Check sorted in C++

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

bool sorted(int arr[], int n) {
    if (n == 1 || n == 0) {
        return true;
    } else if (arr[n - 1] < arr[n - 2]) {
        return false;
    } else {
        return sorted(arr, n - 1);
    }
}

int main() {
    int arr[] = {1, 2, 3, 4, 5};
    int n = sizeof(arr) / sizeof(arr[0]);
    cout << boolalpha << sorted(arr, n) << endl;
    return 0;
}
```

Input

```
arr[] = {1, 2, 3, 4, 5}
n = 5
```

🔄 Recursive Calls

We check if the last two elements are in correct order (`arr[n-2] <= arr[n-1]`), and recursively reduce the array size.

📋 Dry Run Table

Call	n	arr[n-2]	arr[n-1]	Comparison	Result
`sorted(arr, 5)`	5	4	5	4 ≤ 5	✅
`sorted(arr, 4)`	4	3	4	3 ≤ 4	✅
`sorted(arr, 3)`	3	2	3	2 ≤ 3	✅
`sorted(arr, 2)`	2	1	2	1 ≤ 2	✅
`sorted(arr, 1)`	1	—	—	Base case	✅

✅ Output

```
true
```

Your program will print:

```
true
```

Output:-
true