- In that case when b is odd, the recursive call is made with b / 2. I will store the value return by power(a , b / 2) in result variable and will return square of result variable multiplied by a i.e (a * result * result).

Q4 : There are n stairs, a person standing at the bottom wants to reach the top. The person can climb either 1 stair or 2 stairs at a time. Count the number of ways the person can reach the top.

Examples:

Input: n = 1
Output: 1

There is only one way to climb 1 stair

Input: n = 2
Output: 2

There are two ways: (1, 1) and (2)

Input: n = 4
Output: 5

(1, 1, 1, 1), (1, 1, 2), (2, 1, 1), (1, 2, 1), (2, 2)

Solution:
Code : LP_Code5.java

Output :

```
Enter the number : 4
The number of ways to reach nth stair is : 5
```

Approach:

- We can easily find the recursive nature in the above problem. The person can reach the nth stair from either (n-1)th stair or from (n-2)th stair. Hence, for each stair n, we try to find out the number of ways to reach n-1th stair and n-2th stair and add them to give the answer for the nth stair. Therefore the expression for such an approach comes out to be :
- ways(n) = ways(n-1) + ways(n-2)
- The above expression is actually the expression for Fibonacci numbers, but there is one thing to notice, the value of ways(n) is equal to fibonacci(n+1).

  ways(1) = fib(2) = 1

  ways(2) = fib(3) = 2

  ways(3) = fib(4) = 3

  *Please refer to the approach of the fibonacci series.