

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



กระบวนวิชา 229223

Lab	
HW	
Until	

การบ้านปฏิบัติการ 8

Recursion Part I (20 คะแนน)

ข้อกำหนด

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

1) 4 คะแนน (Lab08_1_6XXXXXXX.py) ให้เขียนฟังก์ชัน `recursive gcd(x, y)` เพื่อคำนวณและคืนค่าตัวหารร่วมมากของจำนวนเต็ม x ($x \neq 0$) และ y ($y \neq 0$) ด้วยวิธีของ Euclid จากบทเรียนเรื่อง Numbers ทั้งนี้ไม่อนุญาตให้ใช้ฟังก์ชัน `math.gcd()` ในการแก้ปัญหา

Input	Output
19 71	1
-39 78	39

การวิเคราะห์ปัญหา

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

COMPUTER SCIENCE
Chiang Mai University

Handwritten notes and calculations:

78 % 39 = 0

71 % 19 = 14

19 % 14 = 5

14 % 5 = 4

5 % 4 = 1

4 % 1 = 0

def gcd(x,y):
if x % y == 0:
return y
else:
gcd(y, x%y)

Lab 2

def reverse_digits(x):

if (x % 10) == (x // (10⁴ * (len(str(x)) - 1))):

return x

else:

x = (x % 10) * (10⁴ * (len(str(x)) - 1)) + reverse_digits(x // 10⁴ * (len(str(x)) - 2))

reverse_digits(x)

$$\begin{array}{r} 1234 \\ \hline 1000 \end{array} \rightarrow 1 \quad 3$$

$$10^3 \quad 4^1 - 1$$

$$\begin{array}{r} 12.34 \\ \hline 100 \end{array}$$

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

- 2) 4 คะแนน (Lab08_2_6XXXXXXX.py) ให้เขียนฟังก์ชัน recursive reverse_digits(x) เพื่อคืนค่าผลลัพธ์จากการกลับหลักจำนวนเต็ม x ใดๆ ทั้งนี้ไม่อนุญาตให้ใช้ str, tuple หรือ list ในการแก้ปัญหา (e.g. [::-1])

Input	Output
1234	4321
1	1

• การวิเคราะห์ปัญหา

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

- 3) 4 คะแนน (HW08_1_6XXXXXXX.py) **[Attachments]** ให้เขียนฟังก์ชัน recursive pi(n) เพื่อคำนวณ และคืนค่า π (pi) จากการประมาณ โดยมีความละเอียดตามจำนวนพจน์ที่ระบุด้วยจำนวนเต็ม n จาก series ผลบวกดังนี้

$$\pi = 3 + \left(\frac{4}{2 \times 3 \times 4}\right) - \left(\frac{4}{4 \times 5 \times 6}\right) + \left(\frac{4}{6 \times 7 \times 8}\right) - \left(\frac{4}{8 \times 9 \times 10}\right) + \dots$$

$$\pi = 3 + \frac{4}{2n(2n+1)(2n+2)} - \frac{4}{2n(2n+1)(2n+2)}$$

ค่า π จาก series ผลบวกดังกล่าวเป็นการประมาณค่าจากพจน์ที่ 0 - n ของ series โดยพจน์ที่ 0 จะเท่ากับ 3 ดังนั้น

$$\pi(0) = 3 \quad \text{if : return x} \quad \approx 3.000000$$

$$\pi(2) = 3 + \left(\frac{4}{2 \times 3 \times 4}\right) - \left(\frac{4}{4 \times 5 \times 6}\right) \quad \text{else : } \pi += 4 / (2n \times (2n+1) \times (2n+2)) \quad \approx 3.133333$$

$$\pi(5) = 3 + \left(\frac{4}{2 \times 3 \times 4}\right) - \left(\frac{4}{4 \times 5 \times 6}\right) + \left(\frac{4}{6 \times 7 \times 8}\right) - \left(\frac{4}{8 \times 9 \times 10}\right) + \left(\frac{4}{10 \times 11 \times 12}\right) \quad \approx 3.142713$$

ทั้งนี้ไม่อนุญาตให้ใช้ list หรือ map ในการแก้ปัญหา

Input	Output
0	3
1	3.1666666666666665
2	3.1333333333333333
5	3.1427128427128426

• การวิเคราะห์ปัญหา

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

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

- 4) 4 คะแนน (HW08_2_6XXXXXXX.py) ให้เขียนฟังก์ชัน `recursive base_b(x, b)` เพื่อคืนค่าจำนวนเต็ม แทนจำนวนเต็มบวก x ในฐาน 10 เมื่อเปลี่ยนเป็นฐาน b ($2 \leq b \leq 10$) ทั้งนี้ไม่อนุญาตให้ใช้การแปลงฐานผ่านฟังก์ชัน `int()` ในการคำนวณ

Input	Output
8 2	1000
11 3	102

• การวิเคราะห์ปัญหา

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

- 5) 4 คะแนน (HW08_3_6XXXXXXX.py) ให้เขียนฟังก์ชัน `recursive patterned_message(message, pattern)` เพื่อแสดงผลข้อความ (`message`) ตามรูปแบบ (`pattern`) ที่ระบุ โดยรูปแบบจะประกอบด้วยอักขระ * และ space เรียงกันโดยโปรแกรมต้องแทนที่อักขระ * ด้วย อักขระจาก `message` ที่ไม่ใช่ อักขระว่าง

Hint - ไม่จำเป็นต้องสร้าง String ใหม่เพื่อการแสดงผล

Function Call	Output
<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
	(คืนค่า)	จำนวนข้อมูล
		ชนิดข้อมูล

```
patterned_message("123", "** ** * ** **")
```

```
12 312 31 231
```

```
12 312 31 23 1
```

190 \neq \rightarrow message[0:

n%len(message)

↓
⇒ message[1:] + message[0]

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

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>

