Beaumont Yin
CS 146

Task: You are given an integer array coins representing coins of different denominations and an integer amount representing a total amount of money.

Return the fewest number of coins that you need to make up that amount. If that amount of money cannot be made up by any combination of the coins, return -1.

You may assume that you have an infinite number of each kind of coin.

Approach:

1. Create an array and fill each element with a value greater than amount
2. Index 0 of the array is zero because you need zero coins to make amount zero
3. Iterate through 1 to amounts and for each amount try each denomination
4. If the current coin is less than the current amount, update the current number of coins needed for the current amount

Test cases:

```
int[] coins1 = {1, 2, 5};
        int amount1 = 11;
        System.out.println("Test Case 1:");
        System.out.println("Input: coins = " + Arrays.toString(coins1) + ", amount = " + amount1);
        System.out.println("Output: " + solution.coinChange(coins1, amount1));

        int[] coins2 = {2};
        int amount2 = 3;
        System.out.println("\nTest Case 2:");
        System.out.println("Input: coins = " + Arrays.toString(coins2) + ", amount = " + amount2);
        System.out.println("Output: " + solution.coinChange(coins2, amount2));
```