

[Dashboard](#) / [My courses](#) / [CS23331-DAA-2023-CSE](#) / [Divide and Conquer](#) / [2-Majority Element](#)

<b>Started on</b>	Thursday, 12 September 2024, 10:23 AM
<b>State</b>	Finished
<b>Completed on</b>	Thursday, 12 September 2024, 10:29 AM
<b>Time taken</b>	5 mins 58 secs
<b>Marks</b>	1.00/1.00
<b>Grade</b>	<b>10.00</b> out of 10.00 ( <b>100%</b> )

## 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  $\lfloor n / 2 \rfloor$  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:**

- $n == \text{nums.length}$
- $1 \leq n \leq 5 \cdot 10^4$
- $-2^{31} \leq \text{nums}[i] \leq 2^{31} - 1$

**For example:**

Input	Result
3 3 2 3	3
7 2 2 1 1 1 2 2	2

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

```

1  #include <stdio.h>
2  int m(int arr[],int n) {
3      int ca=0;
4      int c=0;
5      for (int i=0;i<n;i++) {
6          if (c==0) {
7              ca= arr[i];
8          }
9          c+=(arr[i]==ca) ? 1 : -1;
10     }
11     c=0;
12     for (int i=0;i<n;i++) {
13         if (arr[i]==ca) {
14             c++;
15         }
16     }
17     return (c>n/2) ? ca : -1;
18 }
19 int main() {
20     int n;
21     scanf("%d", &n);
22
23     int arr[n];
24     for (int i=0;i<n;i++) {
25         scanf("%d",&arr[i]);
26     }
27     printf("%d",m(arr,n));
28     return 0;
29 }
```

	Input	Expected	Got	
✓	3 3 2 3	3	3	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

◀ 1-Number of Zeros in a Given Array

Jump to...

3-Finding Floor Value ▶