|  |  |
| --- | --- |
| **Tiling with Dominoes in C++** | |
| #include <iostream>  #include <vector>  using namespace std;  int main() {      int n = 2;      vector<int> dp(n + 1);      dp[1] = 1;      dp[2] = 2;      for (int i = 3; i <= n; i++) {          dp[i] = dp[i - 1] + dp[i - 2];      }      cout << dp[n] << endl;      return 0;  } | Given:   * n = 2   We create a dp vector of size n+1 = 3 and initialize the base cases:   * dp[1] = 1 * dp[2] = 2   **Initial dp Table:**   | **i\dp** | **0** | **1** | **2** | | --- | --- | --- | --- | | **i=0** | 0 |  |  | | **i=1** |  | 1 |  | | **i=2** |  |  | 2 |   At this point:   * dp[0] = 0 (this entry is not used) * dp[1] = 1 * dp[2] = 2   **For Loop Execution:**  The for loop iterates from i = 3 to n. But, since n = 2, the loop condition i = 3 <= 2 is **false**, so the loop doesn't run.  **Final dp Table:**  The table remains unchanged from initialization:   | **i\dp** | **0** | **1** | **2** | | --- | --- | --- | --- | | **i=0** | 0 |  |  | | **i=1** |  | 1 |  | | **i=2** |  |  | 2 |   **Final Output:**  The program prints dp[n], where n = 2, so dp[2] = 2 is printed. |
| Output:- 2 | |