



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
  - ให้นักศึกษาพิจารณาเลือกแก้ปัญหาด้วยการใช้ Recursion หรือ Functional Control (`map`, `reduce`, `filter`, and `lambda`) ตามความเหมาะสม
- 1) 4 คะแนน (Lab09\_1\_6XXXXXXX.py) ให้เขียนฟังก์ชัน `patterned_message(message: str, pattern: str) -> None` เพื่อแสดงผลข้อความ (`message`) ตามรูปแบบ (`pattern`) ที่ระบุ โดยรูปแบบจะประกอบด้วยอักขระ '\*' และอักขระว่าง (' ') และอักขระ new line ('\n') เรียงกัน โดยฟังก์ชันจะต้องแทนที่เฉพาะอักขระ '\*' 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

- 2) 4 คะแนน (Lab09\_2\_6XXXXXXX.py) พละได้รับมรดกเป็นหนังสือโหราศาสตร์จากคุณยาย โดยมีศาสตร์ลับการดูดวงชะตาวิธีพิเศษชื่อว่า Life Path Number วิธีนี้ต้องอาศัยการนำวันเดือนปีเกิดมาบวกกัน จนกว่าจะได้เลขหลักเดียว เช่น หากเกิดวันที่ 13 กันยายนปี 2004 จะได้ผลบวก

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

พละเป็นสายมูที่เชื่อในศาสตร์ลับนี้อย่างบ้าคลั่ง จึงนำการคำนวณแบบนี้ไปใช้กับตัวเลขอื่น ๆ ทุกอย่างที่เขาเห็นในชีวิตประจำวันไม่จำกัดจำนวนหลัก เช่นเลขตัวรถประจำทาง หรือเลขสมุดบัญชีอีกด้วย

หน้าที่ของคุณคือให้เขียนฟังก์ชัน `life_path(n: int) -> int` เพื่อคืนค่า Life Path Number ของจำนวนเต็มบวก  $n$  ใด ๆ ด้วยวิธีจากหนังสือโหราศาสตร์ของพละ

Input	Output
13092004	1
7	7
35	8

- 3) 4 คะแนน (HW09\_1\_6XXXXXXX.py) ให้เขียนฟังก์ชัน `left_max(list_a: list[int]) -> list[int]` 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]

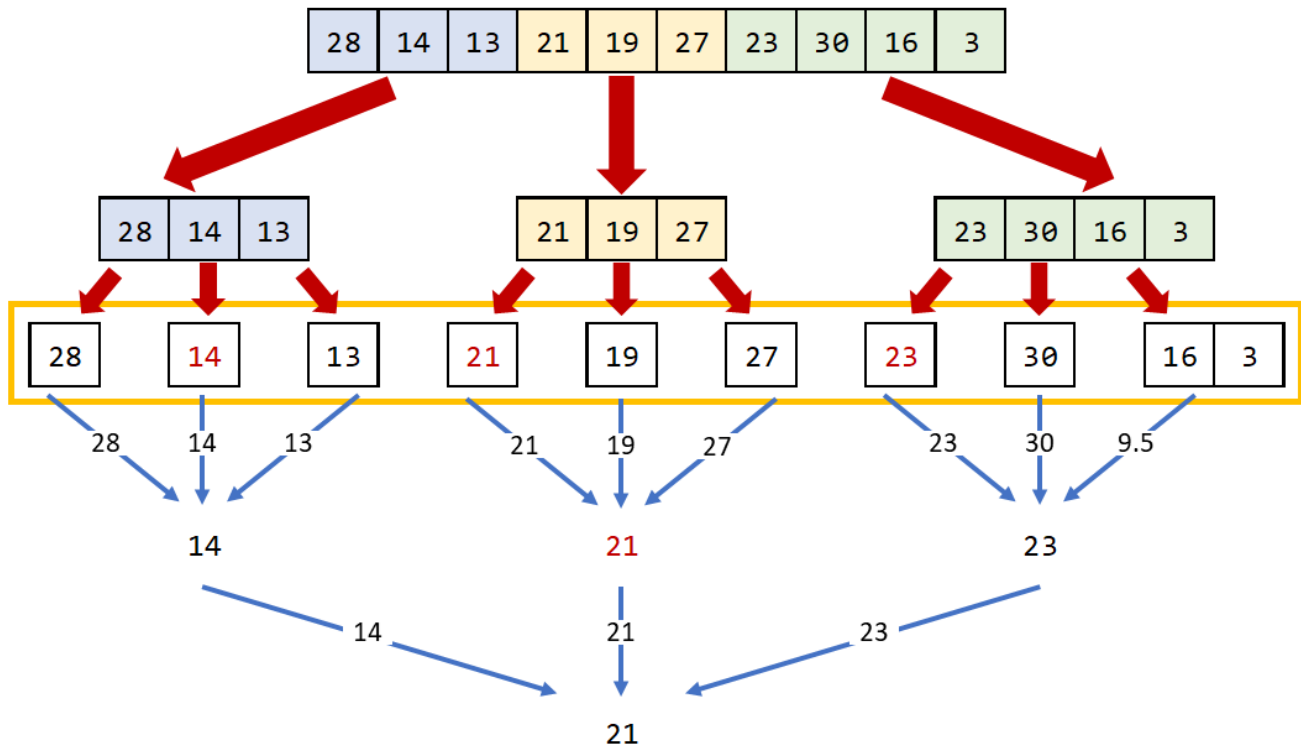
- 4) 4 คะแนน (HW09\_2\_6XXXXXXX.py) ให้เขียนฟังก์ชัน `median_of_median(list_a: list[float]) -> float` เพื่อคืนค่ากลางของ element ใน `list_a` ซึ่งเป็น List ของจำนวนจริง (`list_a` ไม่เป็น List ว่าง) โดยคำนวณค่ากลางดังกล่าวด้วยวิธี Median of Median

การคำนวณ Median of Median เป็นการคำนวณค่ากลางแบบ recursive โดยมีเงื่อนไขคือ

- 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 จะต้องได้ความยาวของทั้งสาม List ย่อยเป็น 13 13 และ 14
- แบ่งย่อยลงไปจนเหลือ List ที่มีความยาว 1 หรือ 2
- จากนั้นให้คำนวณค่ามัธยฐานในแต่ละ List ย่อยก่อน แล้วจึงนำค่าที่ได้มาหาค่ามัธยฐานอีกครั้ง
  - โดยกำหนดให้ค่ามัธยฐานของ List ที่มีความยาว 2 element คือค่าเฉลี่ยของทั้งสอง element นั้น ๆ ในรูปที่แสดงจะเห็นได้ว่าค่าเฉลี่ยของ 16 และ 3 คือ 9.5

ข้อกำหนด:

- ไม่อนุญาตให้ใช้ฟังก์ชัน `sorted()` ในการแก้ปัญหา
- คำตอบที่ถูกต้องจะต้องอยู่ภายใน  $\epsilon = 10^{-3}$

InputOutput

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

- 5) 4 คะแนน (HW09\_3\_6XXXXXXXXX.py) ให้เขียนฟังก์ชัน `unmask_id(masked_id: str) -> list[str]` เพื่อคืนค่า List ของ String แทนเลขบัตรประชาชนไทยที่เป็นไปได้ทั้งหมด เมื่อ `masked_id` เป็น String แทนเลขบัตร 13 หลัก คั่นด้วยเครื่องหมาย hyphen ('-') ตามรูปแบบของกระทรวงมหาดไทยโดยเลือกเอาเฉพาะที่หลักซ้ายสุดไม่เป็น 0

`masked_id` จะมีระหว่าง 1 - 5 ตำแหน่งที่ถูกซ่อนไว้ด้วยเครื่องหมาย '\*' เช่น '1-2345-67\*\*0-12-1' โดยกำหนดให้แต่ละเลขบัตรใน List ที่คืนค่าจะต้องเป็นเลขบัตรประชาชนที่ตรงตามอัลกอริทึมการตรวจสอบเลขบัตรประชาชนของกระทรวงมหาดไทย และจะคืนค่าในลำดับใดก็ได้ โดย โดยให้ศึกษา Algorithm ในการตรวจสอบเลขบัตรประชาชนจาก

<https://memo8.com/check-digit-thai-citizen-id-validator>

Hint:

- ในแต่ละครั้งมีเลขที่ต้องตรวจสอบ  $10^n$  จำนวนเมื่อ  $n$  คือจำนวน '\*' เท่านั้น
- พิจารณาการใช้ฟังก์ชัน `split()` และ `zip()` หรือ `str.format()` เพื่อนำตัวเลขไปแทนที่ในตำแหน่ง '\*'

<u>Input</u>	<u>Output</u>
'1-2345-67890-1*-*'	[ '1-2345-67890-10-4', '1-2345-67890-11-2', '1-2345-67890-12-1', '1-2345-67890-13-9', '1-2345-67890-14-7', '1-2345-67890-15-5', '1-2345-67890-16-3', '1-2345-67890-17-1', '1-2345-67890-18-0', '1-2345-67890-19-8']

### 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>