2559_2_TupleSetDict_Q3

ขอแทนการเก็บเงินด้วย **dict** มี key เป็นมูลค่าของธนบัตรหรือเหรียญ และ value เป็นจำนวนนับของธนบัตรหรือเหรียญนั้น เช่น **{100:1, 20:2, 1:3}** หมายถึง ธนบัตร 100 บาท 1 ใบ ธนบัตร 20 บาท 2 ใบ และเหรียญ 1 บาท 3 เหรียญ รวมเป็นเงิน 143 บาท

โปรแกรมที่แสดงข้างล่างนี้รับอินพุต 2 บรรทัด บรรทัดแรกเป็นธนบัตรและเหรียญต่าง ๆ ที่มีอยู่ในกระเป๋า บรรทัดที่สองเป็นเงินที่ต้องจ่าย เช่น

```
100:1, 20:2, 1:3
21
```

คือมี 143 บาท ต้องจ่าย 21 บาท ก็จ่ายด้วยธนบัตร 20 บาท 1 ใบ และเหรียญบาทอีก 1 เหรียญ กระเป๋าก็เหลือ **{100:1, 20:1, 1:2}**

หลังจากอ่านข้อมูลแล้ว โปรแกรมนี้คำนวณว่ามีเงินในกระเป๋าพอจ่ายหรือไม่ ถ้าไม่พอก็จะพิมพ์ว่า **"Not enough money"**

ถ้ามีเงินพอ ก็คำนวณว่า ต้องจ่ายด้วยธนบัตรหรือเหรียญใด จำนวนกี่ใบกี่เหรียญบ้างที่มีอยู่ในกระเป๋า

แต่ถ้ามีเงินพอ แต่ไม่พอดี (คือถ้าจ่าย ต้องมีการทอนเงิน เช่น มี **{ 100:2 }** ต้องการจ่าย **53** บาท) ก็พิมพ์ว่า **"Not match"**

จงปรับปรุง

โปรแกรมข้างล่างนี้ได้เขียนส่วนอื่น ๆ มาให้ถูกต้องแล้ว เหลือเว้นไว้ 2 ส่วน (บริเวณสีดำ) ดังนี้

- ส่วนสีดำแรก ต้องคำนวณ **pocket_amount** ให้เท่ากับยอดเงินรวมในกระเป๋า
  เช่น กระเป๋า (**pocket_money**) มีเงิน **{100:1,20:2,1:3}** ก็คือได้เงินรวม (**pocket_amount**) เป็น **143** บาท
- ส่วนสีดำที่สอง ให้แก้ไขเงินในกระเป๋า (**pocket_money**) ตามเงินที่จ่ายไป (**pay_money**)
  เช่น กระเป๋ามี **{100:1,20:2,1:3}** ต้องจ่าย **{20:1,1:1}** กระเป๋าก็ต้องเหลือ **{100:1,20:1,1:2}**
  อาจารย์ได้เขียนส่วนที่คำนวณจำนวนธนบัตรและเหรียญไว้ถูกต้องแล้ว รับประกันว่า **pay_money** จะสามารถจ่ายได้จริง

หมายเหตุ 1. สามารถดาวน์โหลดไฟล์เริ่มต้นได้ที่ http://pioneer.netserv.chula.ac.th/~psomchai/change.py

2. โปรแกรมข้างล่างนี้มีการ **def** ฟังก์ชัน แต่ไม่ต้องห่วง เพราะฟังก์ชันที่เขียน และบรรทัดที่ใช้ฟังก์ชันนี้ เขียนถูกต้องอยู่แล้ว

```
def print_money(s, m) :
    sorted_m = sorted([ (v, m[v]) for v in m ])
    sorted_m = sorted_m[::-1] # reverse order --> max to min
    print(s,','.join([str(v)+":"+str(c) for v,c in sorted_m if c>0]))
#-------------------------------------------
money = input().strip().split(",")
pocket_money = dict()
for item in money :
    v, c = [int(e) for e in item.split(":")]
    pocket_money[v] = c

print_money("In pocket:", pocket_money)
#-------------------------------------------
# หา ยอดเงินรวมใน pocket_money

pocket_amount = 1000000  # this 1000000 is WRONG !

   ???

#-------------------------------------------
pay_amount = int(input())
if pay_amount > pocket_amount :
    print("Not enough money")
else :
    # เริ่มคำนวณวิธีการจ่ายเงิน ว่าต้องจ่ายธนบัตรหรือเหรียญใด กี่ใบกี่เหรียญ
    pay_money = dict()
    sorted_money = sorted([ (v, pocket_money[v]) for v in pocket_money ])
    sorted_money = sorted_money[::-1] # reverse order --> max to min
    p = pay_amount
    for value,count in sorted_money :
        if value*count > p:
            c = p // value
        else :
            c = count
```

```
        p -= value*c
        pay_money[value] = c
    if p != 0 :
        print("Not match")
    else :
        print_money("Pay:", pay_money)
        #-------------------------------------
        # แก้ไขข้อมูลใน pocket_money เมื่อต้องจ่ายเงินตามที่คำนวณไว้ใน pay_money
        ???
        #-------------------------------------
        print_money("Left in pocket:", pocket_money)
```

ข้อมูลนำเข้า

บรรทัดแรกเป็นจำนวนธนบัตรและเหรียญต่าง ๆ ที่มีอยู่ในกระเป๋า ในรูปแบบ value:count, value:count, ... โดย value เป็นมูลค่าของธนบัตรหรือเหรียญ และ count เป็นจำนวนนับของธนบัตรหรือเหรียญนั้น ทั้งหมดเป็นจำนวนเต็ม

บรรทัดที่สองเป็นจำนวนเงินที่ต้องจ่าย เป็นจำนวนเต็ม

ข้อมูลส่งออก

กรณีที่มีเงินพอจ่ายได้พอดี ให้แสดง เงินในกระเป๋า เงินที่ต้องจ่าย เงินในกระเป๋าที่เหลือหลังจ่าย

กรณีที่มีเงินไม่พอจ่าย ให้แสดง เงินในกระเป๋า และคำว่า Not enough money

กรณีที่มีเงินพอจ่าย แต่มีธนบัตรและเหรียญไม่พอดีกับจำนวนเงินที่ต้องจ่าย ให้แสดง เงินในกระเป๋า และคำว่า Not match

ตัวอย่าง

input (ทางแป้นพิมพ์)	output (ทางจอภาพ)
10:2, 1000:1, 50:3, 20:4, 500:1 , 100:4 , 5:2, 1:3 1758	In pocket: 1000:1,500:1,100:4,50:3,20:4,10:2,5:2,1:3 Pay: 1000:1,500:1,100:2,50:1,5:1,1:3 Left in pocket: 100:2,50:2,20:4,10:2,5:1
10:2, 1000:1, 50:3, 20:4, 500:1 , 100:4 , 5:2, 1:3 1759	In pocket: 1000:1,500:1,100:4,50:3,20:4,10:2,5:2,1:3 Not match
10:2, 1000:1, 50:3, 20:4, 500:1 , 100:4 , 5:2, 1:3 10000	In pocket: 1000:1,500:1,100:4,50:3,20:4,10:2,5:2,1:3 Not enough money