**Experiment 6**

**Student Name: Mayank**  **UID:** 22BCS10612 **Branch:** BE-CSE **Section/Group:** 22BCS\_NTPP\_603’A’ **Semester:** 6th **Date of Performance:** 17-01-25

**Subject Name:** AP Lab-2 **Subject Code:** 22CSP-351

## Aim: Climbing Stairs

1. **Objective:**

You are climbing a staircase. It takes n steps to reach the top.

Each time you can either climb 1 or 2 steps. In how many distinct ways can you climb to the top?

## Implementation/Code:

class Solution {

public int climbStairs(int n) {

if (n == 0) return 1;

if (n == 1) return 1;

int[] dp = new int[n + 1];

dp[0] = 1;

dp[1] = 1;

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

dp[i] = dp[i - 1] + dp[i - 2];

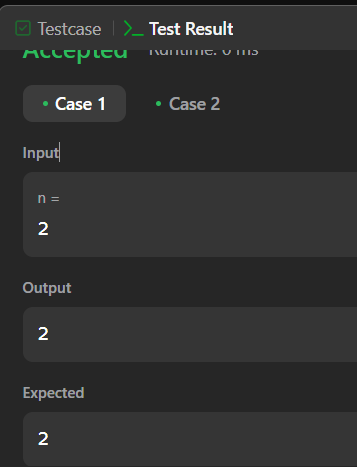
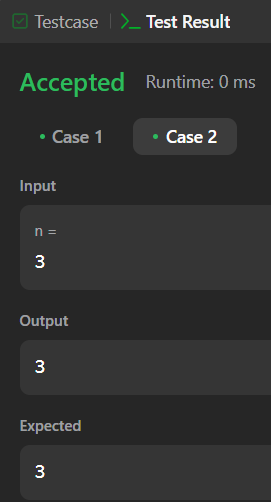
}

return dp[n];

}

}

## Output

****

1. **Learning Outcome:**

 Understand dynamic programming to solve recurrence-based problems.

 Learn how to use an array to store intermediate results for optimization.

 Recognize the Fibonacci-like nature of the "climbing stairs" problem.

 Implement iterative solutions to reduce redundant calculations.

 Improve problem-solving skills with bottom-up dynamic programming

# Question 2.

1. **Aim:** Maximum Subarray

## Objective:

Given an integer array nums, find the subarray with the largest sum, and return its sum.

## Implementation/Code:

class Solution {

    public int maxSubArray(int[] nums) {

        int maxSum = nums[0];

        int currentSum = nums[0];

        for (int i = 1; i < nums.length; i++) {

            currentSum = Math.max(nums[i], currentSum + nums[i]);

            maxSum = Math.max(maxSum, currentSum);

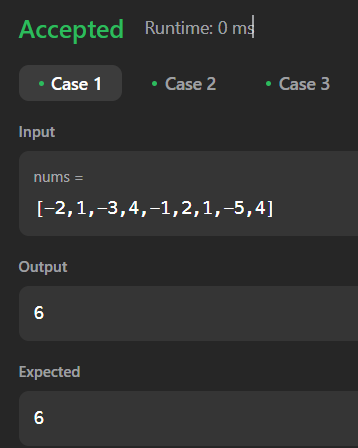
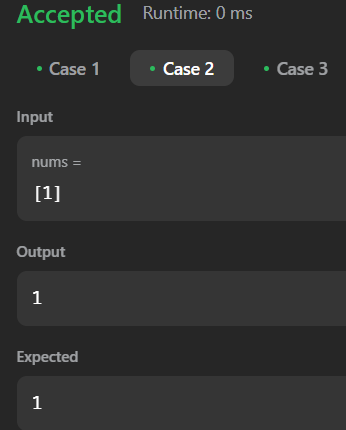
        }

        return maxSum;

    }

}

## Output

****

1. **Learning Outcome:**

 Understand dynamic programming to solve recurrence-based problems.

 Learn how to use an array to store intermediate results for optimization.

 Recognize the Fibonacci-like nature of the "climbing stairs" problem.

 Implement iterative solutions to reduce redundant calculations.

 Improve problem-solving skills with bottom-up dynamic programming