

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

<b>Started on</b>	Thursday, 5 September 2024, 10:13 AM
<b>State</b>	Finished
<b>Completed on</b>	Thursday, 5 September 2024, 11:15 AM
<b>Time taken</b>	1 hour 2 mins
<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 == nums.length`
- `1 <= n <= 5 * 104`
- `-231 <= nums[i] <= 231 - 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 majority(int nums[], int low, int high)
3  {
4      if (low==high)
5          return nums[low];
6      int mid=(low+high)/2;
7      int left=majority(nums,low, mid);
8      int right=majority(nums, mid + 1,high);
9      if (left==right)
10         return left;
11     int lc=0;
12     for (int i=low;i<=high;i++)
13         if (nums[i] == left)
14             lc++;
15     int rc=0;
16     for (int i=low;i<=high;i++)
17         if (nums[i]==right)
18             rc++;
19     if (lc>(low-high+1)/2)
20         return left;
21     if (rc>(low-high+1)/2)
22         return right;
23     return -1;
24 }
25 int main()
26 {
27     int n;
28     scanf("%d",&n);
29     int nums[n];

```

```
30     for(int i=0;i<n;i++)
31         scanf("%d",&nums[i]);
32     printf("%d",majority(nums,0,n-1));
33     return 0;
34 }
35
```

	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 ▶