DP Practice Problems

Problem 1: Climbing Stairs

You are climbing a stair case. It takes n steps to reach to the top. Each time you can either climb 1 or 2 steps. In how many distinct ways can you climb to the top? Note: Given n will be a positive integer.

Example:
Input: 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

1. A function F(i) is defined to denote the number of distinct ways you can climb to the top. Please provide F(1) and F(2). Then provide F(i) in terms of previously computed value of F.

Problem 2: Making Change

You are given k denominations of coins, $d_1, d_2, \ldots d_k$ (all integers). Assume $d_1 = 1$ so it is always possible to make change for any amount of money. We want to find an algorithm that makes change for an amount of money n using as few coins as possible.

1. Argue that the coin changing problem exhibits optimal substructure.
2. Let $C[p]$ be the minimum number of coins of the k denominations that sum to p cents. Recursively define the value of the optimal solution. To get you started, there must exist some "first coin" d_i, where $d_i \leq p$.
3. Provide an algorithm (e.g., pseudocode) to compute the value of the optimal solution bottom up. Provide another algorithm to construct the optimal solution from the computed information.
4. What is the running time of your algorithm?