กำหนดส่ง

Lab	
HW	
Until	

กระบวนวิชา 204111

การบ้านปฏิบัติการ 15

Problem Solving and Algorithm Practice (20 คะแนน)

ข้อกำหนด

i. การเรียกใช้ฟังก์ชันเพื่อการทดสอบ ต้องอยู่ภายใต้เงื่อนไข `if __name__ == '__main__' :` เพื่อความสะดวกในการ `import` จาก Script อื่น ๆ

ii. พิจารณาใช้ Iteration หรือ Recursion แก้ปัญหาที่เหมาะสม

1) **4 คะแนน** (`Lab15_1_6XXXXXXXX.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

`[[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
  - Input: จำนวนข้อมูล_____ชนิดข้อมูล_____
  - Output: (แสดงค่า) จำนวนข้อมูล_____ชนิดข้อมูล_____
    (คืนค่า) จำนวนข้อมูล_____ชนิดข้อมูล_____

2) **4 คะแนน** (`Lab15_2_6XXXXXXXX.py`) ให้เขียนฟังก์ชัน Destructive `radix_int(`*list_x, show_step=False*`)` เพื่อทำการเรียงลำดับสมาชิกจำนวนเต็มที่ไม่เป็นลบใน *list_x* จากน้อยไปมาก โดยใช้ Radix Sort Algorithm และมี Optional Parameter *show_step* เพื่อแสดง/ไม่แสดงขั้นตอนในแต่ละ Iteration

COMPUTER SCIENCE
Chiang Mai University

Function Call	Output

`list_x = \`
`[19, 48, 175, 290, 873, 7, 43, 69]`
`radix_int(list_x, True)`
`print('--------')`
`print(list_x)`
	`[290, 873, 43, 175, 7, 48, 19, 69]`
	`[7, 19, 43, 48, 69, 873, 175, 290]`
	`[7, 19, 43, 48, 69, 175, 290, 873]`
	`--------`
	`[7, 19, 43, 48, 69, 175, 290, 873]`

Function Call	Output

`list_x = \`
`[19, 48, 175, 290, 873, 7, 43, 69]`
`radix_int(list_x)`
`print('--------')`
`print(list_x)`
	`--------`
	`[7, 19, 43, 48, 69, 175, 290, 873]`

- การวิเคราะห์ปัญหา
  - Input: จำนวนข้อมูล_____ชนิดข้อมูล_____
  - Output: (แสดงค่า) จำนวนข้อมูล_____ชนิดข้อมูล_____
    (คืนค่า) จำนวนข้อมูล_____ชนิดข้อมูล_____

3) **4 คะแนน** (HW15_1_6XXXXXXXX.py) ให้เขียนฟังก์ชัน `histogram`($scores$) เพื่อแสดงผลแผนภูมิ histogram ของคะแนนรายวิชาโปรแกรมมิ่ง 101 ณ สถาบันแห่งหนึ่งทางภาคเหนือ โดยให้คำนวณความถี่จากตัวแปร $scores$ ที่อยู่ในรูป `tuple` ความยาว n ($n > 0$) ซึ่งคะแนนของนักศึกษาแต่ละคนจะเป็นจำนวนเต็มตั้งแต่ 0 - 100

ในการแจกแจงความถี่ กำหนดให้ bin size มีขนาด 10 เสมอ (ยกเว้นช่วงคะแนนสุดท้าย) โดยให้ bin แรกสุดสำหรับคะแนน 0 - 9 คะแนน และ bin ถัดไปสำหรับคะแนน 10 - 19 ดังนี้ จนไปถึง bin สุดท้ายสำหรับคะแนน 90 - 100 คะแนน (bin size ขนาด 11) ทั้งนี้ในการแสดงผลเครื่องหมาย `'*****'` หนึ่งแถวในแนวนอนจะแทนคะแนน 5 คะแนน โดยจะแสดงผลแบบปัดขึ้น ดังนั้นความถี่ที่ 48 คน จะแสดงผลด้วย `'*****'` 10 แถวเป็นต้น

Hint: นอกจากวิธี `print()` ทีละบรรทัด เราสามารถสร้าง string สำหรับแต่ละแท่งแทนช่วงความถี่แยกกันเ แล้วนำมารวมด้วย string method ต่าง ๆ ได้

Input

```
(19, 39, 59, 42, 42, 42, 100)
```

Output

```
         1           1     3     1                       1
   0   _____     0  _____ _____ _____     0     0     0  _____
 _____|*****|_____|*****|*****|*****|_____ _____ _____|*****|
0      10    20    30    40    50    60    70    80    90    100
```

Input

```
(62, 49, 75, 86, 71, 63, 74, 42, 57, 75,
 56, 58, 67, 78, 63, 73, 60, 49, 66, 77,
 47, 69, 74, 63, 65, 64, 55, 52, 52, 57,
 86, 75, 68, 70, 34, 34, 68, 46, 60, 56,
 60, 65, 66, 70, 64, 84, 61, 46, 60, 76,
 59, 64, 68, 69, 68, 47, 72, 80, 11, 44,
 53, 70, 50, 79, 81, 68, 75, 48, 62, 68)
```

Output

```
                                    26
                                  _____
                                 |*****| 16
                            11   |*****|_____
                        9  _____|*****|*****|
          1           2   _____|*****|*****|*****|  5
   0  _____    0   _____|*****|*****|*****|*****|_____   0
 _____|*****|_____|*****|*****|*****|*****|*****|*****|_____
0     10    20    30    40    50    60    70    80    90    100
```

- การวิเคราะห์ปัญหา
  - Input: จำนวนข้อมูล_____ชนิดข้อมูล_____
  - Output: (แสดงค่า) จำนวนข้อมูล_____ชนิดข้อมูล_____
    (คืนค่า) จำนวนข้อมูล_____ชนิดข้อมูล_____

4) **4 คะแนน** (HW15_2_6XXXXXXXX.py) ให้เขียนฟังก์ชัน Destructive reshape(*matrix*) เพื่อเปลี่ยนแปลงขนาดของ list สองมิติในตัวแปร *matrix* ให้มีขนาด $m \times n$ โดยกำหนดให้ m น้อยกว่าหรือเท่ากับ n เสมอ และความต่างของ m และ n จะต้องมีค่าไม่เกิน 1 ทั้งนี้ผลลัพธ์ที่ได้จะต้องมีจำนวนสมาชิกเท่ากันในทุก row และเรียงสมาชิกตามลำดับในเดิมในตัวแปร *matrix* ทีละ row และ column จากซ้ายบนไปขวาล่าง โดยสามารถเพิ่มจำนวนสมาชิกที่เป็น 0 ได้ถ้าจำเป็น โดยจำนวน element ที่มีค่า 0 ที่เพิ่มเข้าไปจะต้องมีค่าน้อยที่สุดที่เป็นไปได้

Input	**Output**
[[2, 3, 4], [1, 2, 3]]	[[2, 3, 4], [1, 2, 3]]
[[1, 2], [1, 2, 3], [1, 2], [1, 2], [1]]	[[1, 2, 1, 2], [3, 1, 2, 1], [2, 1, 0, 0]]
[[1, 2], [3, 4], [5, 6]]	[[1, 2, 3], [4, 5, 6]]

COMPUTER SCIENCE
Chiang Mai University

- การวิเคราะห์ปัญหา
  - Input: จำนวนข้อมูล_____ชนิดข้อมูล_____
  - Output: (แสดงค่า) จำนวนข้อมูล_____ชนิดข้อมูล_____
    (คืนค่า) จำนวนข้อมูล_____ชนิดข้อมูล_____

5) **4 คะแนน** (`HW15_3_6XXXXXXXX.py`) ให้เขียนฟังก์ชัน `count_vote`(*pref_matrix*) เพื่อคืนค่าคะแนนโหวตของ Pokémon แต่ละตัวที่คำนวนได้จากการลงคะแนนโหวต 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 โดยการคำนวนคะแนนจะให้น้ำหนักคะแนนที่สูงที่สุดแก่ตัวเลือกอันดับแรกเช่น 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**:

`[['Mewtwo', 'Pikachu', 'Suicune'],`
`['Mewtwo', 'Suicune', 'Pikachu'],`
`['Pikachu', 'Rayquaza', 'Charizard'],`
`['Suicune', 'Pikachu', 'Charizard']]`
	`[('Pikachu', 8),`
	`('Mewtwo', 6),`
	`('Suicune', 6),`
	`('Charizard', 2),`
	`('Rayquaza', 2)]`

- การวิเคราะห์ปัญหา
  - Input: จำนวนข้อมูล_____ชนิดข้อมูล_____
  - Output: (แสดงค่า) จำนวนข้อมูล_____ชนิดข้อมูล_____
    (คืนค่า) จำนวนข้อมูล_____ชนิดข้อมูล_____

การส่งงาน

1. ลักษณะ/ลำดับข้อความของการรับค่า/แสดงผล จะ**ต้องเป็นไปตามที่ระบุ**ในตัวอย่างการ 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/gdr111

COMPUTER SCIENCE
Chiang Mai University