

Programming Task: The Coin Problem

Part 1: Minimum Number of Coins

Have you ever paid with coins at a store and wondered how many you needed? This challenge is based on that idea. It's a classic problem in computer science called the **Coin Change Problem**.

The Task

You are given a list of coin denominations (like 1, 5, 10) and a total amount of money. Your goal is to figure out the minimum number of coins needed to make up that amount.

Important: You can use as many coins of each type as you want.

Input

- A list of coin values (positive integers, e.g., [1, 3, 4])
- A target amount (positive integer, e.g., 6)

Output

- The minimum number of coins needed to make the amount
- If it's not possible to make the amount with the given coins, return -1

Example

- **Input:** Coins = [1, 3, 4], Amount = 6
- **Output:** 2 (You can use 3 + 3 or 4 + 1 + 1)

Part 2: List All Possible Combinations

Instead of just the minimum, can you write a program that lists **all possible combinations** of coins that add up to the amount?

Example

- **Input:** Coins = [1, 2], Amount = 4

- **Output:**

– 1 + 1 + 1 + 1

– 1 + 1 + 2

– 2 + 2

Part 3: Limited Coins

Let's make it harder. Now, you can only use each coin type a limited number of times.

Task

You are given:

- A list of coin values, e.g., [1, 3, 4]
- A matching list of available counts for each coin, e.g., [2, 1, 1] (you have 2 ones, 1 three, 1 four)
- A target amount, e.g., 6

Can you still make the amount? If yes, what is the minimum number of coins used?

Bonus

Print the actual coins used. If it's not possible, output -1.

Example

- **Input:**

– Coins = [1, 3, 4]

– Counts = [2, 1, 1]

– Amount = 6

- **Output:** 1 + 1 + 4