Temple offering In C++

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

int totalOfferings(int* height, int n) {
    int* larr = new int[n]; // Left offerings
array
    int* rarr = new int[n]; // Right offerings
array
    // Calculate left offerings
    larr[0] = 1;
    for (int i = 1; i < n; i++) {
        if (height[i] > height[i - 1]) {
            larr[i] = larr[i - 1] + 1;
        } else {
            larr[i] = 1;
        }
    }

    // Calculate right offerings
    rarr[n - 1] = 1;
    for (int i = n - 2; i >= 0; i--) {
        if (height[i] > height[i + 1]) {
            rarr[i] = rarr[i + 1] + 1;
        } else {
            rarr[i] = 1;
        }
    }

    // Calculate total offerings
    int ans = 0;
    for (int i = 0; i < n; i++) {
        ans += max(larr[i], rarr[i]);
    }

    // Free allocated memory
    delete[] larr;
    delete[] rarr;

    return ans;
}
int main() {
    int height[] = {2, 3, 5, 6, 4, 8, 9};
    int n = sizeof(height) / sizeof(height[0]);
    cout << totalOfferings(height, n) <<
endl;
    return 0;
}
```

Dry Run (Tabular)

Input:

```
height[] = {2, 3, 5, 6, 4, 8, 9}
```

Index `i`	Height `height[i]`	Left Offerings `larr[i]`	Right Offerings `rarr[i]`	Final Offerings `max(larr[i], rarr[i])`
0	2	1	1	1
1	3	2	1	2
2	5	3	1	3
3	6	4	2	4
4	4	1	1	1
5	8	2	2	2
6	9	3	3	3

Total Offerings:

```
1 + 2 + 3 + 4 + 1 + 2 + 3 = 16
```

✅ **Output:**

```
16
```

Output:-
16