

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



กระบวนวิชา **229223**

Lab	
HW	
Until	

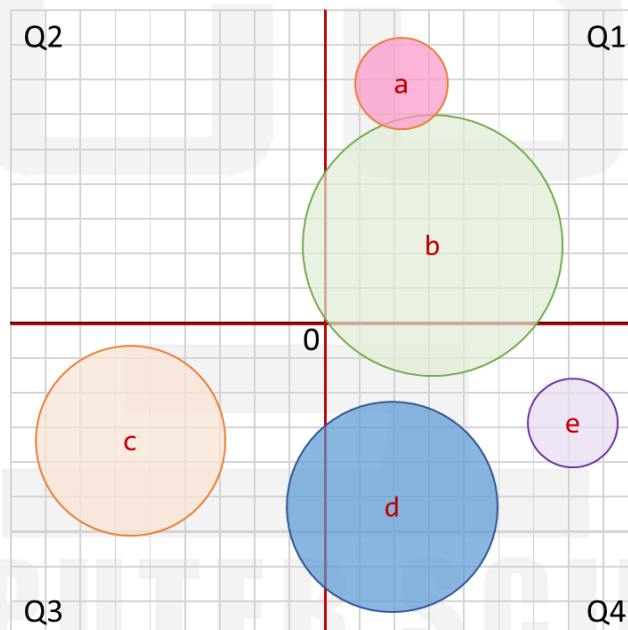
### การบ้านปฏิบัติการ 14

#### File I/O and Problem-Solving Practice (20 คะแนน)

##### ข้อกำหนด

- การเรียกใช้ฟังก์ชันเพื่อการทดสอบ ต้องอยู่ภายใต้เงื่อนไข `if __name__ == '__main__':` เพื่อให้สามารถ import ไปเรียกใช้งานจาก Script อื่น ๆ ได้
- ฟังก์ชัน `main()` ควรอ่าน Input จากไฟล์แนบด้วยวิธี Command Redirection เพื่อความสะดวกในการทดสอบ และหาข้อผิดพลาด

- 4 คะแนน (Lab14\_1\_6XXXXXXX.py) ให้เขียนฟังก์ชัน `count_segment(list_a)` เพื่อบันทึกจำนวนส่วนของวงกลมที่อยู่ใน Quadrant ต่างๆ ที่ระบุด้วย `list_a` โดย `list_a` จะเป็น List ของ tuple ที่อยู่ในรูป `(px, py, r)` โดย `px` และ `py` คือพิกัดในแนวแกน `x` และแกน `y` ตามลำดับ และ `r` คือ รัศมีวงกลม ( $r > 0$ ) โดยฟังก์ชันจะคืนค่า tuple แทนจำนวนวงกลมหรือ Segment (ส่วนของวงกลม) ที่อยู่ใน Quadrant 1, 2, 3 และ 4 ตามลำดับ



เช่นจากรูปด้านบน ฟังก์ชันจะคืนค่า (2, 1, 2, 3)

##### Input

##### Output

<pre>[(2, 7, 1.5),      # a  (3.2, 2.5, 4.06), # b  (-5.5, -4.5, 2.5), # c  (2, -5.2, 3),     # d  (7.2, -2.8, 1.2)] # e</pre>	(2, 1, 2, 3)
--	--------------

- 4 คะแนน (Lab14\_2\_6XXXXXXX.py) ให้เขียนฟังก์ชัน `remove_row_col(list_a, row, col)` เพื่อคืนค่า

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

ผลลัพธ์ที่ได้จากการลบทุก Element ในแถวที่ *row* และ คอลัมน์ที่ *col* ออกจาก List สองมิติ *list\_a* ทั้งนี้หาก *row* หรือ *col* อยู่นอกขอบเขตที่จะทำการลบได้ จะต้องไม่มีการเปลี่ยนแปลงในแนว *row* หรือ *col* ดังกล่าว

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

• การวิเคราะห์ปัญหา

- Input: จำนวนข้อมูล \_\_\_\_\_ ชนิดข้อมูล \_\_\_\_\_
- Output: (แสดงค่า) จำนวนข้อมูล \_\_\_\_\_ ชนิดข้อมูล \_\_\_\_\_  
(คืนค่า) จำนวนข้อมูล \_\_\_\_\_ ชนิดข้อมูล \_\_\_\_\_

3) 4 คะแนน (HW14\_1\_6XXXXXXXXX.py) ให้เขียนฟังก์ชัน Destructive reshape(matrix) เพื่อเปลี่ยนแปลงขนาดของ list สองมิติในตัวแปร *matrix* ให้มีขนาด  $m \times n$  โดยกำหนดให้  $m$  น้อยกว่าหรือเท่ากับ  $n$  เสมอ และความต่างของ  $m$  และ  $n$  จะต้องไม่เกิน 1 ทั้งนี้ผลลัพธ์ที่ได้จะต้องมีจำนวนสมาชิกเท่ากันในทุก row และเรียงสมาชิกตามลำดับในเดิมในตัวแปร *matrix* ที่ละ row และ column จากซ้ายบนไปขวาล่าง โดยสามารถเพิ่มจำนวนสมาชิกที่เป็น 0 ได้ถ้าจำเป็น โดยจำนวน element ที่มีค่า 0 ที่เพิ่มเข้าไปจะต้องมีค่าน้อยที่สุดที่เป็นไปได้

Input	Output
<pre>[[2, 3, 4],  [1, 2, 3]]</pre>	<pre>[[2, 3, 4],  [1, 2, 3]]</pre>
<pre>[[1, 2],  [1, 2, 3],  [1, 2],  [1, 2],  [1]]</pre>	<pre>[[1, 2, 1, 2],  [3, 1, 2, 1],  [2, 1, 0, 0]]</pre>
<pre>[[1, 2],  [3, 4],  [5, 6]]</pre>	<pre>[[1, 2, 3],  [4, 5, 6]]</pre>

• การวิเคราะห์ปัญหา

• Input: จำนวนข้อมูล \_\_\_\_\_ หน่วยข้อมูล \_\_\_\_\_

• Output: (แสดงค่า) จำนวนข้อมูล \_\_\_\_\_ หน่วยข้อมูล \_\_\_\_\_  
(คืนค่า) จำนวนข้อมูล \_\_\_\_\_ หน่วยข้อมูล \_\_\_\_\_

4) **4 คะแนน** (HW14\_2\_6XXXXXXX.py) ให้เขียนฟังก์ชัน `append_ranking(infile_name='score_in.txt', outfile_name='score_out.txt')` เพื่ออ่านข้อมูลจากไฟล์ที่ระบุชื่อด้วยตัวแปร `infile_name` แล้วเขียนผลลัพธ์ลงในไฟล์ที่มีชื่อระบุด้วยตัวแปร `outfile_name` โดยกำหนดให้ content ภายในไฟล์มีลักษณะเดียวกันกับไฟล์ที่อ่านเข้า แต่ให้เพิ่มลำดับในรายวิชาที่นักศึกษาได้รับไว้ที่ส่วนท้ายของแต่ละบรรทัด โดยนักศึกษาที่ได้ลำดับที่ 1 จะเป็นนักศึกษาที่ได้คะแนนเก็บ 10% มากที่สุด

เช่นเดียวกันกับใน Lab12\_1 จากสัปดาห์ที่ 12 การประมวลผลคะแนนจะคิดจากการสอบทั้งหมด 3 ครั้ง ที่แต่ละครั้งมีคะแนนเต็ม 100 คะแนน แล้วเลือกคะแนนสูงสุดเพียงสองครั้ง เพื่อเก็บคะแนนรวม 10 % ของคะแนนเก็บทั้งภาคการศึกษา (ครั้งละ 5%) ทั้งนี้สามารถเรียกใช้ฟังก์ชันจากที่บ้านในสัปดาห์ที่ 12 ได้

**Hint:**

- พิจารณาเรียกใช้ฟังก์ชัน sorted()
- พิจารณาการใช้ dictionary ในการแก้ปัญหา

**Input:** 'score\_in.txt'

**Output:** 'score\_out.txt'

6XXX10111	53.57	59.69	55.36		6XXX10111	53.57	59.69	55.36	3
6XXX10112	64.04	None	62.84		6XXX10112	64.04	None	62.84	2
6XXX10113	88.53	91.22	84.45		6XXX10113	88.53	91.22	84.45	1

- การวิเคราะห์ปัญหา

- Input:                      จำนวนข้อมูล \_\_\_\_\_ ชนิดข้อมูล \_\_\_\_\_
- Output:            (แสดงค่า)          จำนวนข้อมูล \_\_\_\_\_ ชนิดข้อมูล \_\_\_\_\_
- (คีนค่า)                 จำนวนข้อมูล \_\_\_\_\_ ชนิดข้อมูล \_\_\_\_\_

5) 4 คะแนน (HW14 3 6XXXXXXXXX.py) ให้เขียนฟังก์ชัน `count_vote(pref_matrix)` เพื่อคืนค่าคะแนนโหวต

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

ของ Pokémon แต่ละตัวที่คำนวณได้จากการลงคะแนนโหวต Twitter-wide Favorite Pokémon แบบจัดลำดับ (ranked voting)

การลงคะแนนแบบจัดลำดับ (อังกฤษ: ranked voting) หรือเรียกอีกอย่างว่า การลงคะแนนตามลำดับความชอบ (อังกฤษ: ranked-choice voting) หรือ การลงคะแนนตามความชอบ (อังกฤษ: preferential voting) เป็นระบบการลงคะแนนใดๆ ที่ผู้ลงคะแนนเสียงใช้การจัดลำดับผู้สมัคร (หรือลำดับความชอบ) ในบัตรลงคะแนนเพื่อเลือกผู้สมัครมากกว่าหนึ่งรายขึ้นไป และเพื่อเรียงลำดับตัวเลือกผู้สมัครทั้งหมดเป็นลำดับที่หนึ่ง สอง สาม ไปจนครบ (Wikipedia)

ในตัวแปร *pref\_matrix* แต่ละ row จะแทนการเลือกของ voter แต่ละคน และ จำนวน column ทั้งหมดแทนตัวเลือกที่เลือกได้ โดยการคำนวณคะแนนจะให้น้ำหนักคะแนนที่สูงที่สุดแก่ตัวเลือกอันดับแรกเช่น กรณีเลือกได้ 4 ตัวเลือก ตัวเลือกแรกจะได้น้ำหนักคะแนน 4 ตัวเลือกที่ 2 จะได้น้ำหนักคะแนน 3 ลดหลั่นกันไป จนตัวเลือกสุดท้ายจะมีน้ำหนักคะแนนเท่ากับ 1 ในกรณีที่เลือกได้  $n$  ตัวเลือก อันดับที่ 1 ก็จะได้น้ำหนักคะแนนเท่ากับ  $n$  แทน เช่นในตัวอย่างด้านล่าง คะแนนของ Pikachu จะเท่ากับ  $2 + 1 + 3 + 2 = 8$

ฟังก์ชันจะคืนค่า list ของ tuple ที่ประกอบด้วยชื่อ Pokémon ทั้งหมดที่มีผู้ vote ให้ และคะแนนที่ได้ เรียงตามลำดับคะแนนและลำดับตัวอักษรในกรณีที่คะแนนเท่ากัน

#### Input

#### Output:

<pre>[['Mewtwo', 'Pikachu', 'Suicune'],  ['Mewtwo', 'Suicune', 'Pikachu'],  ['Pikachu', 'Rayquaza', 'Charizard'],  ['Suicune', 'Pikachu', 'Charizard']]</pre>	<pre>[('Pikachu', 8),  ('Mewtwo', 6),  ('Suicune', 6),  ('Charizard', 2),  ('Rayquaza', 2)]</pre>
---	---

#### • การวิเคราะห์ปัญหา

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