Started on	Friday, 13 September 2024, 1:39 PM
State	Finished
Completed on	Friday, 4 October 2024, 2:53 PM
Time taken	21 days 1 hour
Marks	1.00/1.00
Grade	**10.00** out of 10.00 (**100**%)

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 `⌊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 find_majority(int arr[], int low, int high) {

    if (low == high) {
        return arr[low];
    }


    int mid = (low + high) / 2;
    int left_majority = find_majority(arr, low, mid);
    int right_majority = find_majority(arr, mid + 1, high);


    if (left_majority == right_majority) {
        return left_majority;
    }

    // Count the occurrences of the left and right majority eleme
    int left_count = 0;
    int right_count = 0;
    for (int i = low; i <= high; i++) {
        if (arr[i] == left_majority) {
            left_count++;
        } else if (arr[i] == right_majority) {
            right_count++;
        }
    }


    if (left_count > right_count) {
        return left_majority;
    } else {
```

```
        return right_majority;
    }
}


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

    int arr[n];

    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
  int majority=  find_majority(arr,0,n-1);
  printf("%d",majority);

```

	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 ►