EXPERIMENT - 10

Student Name: Sandeep Kumar **UID: 20BCS4885**

Branch: CSE **Section/Group: 603/A**

Semester: 6th semester **Subject: Competitive Coding**

Aim: To demonstrate the concept of Dynamic Programming.

Objective:

a) Best time to buy and sell the stock

b) Climbing Stairs

Problem 1: Best time to buy and sell the stock

Solution code:

```
class Solution {
public:
    int maxProfit(vector<int>& prices) {
        int n = prices.size(), maxProfit = 0;
        for(int i=0; i<n-1; i++){
            for(int j=i+1; j<n; j++){
                if(prices[j] - prices[i] > maxProfit){
                    maxProfit = prices[j] - prices[i];
                }
            }
        }
        return maxProfit;
    }
};
```

Approach:

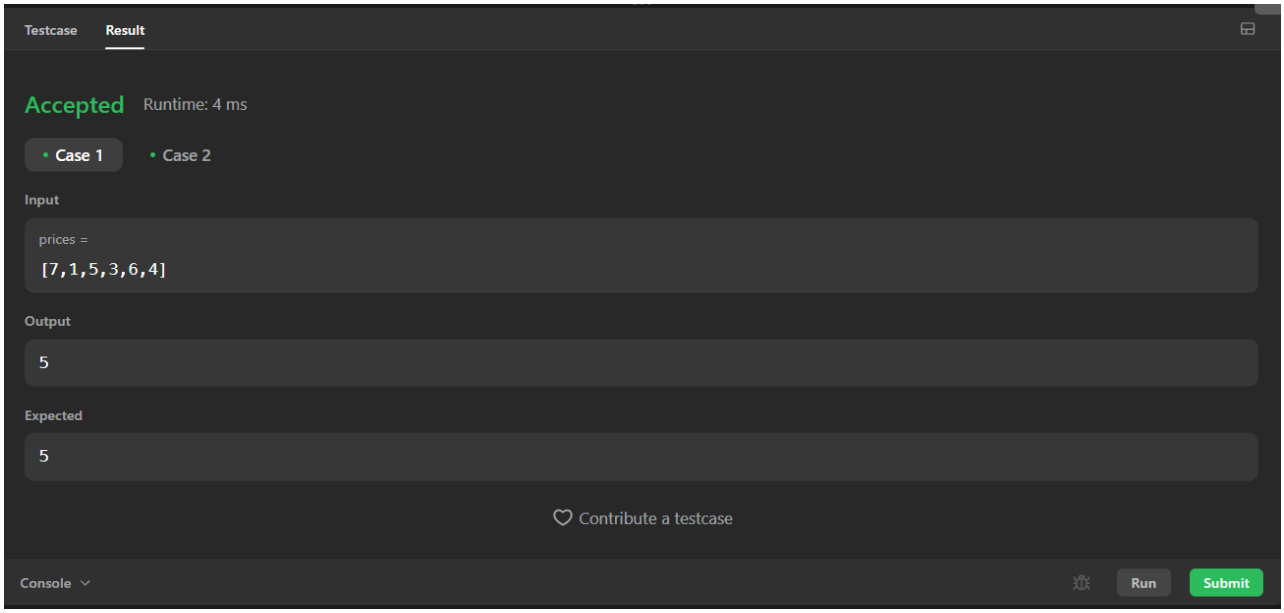
Solved using Array (Two Nested Loop). Brute Force Approach.

Complexity:

Time Complexity: O(N^2)

Space Complexity: O(1)

Output:



Problem 2: Climbing Stairs

Input Code:

```
class Solution {
public:
    int climbStairs(int n) {
        if(n<=2)
         return n;
        vector<int> dp(n+1);
        dp[0]=0;
        dp[1]=1;
        dp[2]=2;
        for(int i=3;i<=n;i++)
         dp[i]=dp[i-1]+dp[i-2];

        return dp[n];
    }
};
```

Approach:

- Using Bottom - up Approach -> Tabulation.

- Storing the values of overlapping sub - problems in a vector.

Complexity:

Time Complexity: O(n) -> As we are traversing the vector at least 1 time.

Space Complexity: O(n) - > (Array of size n)

Output: