

Problem

A cashier has currency notes of denominations 1, 5, 10, 50 and 100. If the amount to be withdrawn is input through the keyboard, find the total number of currency notes of each denomination the cashier will have to give to the withdrawer making sure the cashier hands out as many less number of notes as possible

Problem Breakdown

- A cashier has currency notes of different denominations, namely 1, 5, 10, 50, and 100.
- The problem requires us to determine the number of notes of each denomination the cashier needs to give to a person who wants to withdraw money.
- The input to the problem is the amount that the person wants to withdraw, which will be entered through the keyboard.
- The solution needs to minimize the number of notes given to the person, i.e., the cashier should give the least number of notes of each denomination possible to fulfill the withdrawal amount.
- To solve the problem, we need to use a combination of the given denominations that add up to the withdrawal amount.
- For example, if the person wants to withdraw 150, the cashier could give one note of 100 and one note of 50, instead of giving three notes of 50 or fifteen notes of 10.
- We need to find the combination of notes that minimizes the total number of notes required.
- Once we have the combination of notes, we can count the number of notes of each denomination and output the result.

Solution to The Problem

1. Read the input: the amount to be withdrawn from the keyboard.
2. Initialize the number of notes of each denomination to zero.
3. Find the largest denomination that is smaller than or equal to the withdrawal amount.
4. Subtract the value of this denomination from the withdrawal amount and increment the number of notes of this denomination.
5. Repeat steps 3 and 4 until the withdrawal amount becomes zero.
6. Output the number of notes of each denomination.

Example

Let's go through an example to understand the solution better. Suppose the person wants to withdraw 276:

1. Read the input: 276.
2. Initialize the number of notes of each denomination to zero.
3. The largest denomination smaller than or equal to 276 is 100.
4. Subtract 100 from 276 and increment the number of 100 notes by 1. The remaining amount is 176.
5. The largest denomination smaller than or equal to 176 is 100.
6. Subtract 100 from 176 and increment the number of 100 notes by 1. The remaining amount is 76.
7. The largest denomination smaller than or equal to 76 is 50.
8. Subtract 50 from 76 and increment the number of 50 notes by 1. The remaining amount is 26.
9. The largest denomination smaller than or equal to 26 is 10.
10. Subtract 10 from 26 and increment the number of 10 notes by 1. The remaining amount is 16.
11. The largest denomination smaller than or equal to 16 is 10.
12. Subtract 10 from 16 and increment the number of 10 notes by 1. The remaining amount is 6.
13. The largest denomination smaller than or equal to 6 is 5.

14. Subtract 5 from 6 and increment the number of 5 notes by 1. The remaining amount is 1.
15. The largest denomination smaller than or equal to 1 is 1.
16. Subtract 1 from 1 and increment the number of 1 notes by 1. The remaining amount is 0.
17. Output the number of notes of each denomination: 100 notes: 2, 50 notes: 1, 10 notes: 2, 5 notes: 1, 1 notes: 1.

Pseudocode

```
// Step 1: Read the input
withdrawal_amount = read_input()

// Step 2: Initialize the number of notes of each denomination to zero
num_100_notes = 0
num_50_notes = 0
num_10_notes = 0
num_5_notes = 0
num_1_notes = 0

// Step 3: Find the largest denomination that is smaller than or equal to the withdrawal amount
while withdrawal_amount > 0:
    if withdrawal_amount ≥ 100:
        // Step 4: Subtract the value of this denomination from the withdrawal amount and increment the number of notes
        withdrawal_amount -= 100
        num_100_notes += 1
    elif withdrawal_amount ≥ 50:
        withdrawal_amount -= 50
        num_50_notes += 1
    elif withdrawal_amount ≥ 10:
        withdrawal_amount -= 10
        num_10_notes += 1
    elif withdrawal_amount ≥ 5:
        withdrawal_amount -= 5
        num_5_notes += 1
    else: // withdrawal_amount < 5
        withdrawal_amount -= 1
        num_1_notes += 1

// Step 6: Output the number of notes of each denomination
output(num_100_notes, num_50_notes, num_10_notes, num_5_notes, num_1_notes)
```