

Lab	
HW	
Until	

การบ้านปฏิบัติการ 14 Sorting, Searching and File I/O (20 คะแนน)

ข้อกำหนด

- การเรียกใช้ฟังก์ชันเพื่อการทดสอบ ต้องอยู่ภายใต้เงื่อนไข if __name__ == '__main__' : เพื่อให้สามารถ
 import ไปเรียกใช้งานจาก Script อื่น ๆ ได้อย่างเป็นมาตรฐาน
- ii. ทุกข้อต้องมีการสร้างฟังก์ชัน my_id() โดยให้คืนค่าสายอักขระแทนเลขประจำตัวนักศึกษา 9 หลัก
- 1) 4 คะแนน (Lab14_1_6XXXXXXXX.py) ให้เขียนฟังก์ชัน sort_date(list_x, show_steps=False) เพื่อ เรียงลำดับสมาชิกของ list_x ด้วยวิธีเรียงลำดับแบบ Insertion Sort ซึ่งแต่ละสมาชิกจะเป็น String แทนวันที่ในปี พุทธศักราช (อาจซ้ำกันได้) ในรูปแบบ 'd/Mon/yyyy' เช่น '9/กันยายน/2565' โดยผลลัพธ์จะต้องเรียงลำดับ วันที่จากน้อยไปมาก โดยมี Optional Parameter show_steps เพื่อแสดง/ไม่แสดงขั้นตอนในแต่ละ Iteration ทั้งนี้ไม่ อนุญาตให้ import module เพิ่มเติมในการแก้ปัญหา และกำหนดให้ฟังก์ชันทำงานแบบ Destructive

 Hint: พิจารณาสร้างฟังก์ชัน Boolean less_than(date1, date2) เพื่อใช้เปรียบเทียบ String แทนวันที่

Function Call 1

```
list_x = ['11/ม.ค./2643', '5/ธ.ค./2542', '19/ม.ค./2546', '11/ก.ย./2544']
sort_date(list_x, show_steps=True)
print('---')
print(list_x)
```

Output 1

```
1: ['5/ธ.ค./2542', '11/ม.ค./2643', '19/ม.ค./2546', '11/ก.ย./2544']
2: ['5/ธ.ค./2542', '19/ม.ค./2546', '11/ม.ค./2643', '11/ก.ย./2544']
3: ['5/ธ.ค./2542', '11/ก.ย./2544', '19/ม.ค./2546', '11/ม.ค./2643']
---
['5/ธ.ค./2542', '11/ก.ย./2544', '19/ม.ค./2546', '11/ม.ค./2643']
```

Function Call 2

```
list_x = ['11/ม.ค./2643', '5/ธ.ค./2542', '19/ม.ค./2546', '11/ก.ย./2544']
sort_date(list_x)
print('---')
print(list_x)
```

Output 2

```
---
['5/ธ.ค./2542', '11/ก.ย./2544', '19/ม.ค./2546', '11/ม.ค./2643']
```

• การวิเคราะห์ปัญหา

```
    Input: จำนวนข้อมูล_____ชนิดข้อมูล______
    Output: (แสดงค่า) จำนวนข้อมูล______ชนิดข้อมูล______
    (คืนค่า) จำนวนข้อมูล______ชนิดข้อมูล______
```

2) **4 คะแนน** (Lab14_2_6xxxxxxxx.py) ให้เขียนฟังก์ชัน search_event(*list_x, key, show_steps=False*) เพื่อ<u>คืนค่า</u>ผลลัพธ์จากการคันหา tuple ของ (วันที่, เหตุการณ์) ที่มีวันที่ตรงกับ *key* โดยให้ *list_x* เป็น List ที่มี สมาชิกเป็น tuple ของ String ในรูปแบบ (วันที่, เหตุการณ์) และมีวันที่อยู่ในรูปแบบ 'd/Mon/yyyy' ที่ได้ เรียงลำดับไว้แล้วตามวันที่ในปฏิทิน

ฟังก์ชันจะต้องใช้วิธีการค้นหาแบบ Binary Search โดย input จะ<u>ไม่มี Event ที่มีวันที่ซ้ำกัน</u> โดยฟังก์ชันจะ<u>คืน</u> ค่า None หากหาไม่พบ และฟังก์ชันจะมี Optional Parameter *show_steps* เพื่อแสดง/ไม่แสดงขั้นตอนในแต่ละ Iteration ทั้งนี้ไม่อนุญาตให้ import module เพิ่มเติมในการแก้ปัญหา

Function Call 1

Output 1

```
---
('29/Feb/2032', 'Event H')
```

Function Call 2

Output 2

[4]: 9/Mar/2013 [1]: 5/Dec/2001 [2]: 5/Dec/2002 [3]: 21/Aug/2008

• การวิเคราะห์ปัญหา

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

3) 4 คะแนน (HW14_1_6XXXXXXXX.py) ให้เขียนฟังก์ชัน append_ranking(infile_name='score_in.txt', outfile_name='score_out.txt') เพื่ออ่านข้อมูลจากไฟล์ที่ระบุชื่อด้วยตัวแปร infile_name แล้ว<u>เขียนผลลัพธ์</u> ลงในไฟล์ที่มีชื่อระบุด้วยตัวแปร outfile_name โดยกำหนดให้ Content ภายในไฟล์มีลักษณะเดียวกันกับไฟล์ที่อ่าน เข้า และให้มีความเปลี่ยนแปลงเพียงอย่างเดียวคือ ให้เพิ่มอันดับ (ranking) ในรายวิชาที่นักศึกษาได้รับไว้ที่ส่วนท้าย ของแต่ละบรรทัด โดยนักศึกษาที่ได้ลำดับที่ 1 จะเป็นนักศึกษาที่ได้คะแนนเก็บรายวิชามากที่สุด

การคิดคะแนนเก็บรายวิชาจะคิดจากการสอบทั้งหมด 3 ครั้ง ที่แต่ละครั้งมีคะแนนเต็ม 100 คะแนน แล้วเลือก คะแนนสูงสุดเพียงสองครั้ง

ไฟล์น้ำเข้าจะมีรูปแบบเป็นไฟล์ n บรรทัดเมื่อน n คือจำนวนนักศึกษา และแต่ละบรรทัด จะอยู่ในรูปแบบ รหัส นักศึกษา และ คะแนนสอบทั้ง 3 ครั้งของนักศึกษาคนนั้นคั่นด้วย อักขระว่าง (' ') ดังแสดงด้านล่าง ครั้งใดที่ นักศึกษาคนนั้นขาดสอบคะแนนจะแสดงเป็น None ทั้งนี้จะไม่มีรหัสนักศึกษาที่ซ้ำในไฟล์นำเข้า และจะไม่มีคนที่ได้ คะแนนเก็บรายวิชาเท่ากัน

Hint:

• พิจารณาเรียกใช้ฟังก์ชัน sorted() และ dictionary ในการแก้ปัญหา

Input: 'score_in.txt'	Output: 'score_out.txt'
6XXX10111 53.57 59.69 55.36	6XXX10111 53.57 59.69 55.36 <mark>3</mark>
6XXX10112 64.04 None 62.84	6XXX10112 64.04 None 62.84 <mark>2</mark>
6XXX10113 88.53 91.22 84.45	6XXX10113 88.53 91.22 84.45 <mark>1</mark>

4) **4 คะแนน** (HW14_2_6XXXXXXXX.py) ให้เขียนฟังก์ชัน bottom_up_m_sort(*list_x, show_steps=False*) เพื่อทำการเรียงลำดับสมาชิกใน Integer List *list_x* ตามลำดับจากน้อยไปมากโดยใช้ Merge Sort Algorithm โดยมี Optional Parameter *show_steps* เพื่อแสดง/ไม่แสดงขั้นตอนในแต่ละ Iteration และกำหนดให้ฟังก์ชัน<u>ทำงานแบบ</u>

Destructive

Function Call

Output

```
list_x = [3, 7, 4, 9, 5, 2, 6]
bottom_up_m_sort(list_x, True)
print('-----')
print(list_x)

list_x = [3, 7, 4, 9, 5, 2, 6]
print(list_x)

list_x = [3, 7, 4, 9, 5, 2, 6, 1]
bottom_up_m_sort(list_x)
print('-----')
print(list_x)
[[3], [7], [4], [9], [5], [2], [6]]
[[3, 7], [4, 9], [2, 5], [6]]
[[3, 4, 7, 9], [2, 5, 6]]
[[2, 3, 4, 5, 6, 7, 9]]
----
[2, 3, 4, 5, 6, 7, 9]

------
[1, 2, 3, 4, 5, 6, 7, 9]
```

• การวิเคราะห์ปัญหา

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

5) **4 คะแนน** (HW14_3_6XXXXXXXX.py) ให้เขียนฟังก์ชัน radix_word(*list_x, show_steps=False*) เพื่อทำการ เรียงลำดับสมาชิกใน List ของ String *list_x* ตามลำดับคำในพจนานุกรม โดยใช้ Radix Sort Algorithm โดยมี Optional Parameter *show_steps* เพื่อแสดง/ไม่แสดงขั้นตอนในแต่ละ Iteration และกำหนดให้ฟังก์ชัน<u>ทำงานแบบ</u>

Destructive

Function Call

Output

<pre>list_x = \ ['beer', 'wine', 'vinegar', 'vodka'] radix_word(list_x, True) print('') print(list_x)</pre>	<pre>['beer', 'wine', 'vodka', 'vinegar'] ['beer', 'wine', 'vodka', 'vinegar'] ['beer', 'wine', 'vodka', 'vinegar'] ['wine', 'vinegar', 'vodka', 'beer'] ['vodka', 'beer', 'wine', 'vinegar'] ['beer', 'wine', 'vinegar', 'vodka'] ['beer', 'vinegar', 'vodka', 'wine']</pre>
	['beer', 'vinegar', 'vodka', 'wine']
<pre>list_x = \ ['beer', 'wine', 'vinegar', 'vodka'] radix_word(list_x) print('') print(list_x)</pre>	['beer', 'vinegar', 'vodka', 'wine']

• การวิเคราะห์ปัญหา

Input: จำนวนข้อมูล_____ชนิดข้อมูล______
 Output: (แสดงค่า) จำนวนข้อมูล______ชนิดข้อมูล______
 (คืนค่า) จำนวนข้อมูล________