Catalan in C++

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

int main() {
    int n = 6;
    int dp[n];
    dp[0] = 1;
    dp[1] = 1;

    for (int i = 2; i < n; i++) {
        dp[i] = 0;
        for (int j = 0; j < i; j++) {
            dp[i] += dp[j] * dp[i - j - 1];
        }
    }

    for (int i = 0; i < n; i++) {
        cout << dp[i] << " ";
    }

    return 0;
}
```

This is essentially using the Catalan number recurrence relation.

Iteration 1: i = 2

- dp[2] = 0
- For j = 0: dp[2] += dp[0] * dp[1] = 0 + 1 * 1 = 1
- For j = 1: dp[2] += dp[1] * dp[0] = 1 + 1 * 1 = 2
- So, dp[2] = 2.

Iteration 2: i = 3

- dp[3] = 0
- For j = 0: dp[3] += dp[0] * dp[2] = 0 + 1 * 2 = 2
- For j = 1: dp[3] += dp[1] * dp[1] = 2 + 1 * 1 = 3
- For j = 2: dp[3] += dp[2] * dp[0] = 3 + 2 * 1 = 5
- So, dp[3] = 5.

Iteration 3: i = 4

- dp[4] = 0
- For j = 0: dp[4] += dp[0] * dp[3] = 0 + 1 * 5 = 5
- For j = 1: dp[4] += dp[1] * dp[2] = 5 + 1 * 2 = 7
- For j = 2: dp[4] += dp[2] * dp[1] = 7 + 2 * 1 = 9
- For j = 3: dp[4] += dp[3] * dp[0] = 9 + 5 * 1 = 14
- So, dp[4] = 14.

Iteration 4: i = 5

- dp[5] = 0
- For j = 0: dp[5] += dp[0] * dp[4] = 0 + 1 * 14 = 14
- For j = 1: dp[5] += dp[1] * dp[3] = 14 + 1 * 5 = 19
- For j = 2: dp[5] += dp[2] * dp[2] = 19 + 2 * 2 = 23
- For j = 3: dp[5] += dp[3] * dp[1] = 23 + 5 * 1 = 28
- For j = 4: dp[5] += dp[4] * dp[0] = 28 + 14 * 1 = 42
- So, dp[5] = 42.

Final Output:

The dp array is:

1 1 2 5 14 42

Output:-
1 1 2 5 14 42