

# Coin Change

Try to solve the Coin Change problem.

## We'll cover the following ^

- Statement
- Examples
- Understand the problem
- Figure it out!
- Try it yourself

## Statement

You're given an integer `total` and a list of integers called `coins`. The variable `coins` hold a list of coin denominations, and `total` is the total amount of money.

You have to find the minimum number of coins that can make up the `total` amount by using any combination of the coins. If the amount can't be made up, return `-1`. If the total amount is `0`, return `0`.

**Note:** You may assume that we have an infinite number of each kind of coin.

### Constraints:

$$1 \leq \text{coins.length} \leq 12$$

$$1 \leq \text{coins}[i] \leq 2^{31} - 1$$

$$0 \leq \text{total} \leq 10^3$$

## Examples

### Sample example 1

#### Inputs

coins	1	3	4	5
-------	---	---	---	---

total	7
-------	---

#### Output

2 Denominations = 4 + 3

1 of 4

## Understand the problem

Let's take a moment to make sure you've correctly understood the problem. The quiz below helps you check if you're solving the correct problem:

Coin Change

1

What is the minimum number of coins required to make up the following total with the set of coins available to us?  
  
coins = [1, 2, 3, 4] total = 11

A) -1

B) 3

C) 11

Submit Answer

<


Question 1 of 2  
0 attempted

>

Reset Quiz ↻

Figure it out!

We have a game for you to play. Rearrange the logical building blocks to develop a clearer understanding of how to solve this problem.

 Drag and drop the cards to rearrange them in the correct sequence.

Initialize a counter array that contains elements equal to our total. Furthermore, initialize a variable to store the minimum number of coins needed. The **minimum** variable can be initialized to infinity at the start of each path.

Traverse the **coins** array, and for each element, check the base cases. If the remaining sum is equal to 0, return 0. If it is less than 0, return -1 and if it is greater than 0, return the target value stored at the -  $i^{th}$  index of the counter array.

?

T

⌚

Store this value into a separate variable called `result`.

Increment the value in `result` variable by one and add it to the `minimum` variable. Repeat this process until the  $(rem-1)^{th}$  index of the `counter` is not infinity.

Reset

Show Solution

Submit

## Try it yourself

Implement your solution in the following coding playground:

Java



```
2 public class CoinChange{
3     public static int coinChange(int [] coins, int total) {
4
5         // Write your code here
6
7         return -1;
8     }
9 }
```

Powered by AI



Submit

Test Cases Results

Case 1

Case 2

Case 3

Input #1

[1,2,5]

Input #2

11

Coin Change

← Back

Solution: 0/1 Knapsack

Next →

Solution: Coin Change

✓ Mark as Completed



