Problem 2: Divide And Conquer Method

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:

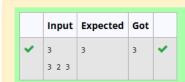
```
Input: nums = [3,2,3]
Output: 3
Example 2:
Input: nums = [2,2,1,1,1,2,2]
Output: 2
```

Constraints:

```
    n == nums.length
    1 <= n <= 5 * 10<sup>4</sup>
    -2<sup>31</sup> <= nums[i] <= 2<sup>31</sup>
```

•

```
1
      #include <stdio.h>
  2
  3 * int majorityElement(int* nums, int numsSize) {
  4
          int count = 0;
          int candidate = 0;
  5
  6
  7 •
           for (int i = 0; i < numsSize; i++) {</pre>
              if (count == 0) {
  8 ,
  9
                    candidate = nums[i];
 10
                    count = 1;
 11 ,
               } else if (nums[i] == candidate) {
 12
                   count++;
 13 ,
               } else {
 14
                   count--;
 15
 16
 17
          return candidate;
      }
 18
 19
      int main() {
 20 +
          int n;
scanf("%d", &n);
 21
 22
 23
         int nums[n];
for (int i = 0; i < n; i++) {
    scanf("%d", &nums[i]);</pre>
24
 25 •
 26
 27
          }
 28
 29
          int result = majorityElement(nums, n);
 30
          printf("%d\n", result);
 31
 32
          return 0;
      }
 33
 34
```



Passed all tests! 🗸

Correct

Marks for this submission: 1.00/1.00.