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;
}
```

Recursive Function Call Flow:

1. **Initial Call:**
   sorted(arr, 5)
   - arr[4] = 5 and arr[3] = 4 → 5 >= 4 → Continue checking with n = 4.
2. **Second Call:**
   sorted(arr, 4)
   - arr[3] = 4 and arr[2] = 3 → 4 >= 3 → Continue checking with n = 3.
3. **Third Call:**
   sorted(arr, 3)
   - arr[2] = 3 and arr[1] = 2 → 3 >= 2 → Continue checking with n = 2.
4. **Fourth Call:**
   sorted(arr, 2)
   - arr[1] = 2 and arr[0] = 1 → 2 >= 1 → Continue checking with n = 1.
5. **Base Case:**
   sorted(arr, 1)
   - n == 1 → Return true.

Backtracking:

- The base case returns true and propagates this result through all the previous recursive calls:
  - sorted(arr, 2) → true
  - sorted(arr, 3) → true
  - sorted(arr, 4) → true
  - sorted(arr, 5) → true

Output:-

```
true
```