4.b. Majority Element

Aim: 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] <=` $2^{31} - 1$

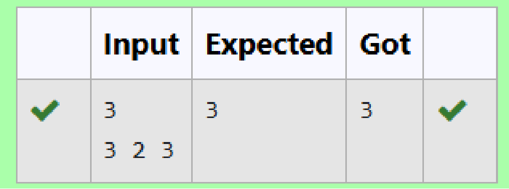
Algorithm:

```
int divide(a, l, r, n) {
    // base case: if left index equals right index
    if l is equal to r {
        return a[l] // return the only element
    }

    initialize mid as (l + r) / 2 // find the middle index

    // recursively divide the array
    initialize min as divide(a, l, mid, n) // find min in left half initialize
    max as divide(a, mid + 1, r, n) // find max in right half

    initialize leftc as 0 // counter for min occurrences
    initialize rightc as 0 // counter for max occurrences
    // count occurrences of min and max in the entire array
    for i from 0 to n – 1 {
        if a[i] is equal to min {
            increment leftc by 1 // count occurrences of min
        } else {
            increment rightc by 1 // count occurrences of max
        }
    }

    // check if min occurs more than n/2 times
    if leftc is greater than (n / 2) {
        return min // return min if it is the majority element
    } else {
        return max // return max otherwise
    }
}

int main() {
    initialize n // number of elements
    read n from user

    initialize a array of size n // array to hold input values

    // read values into the array
    for j from 0 to n – 1 {
        read a[j] from user
    }
    initialize l as 0 // left index
    initialize r as n - 1 // right index

    // call the divide function
    initialize result as divide(a, l, r, n)

    print result // output the final majority element
}
```

Program:

```
#include<stdio.h>
int divide(int a[],int l,int r,int n){
    if(l==r)
    {
        return a[l];
    }
    int mid=(l+r)/2;
    int min=divide(a,l,mid,n);
    int max=divide(a,mid+1,r,n);
    int leftc=0,rightc=0;
    for(int i=0;i<n;i++)
    {
        if(a[i]==min)
        {
            leftc++;
        }
        else
        {
            rightc++;
        }
    }
    if(leftc>(n/2))
    {
        return min;

    }
    else
    {
        return max;
    }
}
int main(){
    int n;
    scanf("%d",&n);
    int a[n];
    for(int j=0;j<n;j++){
        scanf("%d",&a[j]);
    }
    int l=0,r=n-1;
    int result=divide(a,l,r,n);
    printf("%d",result);
}
```

Output:



	Input	Expected	Got	
✓	3 3 2 3	3	3	✓