Max Coins in Box

Problem Statement

There are five boxes arranged from left to right. You keep adding a variable number of coins sequentially in each box. Start from box-1 and keep going right. Once you reach the last box, head back to box-1 and then keep adding coins. In any given turn, the number of coins added to a box is always less than 10.

Find the box which has the maximum number of coins. If there are two boxes which have the same maximum number of coins, output the smaller of the two box numbers. The sequence of coins is represented by a string. For example, if the input is 3972894910, this is how coins are added:

Box	Coins
1	3 + 9 = 12
2	9 + 4 = 13
3	7 + 9 = 16
4	2 + 1 = 3
5	8 + 0 = 8

In this case, 3 is the output as box-3 has the maximum number of coins in it.

Input Format:

The input consists of a single line containing a string representing the sequence of coins.

Output Format:

Output a single integer representing the box number that has the maximum number of coins. If there are two boxes with the same maximum number of coins, output the smaller of the two box numbers.

Example
Input:
123456
Output:
1

Solution

Solution in Java :

```
import java.util.HashMap;
import java.util.Scanner;

public class CoinBoxes {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        String coinsInput = scanner.nextLine();
        scanner.close();

        int[] boxes = new int[6];
        for (int i = 0; i < coinsInput.length(); i++) {
            int coinNumber = i + 1;
            int boxNumber = coinNumber % 5;
            boxNumber = (boxNumber == 0) ? 5 : boxNumber;
            boxes[boxNumber] += Integer.parseInt(Character.toString(coinsInput.charAt(i)));
        }

        int maxBox = -1;
        int maxCoins = -1;
        for (int boxNumber = 1; boxNumber <= 5; boxNumber++) {
            int coins = boxes[boxNumber];
            if (coins > maxCoins) {
                maxCoins = coins;
                maxBox = boxNumber;
            }
        }

        System.out.println(maxBox);
    }
}
```

Solution in python:

```
coins = input()
boxes = dict()
for i in range(1, 6):
    boxes[i] = 0
for i, coin in enumerate(coins):
    coin_number = i + 1
    box_number = coin_number % 5
    box_number = 5 if box_number == 0 else box_number
    boxes[box_number] += int(coin)

max_box, max_coins = -1, -1
for box_number in [1, 2, 3, 4, 5]:
    coins = boxes[box_number]
    if coins > max_coins:
        max_coins = coins
        max_box = box_number
print(max_box)
```

Test Cases

Test case ID	Input	Output	Weight	Comments
1	3972894910	3	20	Basic
2	123456	1	10	Basic
3	12345123451234512345	5	30	Interesting

4	1762746400800846	1	40	Interesting