# 2-Majority Element

| | |
|---|---|
| **Started on** | Friday, 19 September 2025, 2:22 PM |
| **State** | Finished |
| **Completed on** | Saturday, 20 September 2025, 8:50 PM |
| **Time taken** | 1 day 6 hours |
| **Marks** | 1.00/1.00 |
| **Grade** | **10.00** out of 10.00 (**100**%) |

**Question 1** | Correct Mark 1.00 out of 1.00 Flag 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 - 1`

**For example:**

| Input | Result |
|---|---|
| 3<br>3 2 3 | 3 |
| 7<br>2 2 1 1 1 2 2 | 2 |

**Answer:** (penalty regime: 0 %)

```
// #include <stdio.h>

// int majorityElement(int arr[], int n) {
//     int count = 0, candidate = 0;

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

//     return candidate;
// }

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

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

//     printf("%d\n", majorityElement(arr, n));
// }

#include <stdio.h>
#include <stdlib.h>

int cmpfunc(const void* a, const void* b) {
    return (*(int*)a - *(int*)b);
}

int majorityElement(int* nums, int n) {
    qsort(nums, n, sizeof(int), cmpfunc);
    return nums[n / 2];
}
```

```
42
43  int main() {
44      int n;
45      scanf("%d", &n);
46
47      int nums[n];
48      for (int i = 0; i < n; i++) {
49          scanf("%d", &nums[i]);
50      }
51
52      printf("%d\n", majorityElement(nums, n));
```

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

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Finish review

Back to Course