Started on	Thursday, 4 September 2025, 9:13 AM
State	Finished
Completed on	Thursday, 4 September 2025, 9:40 AM
Time taken	26 mins 51 secs
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
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 majority(int l, int r)
    {
        if (l == r) return a[l];
        int m = (l + r) / 2;
        int left = majority(l, m);
        int right = majority(m + 1, r);
        if (left == right) return left;

        int cl = 0, cr = 0;
        for (int i = l; i <= r; i++)
        {
            if (a[i] == left) cl++;
            else if (a[i] == right) cr++;
        }
        return cl > cr ? left : right;
    }
    printf("%d\n", majority(0, n - 1));
    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