**\*\*\*DP 2. Climbing Stairs | Learn How to Write 1D Recurrence Relations**

**Problem-** [**Climbing Stairs**](https://leetcode.com/problems/climbing-stairs/)(same as fibonacci number)

**class Solution {**

**public:**

**int climbStairs(int n) {**

**if(n <= 1) return 1;**

**int prev1 = 1;**

**int prev2 = 1;**

**for(int i=2; i<=n; i++){**

**int temp = prev1 + prev2;**

**prev2 = prev1;**

**prev1 = temp;**

**}**

**return prev1;**

**}**

**};**

**\*\*How to identify recursive problem? (recursion -> memoization)**

If in the ques it is asked:

i) Find all possible ways to do something or the best of all possible ways.

ii) Find the min/ max cost.

**\*\*Trick to convert into recurrence relation**

Step1. Identify a DP Problem.

Step2. To solve the problem after identification. (Express- explore- find ans)

1. Express the given problem in terms of index.

2. Explore all possible operations on that index according to the problem statement.

3. To count all possible ways - sum up all stuff.

To find minimum/maximum - Take Minimum/maximum of all stuff.