Next Greater on the Right in C++

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

long* nextLargerElement(long* arr, int n)
{
    long* ans = new long[n];
    stack<int> st;
    for(int i = 0; i < n; i++){
        while(!st.empty() && arr[i] > arr[st.top()]){
            int idx = st.top();
            st.pop();
            ans[idx] = arr[i];
        }
        st.push(i);
    }
    while(!st.empty()){
        int idx = st.top();
        st.pop();
        ans[idx] = -1;
    }

    return ans;
}

int main() {
    long arr[] = {4, 8, 5, 2, 25};
    int n = sizeof(arr) / sizeof(arr[0]);

    long* result = nextLargerElement(arr, n);

    cout << "Resulting array:" << endl;
    for (int i = 0; i < n; i++) {
        cout << result[i] << " ";
    }
    cout << endl;

    delete[] result; // Free dynamically allocated
memory

    return 0;
}
```

Input:

arr = {4, 8, 5, 2, 25}
n = 5

📋 Iterative Dry Run Table:

i	arr[i]	Stack (indices)	Top Value	Condition Checked	Action Taken	ans Array
0	4	[]	—	—	Push index 0	[-, -, -, -, -]
1	8	[0]	4	8 > 4 → true	Pop 0, set ans[0] = 8, push 1	[8, -, -, -, -]
2	5	[1]	8	5 > 8 → false	Push 2	[8, -, -, -, -]
3	2	[1, 2]	5	2 > 5 → false	Push 3	[8, -, -, -, -]
4	25	[1, 2, 3]	2	25 > 2 → true	Pop 3, ans[3] = 25	[8, -, -, 25, -]
		[1, 2]	5	25 > 5 → true	Pop 2, ans[2] = 25	[8, -, 25, 25, -]
		[1]	8	25 > 8 → true	Pop 1, ans[1] = 25	[8, 25, 25, 25, -]
		[]	—	—	Push 4	[8, 25, 25, 25, -]
—	—	[4]	25	Loop ends	Pop 4, set ans[4] = -1	[8, 25, 25, 25, -1]

✅ Final Output:

Resulting array:
8 25 25 25 -1

Resulting array:
8 25 25 25 -1