|  |  |
| --- | --- |
| **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;  } | Dry Run Table for staircase(7) Track the **calls** and their **return values** from the bottom up (memoized-style for understanding):   | **n** | **staircase(n) Calculation** | **Result** | | --- | --- | --- | | 0 | 1 (base case) | 1 | | 1 | 1 (base case) | 1 | | 2 | 2 (base case) | 2 | | 3 | staircase(2) + staircase(1) + staircase(0) | 2 + 1 + 1 = 4 | | 4 | staircase(3) + staircase(2) + staircase(1) | 4 + 2 + 1 = 7 | | 5 | staircase(4) + staircase(3) + staircase(2) | 7 + 4 + 2 = 13 | | 6 | staircase(5) + staircase(4) + staircase(3) | 13 + 7 + 4 = 24 | | 7 | staircase(6) + staircase(5) + staircase(4) | 24 + 13 + 7 = 44 |  ✅ Final Output: 44 |
| Output:- 44 | |