
    scanf("%d", &n);

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

    printf("%d\n", majorityelement(nums, n));

    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 ►