

# Lab 4 DSA Refresher

Strict Deadline will be Followed. After Deadline, no submission will be allowed.

Duration : 04:00 pm to 05:15 pm

## Problem Statement:

You are climbing a staircase. It takes  $n$  steps to reach the top. Each time you can either climb 1 or 2 steps. Write four functions to calculate the number of distinct ways to climb to the top, each with different time and space complexity specifications.

**Function 1:** Recursive Solution ( $O(2^n)$  Time Complexity)

**Function 2:** Using Extra Space ( $O(n)$  Time Complexity &  $O(n)$  Space Complexity)

**Function 3:** Faster Recursive Solution Without Extra Space ( $O(N)$  Time Complexity &  $O(1)$  Space Complexity)

**Function 4:** Optimized Approach without recursion ( $O(n)$  Time Complexity &  $O(1)$  Space Complexity)

**Note:** There should be only one cpp file in which one main function should be present and 4 functions (mentioned above) should be present.

**Hint:** In Function 3, instead of the last element, the last two elements should be returned.

## Marking Scheme:

*1st Function : 2 marks*

*2nd Function: 3 marks*

*3rd Function: 3 marks*

*4th Function: 2 marks*

**Total : 10 marks**

**Test Case 1:**

Input:  $n = 2$

Output: 2

Explanation: There are two ways to climb to the top.

1. 1 step + 1 step
2. 2 steps

**Test Case 2:**

Input:  $n = 3$

Output: 3

Explanation: There are three ways to climb to the top.

1. 1 step + 1 step + 1 step
2. 1 step + 2 steps
3. 2 steps + 1 step