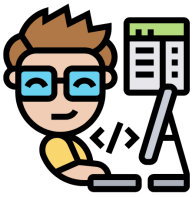

DAILY PROGRAMMING
CHALLENGE

WISDOM SPROUTS
IT Training Hub

The Coin Change Problem

You are given an integer array coins[] of size n, where each element represents the denomination of a coin. You are also given an integer amount, representing the total amount of money. The task is to find the minimum number of coins required to make up the given amount.
If the amount cannot be formed by any combination of the coins, return -1.
You can assume that you have an infinite supply of each denomination.

Input:

- An integer array coins[] where each element represents the value of a coin.
- An integer amount representing the total amount of money.

Output:

- Return the minimum number of coins needed to make up the amount.
- If the amount cannot be formed by any combination of coins, return -1.

Examples:

- Example 1
  Input: coins = [1, 2, 5], amount = 11
  Output: 3
  Explanation: To make the amount 11, the fewest number of coins is 3 (5 + 5 + 1).

Constraints:

- 1 ≤ n ≤ 12 (size of coins[])
- 0 ≤ amount ≤ 10^4
- Each coin's value will be a positive integer.

Test Cases:

1. Input: coins = [1, 2, 5], amount = 11
   Output: 3
2. Input: coins = [2], amount = 3
   Output: -1
3. Input: coins = [1], amount = 0

Output: 0

Edge Cases:

1. No coins are needed, so the result is 0.
2. If the denominations are too large to make up a small amount, return -1.
3. If no combination of coins can form the exact amount, return -1.