Stair Case in C++

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

// Function to calculate number of ways to reach nth step
int staircase(int n) {
    // Base cases
    if (n == 0 || n == 1) {
        return 1;
    }
    if (n == 2) {
        return 2;
    }
    // Recursive case
    return staircase(n-1) + staircase(n-2) + staircase(n-3);
}

int main() {
    // Test case
    int n = 7;
    cout << staircase(n) << endl;
    return 0;
}
```

Initial Call

The function staircase(7) is called.

- Base cases:
  - If n == 0, return 1
  - If n == 1, return 1
  - If n == 2, return 2

The recursive case is staircase(n-1) + staircase(n-2) + staircase(n-3).

For n = 7, we call:

staircase(7) = staircase(6) + staircase(5) + staircase(4)

Step 1: staircase(6)

- Call: staircase(6) = staircase(5) + staircase(4) + staircase(3)
- Let's break it down:

Step 1.1: staircase(5)

- Call: staircase(5) = staircase(4) + staircase(3) + staircase(2)
- Let's break it down:

Step 1.1.1: staircase(4)

- Call: staircase(4) = staircase(3) + staircase(2) + staircase(1)
- Let's break it down:

Step 1.1.1.1: staircase(3)

- Call: staircase(3) = staircase(2) + staircase(1) + staircase(0)
- Let's break it down:
  - staircase(2) = 2
  - staircase(1) = 1
  - staircase(0) = 1

So, staircase(3) = 2 + 1 + 1 = 4.

Step 1.1.1.2: staircase(2)

- Base case: staircase(2) = 2

Step 1.1.1.3: staircase(1)

- Base case: staircase(1) = 1

So, staircase(4) = 4 + 2 + 1 = 7.

	Step 1.2: staircase(3) • We already calculated that staircase(3) = 4. **Step 1.3: staircase(2)** • Base case: staircase(2) = 2. So, staircase(5) = 7 + 4 + 2 = 13. **Step 2: staircase(4)** We already calculated that staircase(4) = 7. **Step 3: staircase(3)** We already calculated that staircase(3) = 4. So, staircase(6) = 13 + 7 + 4 = 24. **Final Calculation: staircase(7)** Now that we have the values for staircase(6), staircase(5), and staircase(4), we can calculate staircase(7): staircase(7) = 24 + 13 + 7 = 44
Output:- 44	