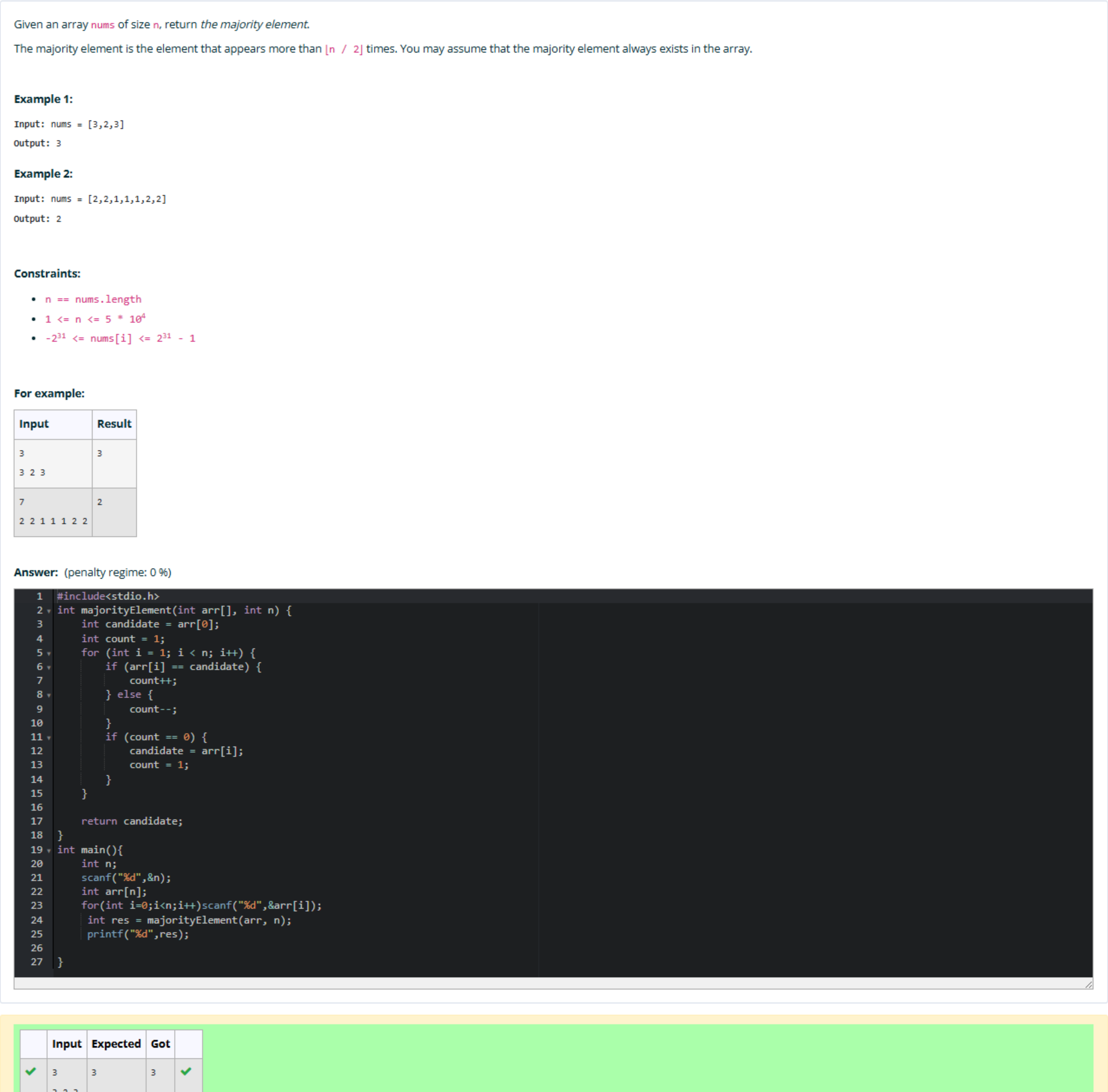

Given an array `nums` of size `n`, return *the majority element*.

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

    return candidate;
}
int main(){
    int n;
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++)scanf("%d",&arr[i]);
     int res = majorityElement(arr, n);
     printf("%d",res);

}
```

	Input	Expected	Got	
✓	3 3 2 3	3	3	✓