

11.17.70.11.0		
Lab		
HW		
Until		

## การบ้านปฏิบัติการ 13

## n-dimensional Lists and Nested Collections (20 คะแหน)

## <u>ข้อกำหนด</u>

- การเรียกใช้ฟังก์ชันเพื่อการทดสอบ ต้องอยู่ภายใต้เงื่อนไข if \_\_name\_\_ == '\_\_main\_\_' : เพื่อให้สามารถ
   import ไปเรียกใช้งานจาก Script อื่น ๆ ได้อย่างเป็นมาตรฐาน
- ii. ทุกข้อต้องมีการสร้างฟังก์ชัน my\_id() โดยให้คืนค่าสายอักขระแทนเลขประจำตัวนักศึกษา 9 หลัก
- 1) **4 คะแนน** (Lab13\_1\_5XXXXXXX.py) [Attachment] ให้เขียนฟังก์ชัน matrix\_mult(m1, m2) เพื่อทำการ หาผลคูณของเมทริกซ์ m1 และ เมทริกซ์ m2 (wikipedia: <a href="https://goo.gl/S0DDZv">https://goo.gl/S0DDZv</a>) โดยฟังก์ชันจะทำงานแบบ Non-destructive กล่าวคือจะ<u>คืนค่า</u>ผลคูณที่ได้โดยไม่เปลี่ยนแปลงเมทริกซ์ m1 และ m2 ที่อยู่ในในรูปแบบ List สองมิติ ทั้งนี้หากไม่สามารถหาผลคูณได้ให้คืนค่า None

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

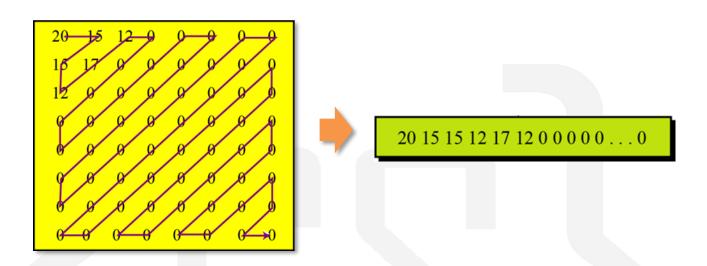
<u>Input</u> <u>Output</u>

[[1, 2, 3], [4, 5, 6]] [[7, 8], [9, 10], [11,12]]	[[58, 64], [139, 154]]
[[1, 2, 3], [4, 5, 6]] [[7, 8, 5, 9, 3], [9, 10, -3, 7, 13], [11, 12, 6, 2, 9]]	[[58, 64, 17, 29, 56], [139, 154, 41, 83, 131]]

• การวิเคราะห์ปัญหา

Input: จำนวนข้อมูล\_\_\_\_ชนิดข้อมูล\_\_\_\_\_
 Output: (แสดงค่า) จำนวนข้อมูล\_\_\_\_\_ชนิดข้อมูล\_\_\_\_\_
 (คืนค่า) จำนวนข้อมูล\_\_\_\_\_ชนิดข้อมูล\_\_\_\_\_

2) **4 คะแนน** (Lab13\_2\_6XXXXXXXX.py) [Attachment] ในการบีบอัดภาพใน Format JPEG หลังจากทำขั้นตอน Discrete Cosine Transform แล้ว เราจะต้องอ่านข้อมูลที่ได้ใน Pixel Block ของจำนวนเต็ม ขนาด 8×8 ในรูปแบบ สลับฟันปลาดังแสดงด้านล่าง เพื่อนำ List ที่ได้ไปทำการบีบอัดแบบ Run-Length Compression อีกที



ให้เขียนฟังก์ชัน read\_pixel(pixel\_block) เพื่อคืนค่า List แทนตัวเลขในลำดับการอ่าน Matrix ในรูปแบบ สลับฟันปลาดังอธิบายเมื่อ pixel\_block เป็น Matrix จัตุรัสของจำนวนเต็ม

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

• การวิเคราะห์ปัญหา

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

3) 4 คะแนน (HW13\_1\_6XXXXXXX.py) [Attachment] ให้เขียนฟังก์ชัน square\_matrix(list\_x) เพื่อทำให้ List 2 มิติ list\_x ที่มีสมาชิกเป็น<u>จำนวนเต็ม</u>กลายเป็น matrix จัตุรัสโดยเติม 0 เพื่อให้มีขนาด row และ column เท่ากัน โดยจะต้องคงทุก element ใน List เดิมไว้ และจำนวน 0 ที่เติมต้องเป็นจำนวนที่น้อยที่สุดที่เป็นไปได้ ทั้งนี้กำหนดให้ ฟังก์ชันทำงานแบบ Destructive และแต่ละ element จะต้องไม่เป็น alias ซึ่งกันและกัน

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

[[2 2 4]	[[2 2 4]
[[2, 3, 4],	[[2, 3, 4],
[1, 2, 3]]	[1, 2, 3],
	[0, 0, 0]]

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

[[1, 2],	[[1, 2, 0, 0, 0],
[1, 2, 3],	[1, 2, 3, 0, 0],
[1, 2],	[1, 2, 0, 0, 0],
[1, 2],	[1, 2, 0, 0, 0],
[1]]	[1, 0, 0, 0, 0]]

• การวิเคราะห์ปัญหา

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

4) **4 คะแนน** (HW13\_2\_6XXXXXXXX.py) **[Attachment]** ให้เขียนฟังก์ชัน count\_vote(*pref\_matrix*) เพื่อ<u>คืนค่า</u> คะแนนโหวตของ Pokémon ที่ได้จากการลงคะแนน Twitter-wide Favorite Pokémon แบบจัดลำดับ

การลงคะแนนแบบจัดลำดับ (อังกฤษ: ranked voting) หรือเรียกอีกอย่างว่า การลงคะแนนตามลำดับความชอบ (อังกฤษ: Ranked-choice Voting) หรือ การลงคะแนนตามความชอบ (อังกฤษ: Preferential Voting) เป็นระบบการ ลงคะแนนใด ๆ ที่ผู้ลงคะแนนเสียงใช้การจัดลำดับผู้สมัคร (หรือลำดับความชอบ) ในบัตรลงคะแนนเพื่อเลือกผู้สมัคร มากกว่าหนึ่งรายขึ้นไป และเพื่อเรียงลำดับตัวเลือกผู้สมัครทั้งหมดเป็นลำดับที่หนึ่ง สอง สาม ไปจนครบ (Wikipedia)

ในตัวแปร  $pref_matrix$  แต่ละ Row จะแทนการเลือกของ Voter แต่ละคน และ จำนวน Column ทั้งหมดแทน ตัวเลือกที่เลือกได้ โดยการคำนวนคะแนนจะให้น้ำหนักคะแนนที่สูงที่สุดแก่ตัวเลือกอันดับแรกเช่น กรณีเลือกได้ 4 ตัวเลือก ตัวเลือกแรกจะได้น้ำหนักคะแนน 4 ตัวเลือกที่ 2 จะได้น้ำหนักคะแนน 3 ลดหลั่นกันไป จนตัวเลือกสุดท้าย จะมีน้ำหนักคะแนนเท่ากับ 1 ในกรณีที่เลือกได้ n ตัวเลือก อันดับที่ 1 ก็จะได้น้ำหนักคะแนนเท่ากับ n แทน เช่นใน ตัวอย่างด้านล่าง คะแนนของ Pikachu จะเท่ากับ n + 1 + 3 + 2 = 8

ฟังก์ชันจะคืนค่า list ของ tuple ที่ประกอบด้วยชื่อ Pokémon ทั้งหมดที่มีผู้ vote ให้ และคะแนนที่ได้ เรียง ตามลำดับคะแนนจากมากไปน้อย และลำดับตัวอักษรในพจนานุกรมภาษาอังกฤษกรณีที่คะแนนเท่ากัน ทั้งนี้ กำหนดให้ทุก row มีความยาวเท่ากัน

[['Mewtwo', 'Pikachu', 'Suicune'],
['Mewtwo', 'Suicune', 'Pikachu'],
['Pikachu', 'Rayquaza', 'Charizard'],
['Suicune', 'Pikachu', 'Charizard']]
('Charizard', 2),
('Rayquaza', 2)]

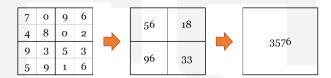
• การวิเคราะห์ปัญหา

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

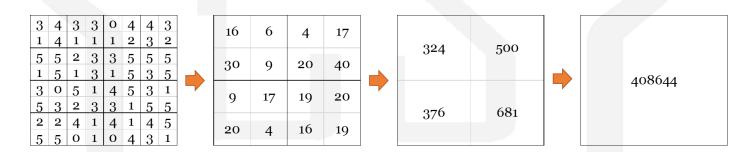
5) **4 คะแนน** (HW13\_3\_6XXXXXXXX.py) **[Attachment]** ให้เขียนฟังก์ชัน sum\_d\_product(m) เพื่อ<u>คืนค่า</u>ผลบวก ของผลคูณทแยงใน matrix m ที่มีขนาด  $n \times n$  เมื่อ n สามารถเขียนในรูปของ  $2^x$  (x เป็นจำนวนเต็มบวก)

โดยกรณี matrix m ขนาด 2×2 เช่น  $\cfrac{a + b}{c + d}$  สามารถหาผลลัพธ์ได้จากสูตร  $a \times d + c \times b$ 

ดังนั้น matrix 3 4 จะมี sum\_d\_product() =  $(1 \times 4) + (3 \times 2) = 10$  กรณีต้องการหา sum\_d\_product() ของ matrix ขนาดใหญ่กว่า  $2 \times 2$  ทำได้โดยการหา sum\_d\_product() ของ matrix ย่อย ขนาด  $2 \times 2$  ก่อน แล้วหา sum\_d\_product() ของ matrix ผลลัพธ์อีกที เช่นกรณี matrix ขนาด  $4 \times 4$  จะมีขั้นตอนดังนี้



หรือกรณี matrix 8 × 8



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

[[3, 3, 3, 2], [2, 0, 3, 1], [2, 1, 2, 3], [1, 0, 2, -1]	33
[[1, 1, 5, -1], [12, 2, -2, 0], [4, 8, 8, 12], [4, 12, 12, 15]]	3856 <b>SELENCE</b>
[[0, -1, -1, 3, 2, 3, -1, 3], [3, -1, -1, 2, 0, -1, 2, 1], [3, 0, 1, 2, 3, 1, 3, 1], [2, 2, 1, -1, -1, 2, 0, 3], [1, 3, 2, 1, 3, 2, 2, 1], [1, 2, 2, 1, 3, 3, 1, 3], [2, 2, 2, 2, 2, 2, 3, 3], [1, 3, 2, 3, 1, 1, 2, 2]]	-6290 University