

Programming Fundamentals I Lab.

8. Programming Lab Test A.

ชื่อ _____ รหัสนิสิต _____

ให้คุณเขียนฟังก์ชันตามที่โจทย์สั่งทุกข้อ รวมอยู่ในไฟล์เดียวกัน ตั้งชื่อไฟล์ว่า 583030xxxx.py โดยใส่ข้อมูลต่อไปนี้เป็น comment ที่หัวไฟล์

```
# ID: 583030xxxx
# Name: __First_Name__ __Last_Name__
# Test: A
```

1. จงเขียนฟังก์ชันชื่อ `sum_money()` ที่รับพารามิเตอร์เพียงตัวเดียว เป็น tuple (a, b, c, d, e, f, g) โดย a, b, ..., g เป็นจำนวนเต็มไม่ติดลบ แทนจำนวนของเหรียญบาท, เหรียญห้าบาท, เหรียญสิบบาท, ธนบัตรห้าสิบบาท, ธนบัตรหนึ่งร้อยบาท, ธนบัตรห้าร้อยบาท, และธนบัตรหนึ่งพันบาท ตามลำดับ และให้ฟังก์ชันคืนค่าเป็นจำนวนเงินรวมทั้งหมด (3 คะแนน)

ตัวอย่างการใช้งานเช่น

```
>>> sum_money((3, 5, 0, 7, 0, 1, 0))
878
```

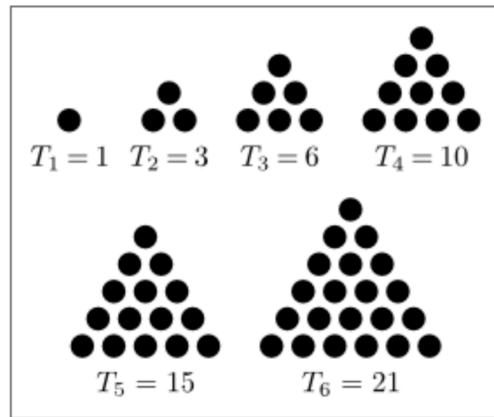
2. จงเขียนฟังก์ชันชื่อ `calculate_balance()` ที่รับพารามิเตอร์เพียงตัวเดียว เป็น list ของรายการฝาก-ถอนเงินในบัญชีหนึ่ง โดยแต่ละรายการฝากจะอยู่ในรูปของ tuple ('D', m) และรายการถอนจะอยู่ในรูป ('W', m) เมื่อ m เป็นจำนวนเงินที่ต้องการฝาก-ถอน ให้ฟังก์ชันนี้ทำการคำนวณว่า หากเริ่มต้นมีเงินในบัญชี 0 บาท และทำรายการฝาก-ถอนตามลำดับใน list สุดท้ายจะเหลือเงินในบัญชีเท่าไร ให้ฟังก์ชันคืนค่าเป็นจำนวนเงินที่เหลือในบัญชี โดยหากรายการถอนครั้งใดต้องการถอนเงินมากกว่าที่มีอยู่ในบัญชี ให้รายการนั้นเป็นโมฆะ นั่นคือรายการถอนครั้งนั้นไม่ถูกนำมาคำนวณด้วย (4 คะแนน)

ตัวอย่างการใช้งานเช่น

```
>>> calculate_balance([('D', 30000), ('W', 400), ('W', 20000),
                        ('W', 15000), ('D', 3500)])
13100
```

อธิบายตัวอย่าง: เริ่มมีเงินในบัญชี 0 บาท รายการแรกเป็นการฝากเงินดังนั้นทำให้มีเงินอยู่ในบัญชี 30,000 บาท ถัดมาเป็นรายการถอนจำนวน 400 บาท เหลือเงินในบัญชี 29,600 บาท ถัดมาเป็นรายการถอนจำนวน 20,000 บาท เหลือเงิน 9,600 บาท ถัดมาเป็นรายการถอนจำนวน 15,000 บาทซึ่งไม่สามารถทำได้ ดังนั้นจึงยังเหลือเงิน 9,600 บาท สุดท้ายเป็นรายการฝากเพิ่ม 3,500 บาท จึงมีเงินอยู่ในบัญชีเท่ากับ 13,100 บาท

3. Triangular number คือจำนวนเต็มที่แทนจำนวนสิ่งของที่สามเหลี่ยมด้านเท่าได้ ภาพต่อไปนี้แสดงตัวอย่าง triangular number 6 ตัวแรก ได้แก่ 1, 3, 6, 10, 15, และ 21



สังเกตว่าเราสามารถคำนวณหา triangular number ตัวที่ n ได้จาก $1 + 2 + \dots + n$

จงเขียนฟังก์ชันชื่อ `is_triangular_number()` ที่รับพารามิเตอร์เป็นจำนวนเต็มบวกหนึ่งตัว และให้ฟังก์ชันคืนค่าเป็น True ถ้าจำนวนเต็มตัวนั้นเป็น triangular number หากไม่ใช่ให้คืนค่าเป็น False (3 คะแนน)

ตัวอย่างการใช้งานเช่น

```
>>> is_triangular_number(276)
True
>>> is_triangular_number(100)
False
```