Started on	Wednesday, 17 September 2025, 10:55 AM
State	Finished
Completed on	Wednesday, 17 September 2025, 11:10 AM
Time taken	15 mins
Marks	1.00/1.00
Grade	**10.00** out of 10.00 (**100**%)

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

Input	Result
7 2 2 1 1 1 2 2	2

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>

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

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

    int maxCount = 0;
    int majorityElement = a[0];

    for (int i = 0; i < n; i++) {
        int count = 0;
        for (int j = 0; j < n; j++) {
            if (a[i] == a[j]) {
                count++;
            }
        }
        if (count > maxCount) {
            maxCount = count;
            majorityElement = a[i];
        }
    }

    printf("%d\n", majorityElement);

    return 0;
}

```

	Input	Expected	Got	
✔	3 3 2 3	3	3	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Back to Course