Currency

Prescribe a dict which key is a price of coin or banknote and value is number of it, for example {100: 5, 50: 2, 10: 5, 1: 15} means 5 of 100-baht banknotes, 2 of 50-baht banknotes, 5 of 10-baht coins, 15 of 1-baht coins.

Your task is to implement these following functions

- **`total(pocket)`** returns the amount of money in **`pocket`**
- **`take(pocket, money)`** add money from **`money`** to **`pocket`** (both are dict which contains money)
- **`pay(pocket, amt)`** remove **`amt`** (integer) of money from p. The function will return money (dict) which is paid from this function. If money is not enough to pay **`amt`**, it will return empty dict.

  To pay money, this function must select the highest value ones first, for example, if there is {100: 5, 50: 2, 10: 5, 1: 15}, and you want to pay for 57 baht, a 50-baht banknote and 7 of 1-baht coins will be removed.

```
def total(pocket):


def take(pocket, money_in):


def pay(pocket, amt):


exec(input().strip())      # This command is necessary to grade your answer
```

Input

Python commands which are used to test the functions.

Output

Result from the system.

Example

Input (from keyboard)	Output (on screen)
`p={100:2, 50:2, 5:2, 1:2};print(total(p))`	`312`
`p={100:5};take(p,{100:2, 1:3});print(p)`	`{100: 7, 1: 3}`
`p={100:5};take(p,{100:0, 1:0});print(p)`	`{100: 5, 1: 0}`
`p={10:5, 1:7};print(pay(p, 12));print(p)`	`{10: 1, 1: 2}` `{10: 4, 1: 5}`
`p={10:5, 1:7};print(pay(p, 18));print(p)`	`{}` `{10: 5, 1: 7}` Not enough
`p={10:5, 1:7};print(pay(p, 100));print(p)`	`{}` `{10: 5, 1: 7}` Not enough
`p={10:5, 1:7};print(pay(p, 57));print(p)`	`{10: 5, 1: 7}` `{10: 0, 1: 0}` enough