

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

<b>Started on</b>	Tuesday, 3 September 2024, 2:43 PM
<b>State</b>	Finished
<b>Completed on</b>	Tuesday, 1 October 2024, 2:36 PM
<b>Time taken</b>	27 days 23 hours
<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:**

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

**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
3  int count(int arr[], int low, int high, int x)
4  {
5      if ((low > high) || (low == high && arr[low] != x))
6          {return 0;}
7
8      if (low == high && arr[low] == x)
9          {return 1;}
10
11     return count(arr, low, (low + high) / 2, x) + count(arr, 1 + (low + high) / 2, high, x);
12 }
13
14
15 int main()
16 {
17     int n;
18     scanf("%d", &n);
19     int arr[n], a[n];
20     for(int i=0; i<n; i++){
21         scanf("%d", &arr[i]);
22     }
23
24     for(int i=0; i<n; i++){
25         a[i] = count(arr, 0, n - 1, arr[i]);
26     }
27
28     int m;
29     int max = a[0];
30
31     for (int i = 0; i < n; i++){
32         if (a[i] > max)
33             max = arr[i];
34         m=i;

```

```
35     }
36
37     printf("%d",arr[m]);
38 }
39
40
```

	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 ▶