



Lab	
HW	
Until	

## การบ้านปฏิบัติการ 9

## Recursion Part II (20 คะแนน)

ข้อกำหนด

- การเรียกใช้ฟังก์ชันเพื่อการทดสอบ ต้องอยู่ภายใต้เงื่อนไข `if __name__ == '__main__':` เพื่อให้สามารถ `import` ไปเรียกใช้งานจาก Script อื่น ๆ ได้อย่างเป็นมาตรฐาน
- ไม่อนุญาตให้ใช้การทำซ้ำแบบ Iterations เช่น `for`, `while`, หรือ Data Type อื่น ๆ ที่ยังไม่สอนในบทเรียน เช่น `set` หรือ `dict` ในการแก้ปัญหา
- ทุกข้อต้องมีการสร้างฟังก์ชัน `my_id()` โดยให้คืนค่าสายอักขระแทนเลขประจำตัวนักศึกษา 9 หลัก

- 4 คะแนน (Lab09\_1\_6XXXXXXX.py) ให้เขียนฟังก์ชัน `recursive patterned_message(message, pattern)` เพื่อแสดงผลข้อความ (`message`) ตามรูปแบบ (`pattern`) ที่ระบุ โดยรูปแบบจะประกอบด้วยอักขระ '\*' และอักขระว่าง (' ') และอักขระ new line ('\n') เรียงกันโดยโปรแกรมต้องแทนที่อักขระ '\*' ด้วย อักขระจาก `message` ที่ไม่ใช่ อักขระว่าง

**Hint** - ไม่จำเป็นต้องสร้าง String ใหม่เพื่อการแสดงผล

Function CallOutput

<code>patterned_message("123", "*** ** * ** **")</code>	12 312 31 23 1
<code>patterned_message("D and C", ''' ***** ***** ***** ''')</code>	DandCDandCDandC DandCD andCDa ndCDandCDandCDa
<code>patterned_message("Three Diamonds!", ''' * * * *** *** ***** *** *** * * * ''')</code>	T h r eeD iam ond s!Thr eeDia monds !Th ree Dia m o n

- การวิเคราะห์ปัญหา

- Input: จำนวนข้อมูล \_\_\_\_\_ ชนิดข้อมูล \_\_\_\_\_
- Output: (แสดงค่า) จำนวนข้อมูล \_\_\_\_\_ ชนิดข้อมูล \_\_\_\_\_  
(คืนค่า) จำนวนข้อมูล \_\_\_\_\_ ชนิดข้อมูล \_\_\_\_\_

2) 4 คะแนน (Lab09\_2\_6XXXXXXX.py) ให้เขียนฟังก์ชัน `recursive life_path(n)` เพื่อคืนค่า Life Path Number ของจำนวนเต็มบวก  $n$  ใด ๆ ด้วยวิธีจากหนังสือโหราศาสตร์ของพละ

พละได้รับมรดกเป็นหนังสือโหราศาสตร์จากคุณยาย โดยมีศาสตร์ลับการดูดวงชะตาวิธีพิเศษชื่อว่า Life Path Number วิธีนี้ต้องอาศัยการนำวันเดือนปีเกิดมาบวกกัน จนกว่าจะได้เลขหลักเดียว เช่น หากเกิดวันที่ 13 กันยายนปี 2004 จะได้ผลบวก

$$13\ 09\ 2004 \rightarrow (1 + 3 + 0 + 9 + 2 + 0 + 0 + 4) = (19) \rightarrow (1 + 9) = (10) \rightarrow (1 + 0) = (1)$$

โดยการคำนวณแบบนี้สามารถนำไปใช้กับตัวเลขอื่น ๆ ไม่จำกัดจำนวนหลัก เช่น เลขตัวรถประจำทาง หรือเลขสมุดบัญชีอีกด้วย

Input	Output
13092004	1
7	7
35	8

• การวิเคราะห์ปัญหา

- Input: จำนวนข้อมูล \_\_\_\_\_ ชนิดข้อมูล \_\_\_\_\_
- Output: (แสดงค่า) จำนวนข้อมูล \_\_\_\_\_ ชนิดข้อมูล \_\_\_\_\_  
(คืนค่า) จำนวนข้อมูล \_\_\_\_\_ ชนิดข้อมูล \_\_\_\_\_

3) 4 คะแนน (HW09\_1\_6XXXXXXX.py) ให้เขียนฟังก์ชัน `recursive left_max(list_a)` เพื่อคืนค่าผลลัพธ์เมื่อทำ `list_a` ที่เป็น list ของ จำนวนเต็มที่ไม่เป็นลบ ให้อยู่ในรูป left max กล่าวคือ เมื่อพิจารณาจากซ้ายไปขวา สมาชิกในแต่ละ index จะถูกแทนที่ด้วยจำนวนที่สูงที่สุด เมื่อเปรียบเทียบค่าของสมาชิกใน index นั้น ๆ และ สมาชิกใน index ทั้งหมดทางด้านซ้ายของ index นั้น เช่น `left_max([2, 8, 1])` จะมีผลลัพธ์เป็น `[2, 8, 8]` เนื่องจากสมาชิกใน index ที่ 1 จะต้องถูกแทนที่ด้วยตัวเลขที่สูงที่สุดระหว่าง 2 และ 8 สมาชิกใน index ที่ 2 จะต้องถูกแทนที่ด้วยตัวเลขที่สูงที่สุด ระหว่าง 2, 8 และ 1

Input	Output
[2, 8, 1]	[2, 8, 8]
[3, 3, 1, 1, 2, 4]	[3, 3, 3, 3, 3, 4]

• การวิเคราะห์ปัญหา

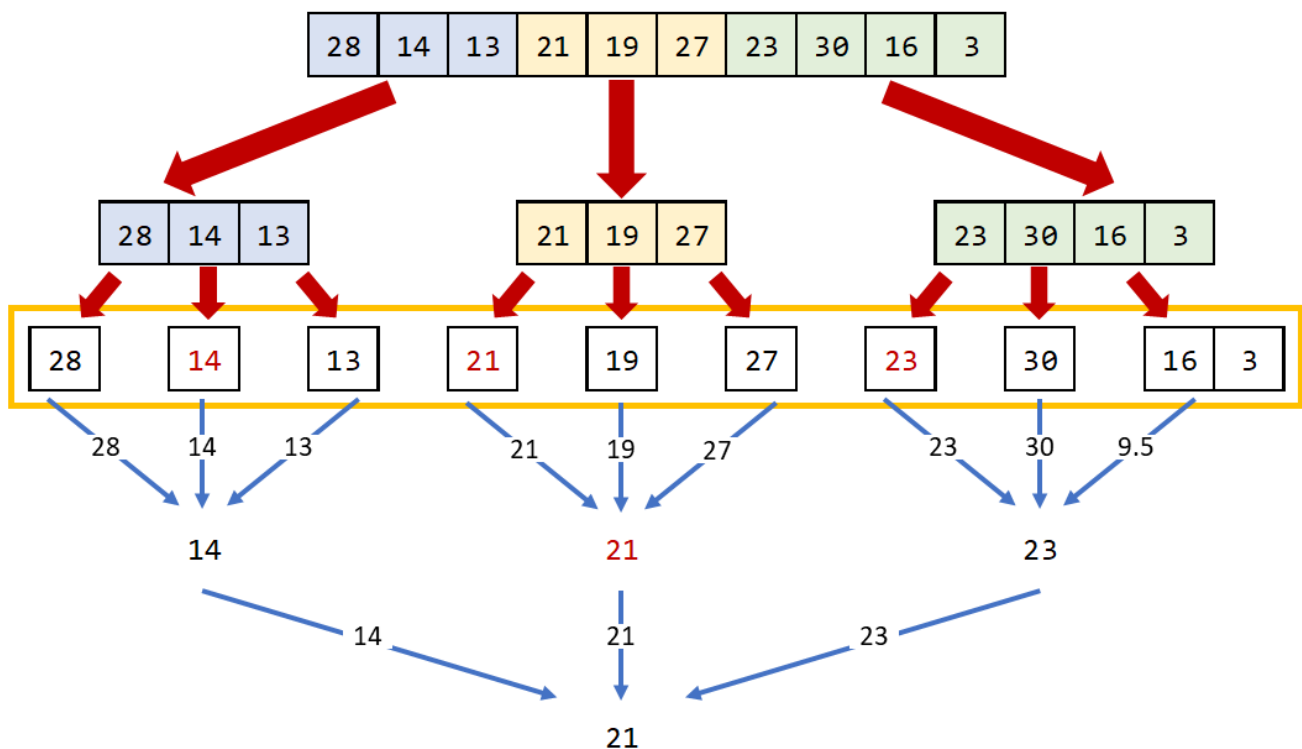
- Input: จำนวนข้อมูล \_\_\_\_\_ ชนิดข้อมูล \_\_\_\_\_
- Output: (แสดงค่า) จำนวนข้อมูล \_\_\_\_\_ ชนิดข้อมูล \_\_\_\_\_  
(คืนค่า) จำนวนข้อมูล \_\_\_\_\_ ชนิดข้อมูล \_\_\_\_\_

- 4) 4 คะแนน (HW09\_2\_6XXXXXXX.py) ให้เขียนฟังก์ชัน `median_of_median(list_a)` เพื่อคืนค่ากลางของ element ใน `list_a` ซึ่งเป็น List ของจำนวนจริง (`list_a` ไม่เป็น List ว่าง) โดยคำนวณค่ากลางดังกล่าวด้วยวิธี Median of Median

การคำนวณ Median of Median เป็นการคำนวณค่ากลางแบบ recursive โดยมีเงื่อนไขคือ

1. หาก `list_a` มีความยาวตั้งแต่ 3 ขึ้นไป จะต้องแบ่ง `list_a` เป็น List ย่อย 3 List เท่าๆ กัน
  - หากมีเศษจากการแบ่ง ให้นำไว้ List สุดท้ายเสมอ เช่น ถ้า `list_a` ยาว 40 จะต้องได้ความยาวทั้งสาม List ย่อยเป็น 13 13 และ 14
2. แบ่งย่อยลงไปจนเหลือ List ที่มีความยาว 1 หรือ 2
3. จากนั้นให้คำนวณค่ามัธยฐานในแต่ละ List ย่อยก่อน แล้วจึงนำค่าที่ได้มาหาค่ามัธยฐานอีกที
  - โดยกำหนดให้ค่ามัธยฐานของ List ที่มีความยาว 2 element คือค่าเฉลี่ยของทั้งสอง element นั้น ๆ ในรูปด้านล่างจะเห็นได้ว่าค่าเฉลี่ยของ 16 และ 3 คือ 9.5

ข้อกำหนด: ไม่อนุญาตให้ใช้ฟังก์ชัน `sorted()` ในการแก้ปัญหา



Input

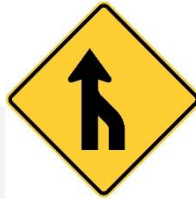
Output

[28, 14, 13, 21, 19, 27, 23, 30, 16, 3]	21.0
---	------

- การวิเคราะห์ปัญหา

- Input: จำนวนข้อมูล \_\_\_\_\_ ชนิดข้อมูล \_\_\_\_\_
- Output: (แสดงค่า) จำนวนข้อมูล \_\_\_\_\_ ชนิดข้อมูล \_\_\_\_\_  
(คืนค่า) จำนวนข้อมูล \_\_\_\_\_ ชนิดข้อมูล \_\_\_\_\_

- 5) 4 คะแนน (HW09\_3\_6XXXXXXX.py) ท่ามกลางพายุฝนโหมกระหน่ำ ขบวนรถบัสจากบริษัทแดงและบริษัทส้ม สองบริษัทคู่แข่งที่บาดหมางกันมานานจากอุดมคติทางการเมืองที่แตกต่าง กำลังพานักท่องเที่ยวต่างชาติไปพัก ณ โรงแรมชื่อดังบนอุทยานแห่งชาติเขาใหญ่ที่เป็นข่าวดังเมื่อไม่นานมานี้ ทางขึ้นโรงแรมเป็นถนน One way สองเลน รถบัสทั้งหมดของบริษัทแดงวิ่งอยู่ทางเลนซ้าย รถบัสทั้งหมดของบริษัทส้มวิ่งอยู่ทางเลนขวา ที่มีคนขับรถทั้งสองบริษัทต่างเหยียบคันเร่งสุดกำลังหวังจะไปถึงที่หมายก่อน โชคร้ายที่ก่อนถึงโรงแรมประมาณ 1 กิโลเมตร ทางเลนขวามีต้นไม้หักโค่นลงมาทำให้ต้องเดินรถได้เพียงทางเดียว เพื่อความปลอดภัย คนขับรถจากทั้งสองบริษัทจึงตกลงร่วมกันว่า จะไม่มีการขับถอยหลัง และรถที่มาจากเลนเดียวกันจะไม่แซงกัน



ให้เขียนฟังก์ชัน `arrival_sequences(left_lane, right_lane)` เพื่อคืนค่า List ของ String แทนลำดับที่เป็นไปได้ทั้งหมดที่รถบัสของทั้งสองบริษัทจะถึงโรงแรม โดย `left_lane` เป็น Tuple ของ String แทนเลขประจำรถจากบริษัทแดง และ `right_lane` เป็น Tuple ของ String แทนเลขประจำรถจากบริษัทส้ม กำหนดให้แต่ละบริษัทมีรถในขบวนอย่างน้อยหนึ่งคัน โดย String ของลำดับที่เป็น Output จะอยู่ในรูป เลขประจำรถคันด้วย '>' (เครื่องหมายมากกว่า) ทั้งนี้ String ใน List ที่คืนค่าจะอยู่ในลำดับใดก็ได้

Input	Output
<code>arrival_sequences(('R2', 'R4'), ('034', '022'))</code>	<code>['R2&gt;R4&gt;034&gt;022', 'R2&gt;034&gt;R4&gt;022', 'R2&gt;034&gt;022&gt;R4', '034&gt;R2&gt;R4&gt;022', '034&gt;R2&gt;022&gt;R4', '034&gt;022&gt;R2&gt;R4']</code>

#### • การวิเคราะห์ปัญหา

- Input: จำนวนข้อมูล \_\_\_\_\_ ชนิดข้อมูล \_\_\_\_\_
- Output: (แสดงค่า) จำนวนข้อมูล \_\_\_\_\_ ชนิดข้อมูล \_\_\_\_\_  
(คืนค่า) จำนวนข้อมูล \_\_\_\_\_ ชนิดข้อมูล \_\_\_\_\_

#### 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>