Task 1: Fibonacci Sequence

Objective: Use dynamic programming to compute the Fibonacci sequence.

Instructions:

1. Implement a function that returns the nth Fibonacci number using dynamic programming.
2. Use memoization or tabulation to optimize the solution.

Expected Outcome:*The function should compute large Fibonacci numbers efficiently compared to a recursive approach.

Task 2: Minimum Coin Change

Objective: Solve the coin change problem using dynamic programming.

Instructions:

1. Given an array of coin denominations and a target amount, find the minimum number of coins required to make the target.
2. Implement the solution using a bottom-up dynamic programming approach.

Expected Outcome: The program should return the minimum number of coins required or indicate if the target cannot be reached.

Task 3: Longest Common Subsequence

Objective: Find the length of the longest common subsequence between two strings.

Instructions:

1. Write a function that takes two strings and finds the length of their longest common subsequence.
2. Use dynamic programming to optimize the solution.

Expected Outcome: The function should return the length of the longest common subsequence.

Task 4: Climbing Stairs

Objective: Solve the staircase problem using dynamic programming.

Instructions:

1. A person can climb either 1 or 2 steps at a time. Given the total number of stairs `n`, find how many distinct ways the person can climb to the top.
2. Use dynamic programming to avoid recalculating the number of ways for each step.

Expected Outcome: The program should return the number of distinct ways to climb `n` stairs.

Task 5: Knapsack Problem (0/1)

Objective: Solve the 0/1 knapsack problem using dynamic programming.

Instructions:

1. Given a set of items with values and weights, and a knapsack with a weight capacity, find the maximum value that can be obtained.
2. Use dynamic programming to determine the optimal solution.

Expected Outcome: The program should return the maximum value that can be achieved without exceeding the knapsack’s capacity.