

ชื่อ-นามสกุล.....รหัสนักศึกษา.....ตอนเรียน.....ลำดับที่.....
กำหนดส่ง



กระบวนวิชา **229223**

Lab	
HW	
Until	

การบ้านปฏิบัติการ 9

Recursion Part II (20 คะแนน)

ข้อกำหนด

- การเรียกใช้ฟังก์ชันเพื่อการทดสอบ ต้องอยู่ภายใต้เงื่อนไข `if __name__ == '__main__':` เพื่อให้สามารถ import ไปเรียกใช้งานจาก Script อื่น ๆ ได้
- ไม่อนุญาตให้ใช้การทำซ้ำ `for`, `while` (Iterations), หรือ Data Type อื่น ๆ ที่ยังไม่สอนในบทเรียน เช่น `set` หรือ `dict` ในการแก้ปัญหา
- นักศึกษาสามารถสร้างฟังก์ชันย่อยต่าง ๆ เพิ่มเติมได้ตามความเหมาะสม
- ให้ศึกษาการสร้างฟังก์ชันทดสอบ และการเขียน `main()` จาก template ในสัปดาห์ก่อนๆ และหัดเขียนฟังก์ชันดังกล่าวเองในในสัปดาห์นี้และสัปดาห์ถัดๆ ไป

- 4 คะแนน (Lab09_1_6XXXXXXX.py) ให้เขียนฟังก์ชัน Boolean recursive `is_anagram(s1, s2)` เพื่อตรวจสอบว่า non-empty string (string ที่ไม่ใช่ string ว่าง) `s1` และ `s2` ที่ประกอบด้วยอักขระ space และตัวอักษรภาษาอังกฤษ เป็น แอนาแกรม (anagram) ซึ่งกันและกันหรือไม่ โดยฟังก์ชันจะพิจารณาเฉพาะตัวอักษรภาษาอังกฤษเท่านั้น และทำงานแบบ case-insensitive (ไม่แบ่งแยกระหว่างตัวอักษรพิมพ์ใหญ่และพิมพ์เล็ก) **ทั้งนี้ ไม่อนุญาตให้ใช้ ฟังก์ชัน `sorted()` ในการแก้ปัญหา**

คำสลับอักษร หรือ อะนาแกรม หรือ แอนาแกรม (อังกฤษ: anagram ; กรีก: *anagramma*) คือข้อความเกิดจากการนำตัวอักษรในอีกข้อความหนึ่งมาเรียงสลับที่กัน เช่น Eleven plus two เป็นอะนาแกรมของ Twelve plus one

Ref: <https://th.wikipedia.org/wiki/คำสลับอักษร>

Input	Output
Tom Marvolo Riddle I am Lord Voldemort	True
cat tab	False

- การวิเคราะห์ปัญหา

- Input: จำนวนข้อมูล _____ ชนิดข้อมูล _____
- Output: (แสดงค่า) จำนวนข้อมูล _____ ชนิดข้อมูล _____
(คืนค่า) จำนวนข้อมูล _____ ชนิดข้อมูล _____

$n = \text{list}(\text{range}(x, y+1))$

2 / 3

ชื่อ-นามสกุล.....รหัสนักศึกษา.....ตอนเรียน.....ลำดับที่.....

4 คะแนน (Lab09_2_6XXXXXXX.py) ให้เขียนฟังก์ชัน recursive `sum_prime_in_range(x, y)` เพื่อคืนค่าผลบวกของจำนวนเฉพาะในช่วงตั้งแต่ x ($x > 1$) ถึง y ($y \geq x > 1$) (Inclusive)

Hint: พิจารณาเขียนฟังก์ชัน boolean `is_prime(n)` และเรียกใช้ฟังก์ชัน `filter()`

Input	Output
3 20	75
3 3	3

• การวิเคราะห์ปัญหา

• Input:	จำนวนข้อมูล	ชนิดข้อมูล
• Output:	(แสดงค่า)	จำนวนข้อมูล
	(คืนค่า)	จำนวนข้อมูล
		ชนิดข้อมูล

2) 4 คะแนน (HW09_1_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

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

โดยการคำนวณแบบนี้สามารถนำไปใช้กับตัวเลขอื่นๆ ไม่จำกัดจำนวนหลัก เช่น เลขตัวรถประจำทาง หรือเลขสมุดบัญชีอีกด้วย

Input	Output
13092004	1
7	7
35	8

• การวิเคราะห์ปัญหา

• Input:	จำนวนข้อมูล	ชนิดข้อมูล
• Output:	(แสดงค่า)	จำนวนข้อมูล
	(คืนค่า)	จำนวนข้อมูล
		ชนิดข้อมูล

```
if len(str(n)) == 1:
    return n
else:
    n = sum(int(n[:]))
    return life_path(n)
```

ชื่อ-นามสกุล.....รหัสนักศึกษา.....ตอนเรียน.....ลำดับที่.....

- 3) 4 คะแนน (HW09_2_6XXXXXXX.py) ให้เขียนฟังก์ชัน recursive longest_digit_run(n) เพื่อคืนค่าความยาวหลักของของตัวเลขที่ปรากฏติดกันมากที่สุดในจำนวนเต็ม n ($n \neq 0$) เช่น longest_digit_run(11777332) จะมีผลลัพธ์เป็น 3 เนื่องจากตัวเลขที่ติดกันยาวที่สุดคือ 7 เป็นจำนวน 3 หลัก

Input	Output
11777332	3
1177332	2

• การวิเคราะห์ปัญหา

- Input: จำนวนข้อมูล _____ ชนิดข้อมูล _____
- Output: (แสดงค่า) จำนวนข้อมูล _____ ชนิดข้อมูล _____
(คืนค่า) จำนวนข้อมูล _____ ชนิดข้อมูล _____

- 4) 4 คะแนน (HW09_3_6XXXXXXX.py) ให้เขียนฟังก์ชัน recursive left_max(n) เพื่อคืนค่าผลลัพธ์จากการแปลง 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 _____

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/gdr223>

Input

↓ = 1/2

- Input:

จำนวนข้อมูล

ชนิดข้อมูล

- (แสดงค่า)

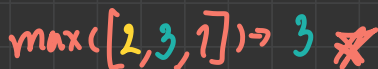
จำนวนข้อมูล

ชนิดข้อมูล

- (คินค่า)

จำนวนข้อมูล

ชนิดข้อมูล



```
n = str(n)
```

```
if len(n) == 1:
```

```
return n
```

```
else:
```

เมื่อ +1

```
if n[0] == n[1]: # ตัวแรกเหมือนกับตัวถัดไป
```

```
current_num += 1
```

else: # $n[0] \neq n[1]$ หักไปครึ่งเดียว อีกที

```
current_max = 7
```

```
if current_num
```

```
def helper(n,max,current):
    n = str(n)
    if len(str(n)) == 1:
        return n
```

```

else:
    if n[0] == n[1]:
        current += 1
    else:
        current = 1
    if current >= max:
        max = current

```

```
helper(n[1:],max,current)
```

```
return max
```

```
print(longest digit number(11177))
```

↑ 7: return max