Tapping Rain Water in C++

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

int getWater(int arr[], int n) {
    int res = 0;
    for (int i = 0; i < n; i++) {
        int lmax = arr[i];
        for (int j = 0; j < i; j++) {
            lmax = max(arr[j], lmax);
        }
        int rmax = arr[i];
        for (int j = i + 1; j < n; j++) {
            rmax = max(arr[j], rmax);
        }

        res += min(lmax, rmax) - arr[i];
    }
    return res;
}

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

Problem Explanation: Trapping Rain Water

At each index i, the amount of water it can hold is:

water_at_i = min(lmax, rmax) - arr[i]

Where:

- lmax: Max height to the left of i (including i)
- rmax: Max height to the right of i (including i)
- If min(lmax, rmax) - arr[i] > 0, it adds to total water trapped.

📊 Dry Run Table

Array: {3, 0, 1, 2, 5}

i	arr[i]	lmax (max left)	rmax (max right)	min(lmax, rmax)	Water at i = min(lmax, rmax) - arr[i]	res
0	3	3	5	3	0	0
1	0	3	5	3	3	3
2	1	3	5	3	2	5
3	2	3	5	3	1	6
4	5	5	5	5	0	6

✅ Final Output:

```
6
```

Output:

```
6
```