

# Programming Fundamentals I Lab.

## 8. Programming Lab Test B.

ชื่อ \_\_\_\_\_ รหัสนิสิต \_\_\_\_\_

ให้คุณเขียนฟังก์ชันตามที่โจทย์สั่งทุกข้อ รวมอยู่ในไฟล์เดียวกัน ตั้งชื่อไฟล์ว่า 583030xxxx.py โดยใส่ข้อมูลต่อไปนี้เป็น comment ที่หัวไฟล์

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

1. งานแข่งขันหนึ่ง กำหนดอายุผู้เข้าแข่งขันโดยพิจารณาจากวันเกิด ผู้ที่จะเข้าแข่งขันได้จะต้องเกิดในวันที่ 15 มิถุนายน 2530 หรือหลังจากนั้น ให้คุณเขียนฟังก์ชันชื่อ `pass_birthday_rule()` ที่รับวันเกิดเป็น tuple (d, m, y) โดย d เป็นเลขวัน m เป็นเลขเดือน และ y เป็นเลขปีพ.ศ.เกิด และให้ฟังก์ชันคืนค่าเป็น True หากสามารถเข้าแข่งขันได้ ไม่เช่นนั้นให้คืนค่าเป็น False (3 คะแนน)

ตัวอย่างการใช้งานเช่น

```
>>> pass_birthday_rule((10, 6, 2530))
False
>>> pass_birthday_rule((19, 6, 2533))
True
```

2. จงเขียนฟังก์ชันชื่อ `sum_money()` ที่รับพารามิเตอร์เพียงตัวเดียว เป็น list ของ tuple (t, n) โดย t แทนมูลค่าของเหรียญหรือธนบัตร มีค่าที่เป็นไปได้คือ 1, 2, 5, 10, 20, 50, 100, 500, และ 1000 เท่านั้น และ n แทนจำนวนเหรียญหรือธนบัตรนั้น และให้ฟังก์ชันคืนค่าเป็นจำนวนเงินรวมทั้งหมด หาก tuple ใดมีค่า t ไม่ตรงกับค่าที่เป็นไปได้ ไม่ต้องนำ tuple นั้นมาคิด (4 คะแนน)

ตัวอย่างการใช้งานเช่น

```
>>> sum_money([(5, 3), (100, 1), (20, 2), (22, 8)])
155
```

อธิบายตัวอย่าง: ในตัวอย่าง มีเหรียญห้าบาท 3 เหรียญ ธนบัตรใบละ 100 บาท 1 ใบ และธนบัตรใบละ 20 บาท 2 ใบ รวมเป็นเงิน 155 บาท (tuple สุดท้ายมีค่า `t` เป็น 22 ไม่ตรงกับที่โจทย์กำหนด ดังนั้นจึงไม่นำมาคิดรวม)

3. จำนวนเฉพาะคือจำนวนเต็มบวกที่มากกว่า 1 ซึ่งไม่สามารถแยกตัวประกอบได้อีก นั่นคือจะมีเพียงเลข 1 และตัวมันเองเท่านั้นที่หารมันลงตัว จำนวนเฉพาะ 5 ตัวแรกได้แก่ 2, 3, 5, 7, และ 11 จงเขียนฟังก์ชันชื่อ `prime()` ที่รับพารามิเตอร์เป็นจำนวนเต็มบวก `n` หนึ่งตัว และคืนค่าเป็นจำนวนเฉพาะตัวที่ `n` (3 คะแนน)

ตัวอย่างการใช้งานเช่น

```
>>> prime(1)
2
>>> prime(5)
11
```