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 %)

```c
#include <stdio.h>

int majorityElement(int nums[], int n) {
    int candidate = nums[0];
    int count = 1;

    for (int i = 1; i < n; i++) {
        if (nums[i] == candidate) {
            count++;
        } else {
            count--;
            if (count == 0) {
                candidate = nums[i];
                count = 1;
            }
        }
    }

    count = 0;
    for (int i = 0; i < n; i++) {
        if (nums[i] == candidate) {
            count++;
        }
    }

    if (count > n / 2) {
        return candidate;
    } else {
        return -1;
    }
}

int main() {
    int n;
    scanf("%d", &n);

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

    int majority = majorityElement(nums, n);
    if (majority != -1) {
        printf("%d\n", majority);
    } else {
        printf("No majority element found\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 ►