1.13 CLIMBING STAIRS PROBLEM – COUNT DISTINCT WAYS

Question:

You are climbing a staircase. It takes n steps to reach the top. Each time you can climb either 1 or 2 steps. In how many distinct ways can you climb to the top?

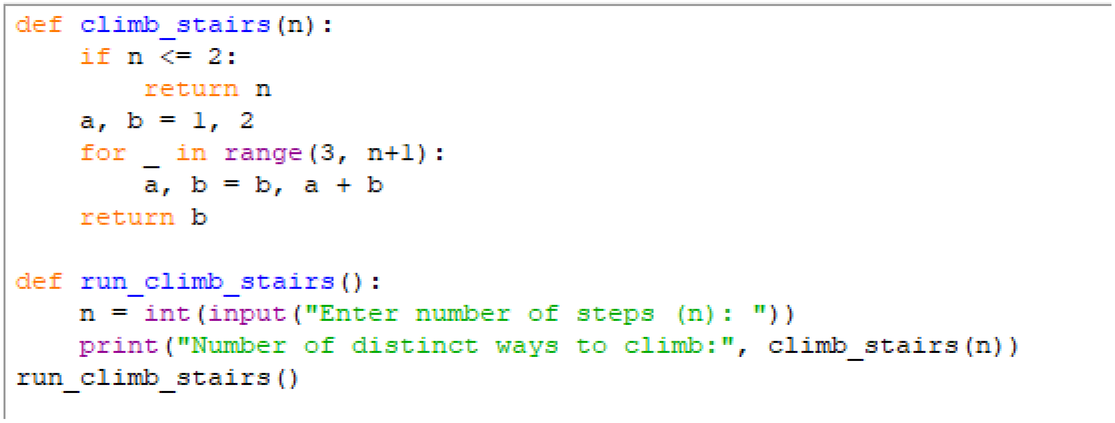
AIM:

To find the number of distinct ways to climb to the top of a staircase with n steps, where each move can be 1 step or 2 steps.

ALGORITHM:

1. Define a recursive function count(n) that returns the number of ways to reach step n.
2. If n <= 2, return n (base cases).
3. If result for n is already in memo, return it.
4. Otherwise, compute count(n-1) + count(n-2) and store it in memo.
5. Return count(n).

PROGRAM:



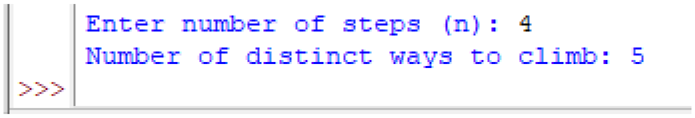
```
def climb_stairs(n):
    if n <= 2:
        return n
    a, b = 1, 2
    for _ in range(3, n+1):
        a, b = b, a + b
    return b

def run_climb_stairs():
    n = int(input("Enter number of steps (n): "))
    print("Number of distinct ways to climb:", climb_stairs(n))
run_climb_stairs()
```

Input:

n = 4

Output:



```
Enter number of steps (n): 4
Number of distinct ways to climb: 5
>>>
```

RESULT:

Thus the program is successfully executed, and the output is verified.

PERFORMANCE ANALYSIS:

- Time Complexity: O(n)
- Space Complexity: O(1)