Majority element in C++

```
#include <iostream>
using namespace std;

int majority(int arr[], int n) {
    int res = 0, count = 1;
    for (int i = 1; i < n; i++) {
        if (arr[res] == arr[i]) {
            count++;
        } else {
            count--;
        }
        if (count == 0) {
            res = i;
            count = 1;
        }
    }

    count = 0;
    for (int i = 0; i < n; i++) {
        if (arr[res] == arr[i]) {
            count++;
        }
    }

    if (count <= n / 2) {
        res = -1;
    }
    return res;
}

int main() {
    int arr[] = {6, 8, 4, 8, 8};
    int n = sizeof(arr) / sizeof(arr[0]);

    cout << majority(arr, n) << endl;

    return 0;
}
```

Array Given:

```
arr[] = {6, 8, 4, 8, 8}
n = 5
```

We need to find the element (if any) that appears **more than 5 / 2 = 2** times.

🔁 Moore’s Voting Algorithm Dry Run

We'll go step-by-step through the first for loop which finds a *candidate*.

i	arr[i]	arr[res]	count	Explanation
0	6	6	1	Initial candidate at index 0
1	8	6	0	8 ≠ 6 → count--
		8	1	count = 0 → new candidate at index 1
2	4	8	0	4 ≠ 8 → count--
		4	1	count = 0 → new candidate at index 2
3	8	4	0	8 ≠ 4 → count--
		8	1	count = 0 → new candidate at index 3
4	8	8	2	8 == 8 → count++

Candidate Index: res = 3, arr[3] = 8

✓ Second loop: Confirm the candidate

We check how many times 8 appears in the array.

```
count = 0;
for (int i = 0; i < n; i++) {
    if (arr[i] == 8) count++;
}
```

8 appears **3 times** (at indices 1, 3, and 4).

Since 3 > 2, it **is** the majority element.

✓ Final Output

```
3
```

	That’s the index of the majority element 8.
3	