

Lab	พฤ. 16 พ.ย. 2560 ก่อนเที่ยงคืน

การบ้านปฏิบัติการ 11 *n*-Dimensional Lists (20 คะแนน)

ข้อกำหนด

I manut

- การเรียกใช้ฟังก์ชันเพื่อการทดสอบ ต้องอยู่ภายใต้เงื่อนไข if __name__ == '__main__' : เพื่อความ
 สะดวกในการ import จาก Script อื่น ๆ
- ii. ไม่อนุญาตให้ใช้ recursion, set, dict หรือเรื่องอื่นๆ ที่ยังไม่ได้กล่าวถึงในรายวิชาในการแก้ปัญหา
- iii. นักศึกษาสามารถสร้างฟังก์ชันย่อยต่างๆ เพิ่มเติมได้ตามความเหมาะสม
- 1) **4 คะแนน** (Lab11_1_6XXXXXXX.py) ให้เขียนฟังก์ชัน matrix_mult(m1, m2) เพื่อทำการหาผลคูณ (wikipedia: https://goo.gl/S0DDZv) ของเมทริกซ์ m1 และ เมทริกซ์ m2 โดยฟังก์ชันจะทำงานแบบ Nondestructive กล่าวคือจะ<u>คืนค่า</u>ผลคูณที่ได้โดยไม่เปลี่ยนแปลงเมทริกซ์ m1 และ m2 ในรูปแบบ List สองมิติ หากไม่ สามารถหาผลคูณได้ให้คืนค่า None

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \times \begin{bmatrix} 7 & 8 \\ 9 & 10 \\ 11 & 12 \end{bmatrix} = \begin{bmatrix} 58 \\ \end{bmatrix}$$

<u>input</u>	Output
[[1, 2, 3], [4, 5, 6]] [[7, 8], [9, 10], [11 12]]	[[58, 64], [139, 154]]

• การวิเคราะห์ปัญหา

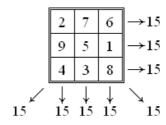
• Input: จำนวนข้อมูล ชนิดข้อมูล
 • Output: (แสดงค่า) จำนวนข้อมูล ชนิดข้อมูล ชนิดข้อมูล
 (คืนค่า) จำนวนข้อมูล ชนิดข้อมูล

2) **4 คะแนน** (Lab11_2_6xxxxxxxx.py) ให้เขียนฟังก์ชัน remove_row_col(*list_a*, row, col) เพื่อทำการ ลบทุก Element ในแถวที่ row และ คอลัมน์ที่ col ออกจาก List สองมิติ *list_a* แล้ว<u>คืนค่า</u>ผลลัพธ์ที่ได้ โดย ฟังก์ชันจะต้องทำงานแบบ Non-destructive ทั้งนี้หาก row หรือ col อยู่นอกขอบเขตที่จะทำการลบได้ จะต้องไม่มี การเปลี่ยนแปลงในแนว row หรือ col ดังกล่าว

<u>Input</u>	Output
[[2, 3, 4, 5], [8, 7, 6, 5], [0, 1, 2, 3]] 1 2	[[2, 3, 5], [0, 1, 3]]
[[2, 3, 4, 5], [8, 7, 6, 5], [0, 1, 2, 3]] 1 -3	[[2, 4, 5], [0, 2, 3]]

• การวิเคราะห์ปัญหา

3) **4 คะแนน** (Lab11_3_6xxxxxxxx.py) ให้เขียนฟังก์ชัน Boolean is_magic_square(board) เพื่อทดสอบว่า list 2 มิติ board เป็น จัตุรัสกล (Magic Square) หรือไม่



ในทางคณิตศาสตร์ จัตุรัสกล (magic square) ขนาด n คือการนำตัวเลขจำนวนเต็ม<u>ทุกตัว</u> ตั้งแต่ 1 ถึง n^2 (ไม่มี ตัวเลขซ้ำ) มาเรียงลงในตารางรูปสี่เหลี่ยมจัตุรัสขนาด $n \times n$ ซึ่งผลบวกของจำนวนในแต่ละแถว, แต่ละหลัก, แต่ละ แนวทแยงจะเท่ากันทั้งหมด ดังตัวอย่าง Magic Square ที่มี n=3 ในรูปด้านบน

<u>Input</u>	Output
[[2, 7, 6], [9, 5, 1], [4, 3, 8]]	True
[[5, 5, 5], [5, 5, 5], [5, 5, 5]]	False

<u>Input</u> <u>Output</u>

[[7, 12, 1, 14],	True
[2, 13, 8, 11],	
[16, 3, 10, 5],	
[9, 6, 15, 4]]	

• การวิเคราะห์ปัญหา

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

4) **4 คะแนน** (Lab11_5_6xxxxxxxx.py) ให้เขียนฟังก์ชันแบบ sum_nested_list(*list_a*) ที่ทำงานแบบ Non-destructive เพื่อ<u>คืนค่า</u>ผลรวมของจำนวนเต็มทั้งหมดใน *list_a* โดยแต่ละสมาชิกของ *list_a* มีชนิดข้อมูลที่เป็นไป ได้ <u>2 ประเภท</u> คือ เป็นจำนวนเต็ม (int) หรือเป็น list โดย list ที่เป็นสมาชิกดังกล่าว ก็สามารถมีสมาชิกเป็น จำนวนเต็มและ list ได้เช่นกัน ตัวอย่างเช่น

จากตัวอย่างเป็น list ที่มี 4 สมาชิก โดยสมาชิก ที่ 0, และ 1 และ 3 ของ list มีชนิดเป็นจำนวนเต็ม ในขณะที่ <u>สมาชิกที่ 2</u> มีชนิดเป็น list: [[3, 0], 4] และผลรวมของจำนวนเต็มทั้งหมดจะมีค่า 1 + 2 + 3 + 0 + 4 + 8 = 18

Hint:

- พิจารณาเรียกใช้ฟังก์ชัน isinstance(object, classinfo) เพื่อตรวจสอบชนิดของสมาชิก เช่น isinstance([3], list) จะคืนค่าเป็น **True**
- พิจารณาความแตกต่างของ list_x.append([5]) และ list_x.extend([5])
- พิจารณาเรียกใช้ฟังก์ชัน sum($list_x$) เพื่อหาผลรวมของสมาชิกใน $list_x$

<u>Input</u> <u>Output</u>

[1, 2, [[2, [[145], 34]], [48, 22]]]	254
[58, [31, [90]]]	179
[61, [[2, [75]], 8000, [39]], [58, [46]]]	8281
[81, [[31, [159]], 9577, [22, [181, [41]]]]]	10092

• การวิเคราะห์ปัญหา

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

การส่งงาน

- 1. ลักษณะ/ลำดับข้อความของการรับค่า/แสดงผล จะ<u>ต้องเป็นไปตามที่ระบ</u>ุในตัวอย่างการ run
- 2. ไฟล์งานที่ส่ง จะต้องมีการแทรก comment ที่ต้นไฟล์ตามข้อกำหนดใน website รายวิชา
- 3. ไฟล์งานโปรแกรมที่ส่ง จะต้องมีการแทรก pseudocode เป็น comment ในแต่ละขั้นตอน
- 4. Upload ไฟล์ source code ตามที่ระบุในแต่ละข้อ ไปยัง website ที่ใช้ส่งการบ้าน http://hw.cs.science.cmu.ac.th/CS_HW/p204111.html ตาม section ที่นักศึกษาเรียน