



Lab	
HW	
Until	

การบ้านปฏิบัติการ 10

Iteration Part I (20 คะแนน)

ข้อกำหนด

- การเรียกใช้ฟังก์ชันเพื่อการทดสอบ ต้องอยู่ภายใต้เงื่อนไข `if __name__ == '__main__':` เพื่อให้สามารถ import ไปเรียกใช้งานจาก Script อื่น ๆ ได้อย่างเป็นมาตรฐาน
 - ไม่อนุญาตให้ใช้ Data Type อื่น ๆ ที่ยังไม่สอนในบทเรียน เช่น **set** หรือ **dict** ในการแก้ปัญหา
 - สามารถเลือกแก้ปัญหาด้วยการใช้ Iteration (**for**, **while**) , Recursion หรือ Functional Control (**map**, **reduce**, **filter**, and **lambda**) ตามความเหมาะสม
- 1) 4 คะแนน (Lab10_1_6XXXXXXX.py) ให้เขียนฟังก์ชัน `comma_separated(n: int, group: int=3)` เพื่อคืนค่า String แทนจำนวนเต็มบวก n ในรูป comma separated (คั่นด้วยเครื่องหมาย comma) ตาม จำนวนเต็มบวก group ที่ใช้ระบุจำนวนหลักระหว่างเครื่องหมาย comma และจะต้องเริ่มนับจำนวนหลักจากหลักหน่วย ทั้งนี้ กำหนดให้ group เป็น Default Argument ที่มีค่าเป็น 3

<u>Input</u>	<u>Output</u>
3400 3	3,400
3400 4	3400
781588 5	7,81588
1234	1,234
1000000	1,000,000

- 2) 4 คะแนน (Lab10_2_6XXXXXXX.py) ให้เขียนฟังก์ชัน `longest_digit_run(n: int) -> int` เพื่อคืนค่าความยาวหลักของตัวเลขที่ปรากฏติดกันมากที่สุดในจำนวนเต็มบวก n เช่นเมื่อพิจารณา 11777332 จะเห็นว่าตัวเลขที่ติดกันยาวที่สุดคือ 7 เป็นจำนวน 3 หลัก

<u>Input</u>	<u>Output</u>
11777332	3
1177332	2

- 3) 4 คะแนน (HW10_2_6XXXXXXX.py) ให้เขียนฟังก์ชัน `eratosthenes(n: int, show_step: bool=False) -> list[int]` เพื่อคืนค่า List ของจำนวนเฉพาะตั้งแต่ 2 ถึง จำนวนเต็มบวก n ตามวิธี 'Sieve of

Eratosthenes' ในบทเรียนโดยมี Optional Parameter *show_step* เพื่อแสดง/ไม่แสดงขั้นตอนในแต่ละ Iteration ดังแสดงด้านล่าง

Function Call	Output
<pre>result = eratosthenes(20, True) print('----') print(result)</pre>	<pre>2: [2, 3, 5, 7, 9, 11, 13, 15, 17, 19] 3: [2, 3, 5, 7, 11, 13, 17, 19] ---- [2, 3, 5, 7, 11, 13, 17, 19]</pre>
<pre>result = eratosthenes(20) print('----') print(result)</pre>	<pre>---- [2, 3, 5, 7, 11, 13, 17, 19]</pre>

- 4) 4 คะแนน (HW10_1_6XXXXXXX.py) ท่ามกลางพายุฝนโหมกระหน่ำ ขบวนรถบัสจากบริษัทแดงและบริษัทส้ม สองบริษัทคู่แข่งที่บาดหมางกันมานานจากอุดมคติทางการเมืองที่แตกต่าง กำลังพานักท่องเที่ยวต่างชาติไปพัก ณ โรงแรมชื่อดังบนอุทยานแห่งชาติเขาใหญ่ที่เป็นข่าวดังเมื่อไม่นานมานี้ ทางขึ้นโรงแรมเป็นถนน One way สองเลน รถบัสทั้งหมดของบริษัทแดงวิ่งอยู่ทางเลนซ้าย รถบัสทั้งหมดของบริษัทส้มวิ่งอยู่ทางเลนขวา ทีมคนขับรถทั้งสองบริษัทต่างเหยียบคันเร่งสุดกำลังหวังจะไปถึงที่หมายก่อน โชคร้ายที่ก่อนถึงโรงแรมประมาณ 1 กิโลเมตร ทางเลนขวามีต้นไม้หักโค่นลงมาทำให้ต้องเดินรถได้เพียงทางเดียว เพื่อความปลอดภัย คนขับรถจากทั้งสองบริษัทจึงตกลงร่วมกันด้วย MOU ว่าจะไม่มีการหักหลังกัน จะไม่มีการขับถอยหลัง และรถที่มาจากเลนเดียวกันจะไม่แซงกัน



หน้าที่ของคุณคือให้เขียนฟังก์ชัน `arrival_sequences(l_lane: tuple[str], r_lane: tuple[str]) -> list[str]` เพื่อคืนค่า List ของ String แทนลำดับที่เป็นไปได้ทั้งหมดที่รถบัสของทั้งสองบริษัทจะถึงโรงแรม โดย `l_lane` เป็น Tuple ของ String แทนเลขประจำรถจากบริษัทแดงที่วิ่งทางซ้าย และ `r_lane` เป็น Tuple ของ String แทนเลขประจำรถจากบริษัทส้มที่วิ่งทางขวา และกำหนดให้แต่ละบริษัทมีรถในขบวนอย่างน้อยหนึ่งคัน โดย String ของลำดับที่เป็น Output จะอยู่ในรูป เลขประจำรถคันด้วย '>' (เครื่องหมายมากกว่า) ทั้งนี้ String ใน List ที่คืนค่าจะอยู่ในลำดับใดก็ได้

Input	Output
<code>arrival_sequences(('R32', '09'), ('05'))</code>	<pre>['09>05>R32', '09>R32>05', 'R32>09>05']</pre>
<code>arrival_sequences(('R2', 'R4'), ('034', '022'))</code>	<pre>['R2>R4>034>022', 'R2>034>R4>022', 'R2>034>022>R4', '034>R2>R4>022', '034>R2>022>R4', '034>022>R2>R4']</pre>

5) 4 คะแนน (HW10_3_6XXXXXXXXX.py) ให้เขียนฟังก์ชัน

```
polynomial_addition(p1: list[tuple[int, float]],
                    p2: list[tuple[int, float]]) -> list[tuple[int, float]]
```

เพื่อคืนค่าผลลัพธ์ของการบวกพหุนามตัวแปรเดียว 2 จำนวนตัวอย่างเช่น

$$a = 6x^2 + 34x - 8$$

$$b = -6x^2 + x + 2$$

จะได้ผลลัพธ์ $a + b = 35x - 6$ หรือ $[(1, 35), (0, -6)]$ โดย $p1$ และ $p2$ จะเป็น List ของ Tuple ที่อยู่ในรูป (กำลัง, สัมประสิทธิ์) หรือ (power, coefficient) โดย power จะเป็นจำนวนเต็ม ($power \geq 0$) ในขณะที่ coefficient เป็นจำนวนจริงใดๆ ทั้งนี้ฟังก์ชันจะต้องทำงานแบบ Non-destructive และคืนค่า List ผลลัพธ์เรียงตามกำลังจากมากไปน้อย

Hint:

- พิจารณาการใช้ฟังก์ชัน sorted() เพื่อเรียงลำดับ tuple ภายใน List ก่อนทำการบวก
- พิจารณาการใช้ Merge Algorithm ในการแก้ปัญหา

Input

Output

[(2, 6), (1, 34), (0, -8)]	[(1, 35), (0, -6)]
[(2, -6), (0, 2), (1, 1)]	

การส่งงาน

1. ลักษณะ/ลำดับข้อความของการรับค่า/แสดงผล จะต้องเป็นไปตามที่ระบุในตัวอย่างการ run
2. ไฟล์งานที่ส่ง จะต้องมีการแทรก comment ที่ต้นไฟล์ตามข้อกำหนดใน canvas รายวิชา
3. ไฟล์งานโปรแกรมที่ส่ง จะต้องมีการแทรก pseudocode เป็น comment ในแต่ละขั้นตอน
4. Upload ไฟล์ source code ตามที่ระบุในแต่ละข้อ ไปยังระบบตรวจให้คะแนนอัตโนมัติ <https://cmu.to/gdr111>

COMPUTER SCIENCE

Chiang Mai University