Problem 2: Divide And Conquer Method

Question:

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`

```c
#include <stdio.h>

int majorityElement(int* nums, int numsSize) {
    int count = 0;
    int candidate = 0;

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

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

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

    int result = majorityElement(nums, n);
    printf("%d\n", result);

    return 0;
}
```

| | Input | Expected | Got | |
|---|---|---|---|---|
| ✔ | 3<br>3 2 3 | 3 | 3 | ✔ |

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.