

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

- 2) 4 คะแนน (Lab14\_1\_6XXXXXXX.py) ให้เขียนฟังก์ชัน `remove_row_col(list_a, row, col)` เพื่อคืนค่าผลลัพธ์ที่ได้จากการลบทุก 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\_6XXXXXXX.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>

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

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

- พิจารณาเรียกใช้ฟังก์ชัน sorted()
- พิจารณาการใช้ dictionary ในการแก้ปัญหา

**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

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

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

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_____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>