



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

- 1) 4 คะแนน (Lab14_1_6XXXXXXX.py) ให้เขียนฟังก์ชัน `sort_date(list_x: list[str], show_steps: bool=False)` เพื่อเรียงลำดับสมาชิกของ `list_x` ด้วยวิธีเรียงลำดับแบบ Insertion Sort ซึ่งแต่ละสมาชิกจะเป็น String แทนวันที่ในปีพุทธศักราช (อาจซ้ำกันได้) ในรูปแบบ 'd/Mon/yyyy' เช่น '9/กันยายน/2565' โดยผลลัพธ์จะต้องเรียงลำดับวันที่จากน้อยไปมาก โดยมี Optional Parameter `show_steps` เพื่อแสดง/ไม่แสดงขั้นตอนในแต่ละ Iteration ทั้งนี้ไม่อนุญาตให้ 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']
```

- 2) 4 คะแนน (Lab14_2_6XXXXXXXXX.py) ให้เขียนฟังก์ชัน `search_event(list_x: list[tuple[str]], key: str, show_steps: bool=False) -> tuple[str] | None` เพื่อคืนค่าผลลัพธ์จากการค้นหา tuple ของ (วันที่, เหตุการณ์) ที่มีวันที่ตรงกับ `key` โดยให้ `list_x` เป็น List ที่มีสมาชิกเป็น tuple ของ String ในรูปแบบ (วันที่, เหตุการณ์) และมีวันที่อยู่ในรูปแบบ 'd/Mon/yyyy' ที่ได้เรียงลำดับไว้แล้วตามวันที่ในปฏิทิน ฟังก์ชันจะต้องใช้วิธีการค้นหาแบบ Binary Search โดย input จะไม่มี Event ที่มีวันที่ซ้ำกัน โดยฟังก์ชันจะคืนค่า None หากหาไม่พบ และฟังก์ชันจะมี Optional Parameter `show_steps` เพื่อแสดง/ไม่แสดงขั้นตอนในแต่ละ Iteration ทั้งนี้ไม่อนุญาตให้ import module เพิ่มเติมในการแก้ปัญหา

Function Call 1

```
list_x = [('11/Jan/1900', 'Event A'), ('5/Dec/2001', 'Event B'),
          ('5/Dec/2002', 'Event C'), ('21/Aug/2008', 'Event D'),
          ('9/Mar/2013', 'Event E'), ('11/Mar/2017', 'Event F'),
          ('7/May/2019', 'Event G'), ('29/Feb/2032', 'Event H'),
          ('9/Nov/2042', 'Event I')]
event = search_event(list_x, '29/Feb/2032')
print('---')
print(event)
```

Output 1

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

Function Call 2

```
list_x = [('11/Jan/1900', 'Event A'), ('5/Dec/2001', 'Event B'),
          ('5/Dec/2002', 'Event C'), ('21/Aug/2008', 'Event D'),
          ('9/Mar/2013', 'Event E'), ('11/Mar/2017', 'Event F'),
          ('7/May/2019', 'Event G'), ('29/Feb/2032', 'Event H'),
          ('9/Nov/2042', 'Event I')]
event = search_event(list_x, '23/Aug/2008', show_steps=True)
print('---')
print(event)
```

Output 2

```
[4]: 9/Mar/2013
[1]: 5/Dec/2001
[2]: 5/Dec/2002
[3]: 21/Aug/2008
---
None
```

3) 4 คะแนน (HW14_1_6XXXXXXXXX.py) ให้เขียนฟังก์ชัน `append_ranking(infile_name:`

`str='score_in.txt', outfile_name: str='score_out.txt')` เพื่ออ่านข้อมูลจากไฟล์ที่ระบุชื่อด้วยตัวแปร `infile_name` แล้วเขียนผลลัพธ์ลงในไฟล์ที่มีชื่อระบุด้วยตัวแปร `outfile_name` โดยกำหนดให้ Content ภายในไฟล์มีลักษณะเดียวกันกับไฟล์ที่อ่านเข้า และให้มีความเปลี่ยนแปลงเพียงอย่างเดียวคือ ให้เพิ่มอันดับ (ranking) ในรายวิชาที่นักศึกษาได้รับไว้ที่ส่วนท้ายของแต่ละบรรทัด โดยนักศึกษาที่ได้ลำดับที่ 1 จะเป็นนักศึกษาที่ได้คะแนนเก็บรายวิชามากที่สุด

การคิดคะแนนเก็บรายวิชาจะคิดจากการสอบทั้งหมด 3 ครั้ง ที่แต่ละครั้งมีคะแนนเต็ม 100 คะแนน แล้วเลือกคะแนนสูงสุดเพียงสองครั้ง

ไฟล์นำเข้าจะมีรูปแบบเป็นไฟล์ n บรรทัดเหมือน n คือจำนวนนักศึกษา และแต่ละบรรทัด จะอยู่ในรูปแบบ รหัสนักศึกษา และ คะแนนสอบทั้ง 3 ครั้งของนักศึกษาคนนั้นคั่นด้วย อักขระว่าง (' ') ดังแสดงด้านล่าง ครึ่งใดที่นักศึกษาคนนั้นขาดสอบคะแนนจะแสดงเป็น None ทั้งนี้จะไม่มีรหัสนักศึกษาที่ซ้ำในไฟล์นำเข้า และจะไม่มีคนที่ได้คะแนนเก็บรายวิชาเท่ากัน

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 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

4) 4 คะแนน (HW14_2_6XXXXXXXXX.py) ให้เขียนฟังก์ชัน `bottom_up_m_sort(list_x: list[int],`

`show_steps: bool=False) -> None` เพื่อทำการเรียงลำดับสมาชิกใน Integer List `list_x` ตามลำดับจากน้อยไปมากโดยใช้ Merge Sort Algorithm โดยมี Optional Parameter `show_steps` เพื่อแสดง/ไม่แสดงขั้นตอนในแต่ละ Iteration และกำหนดให้ฟังก์ชันทำงานแบบ **Destructive**

Function Call

Output

<pre>list_x = [3, 7, 4, 9, 5, 2, 6] bottom_up_m_sort(list_x, True) print('-----') print(list_x)</pre>	<pre>[[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]</pre>
<pre>list_x = [3, 7, 4, 9, 5, 2, 6, 1] bottom_up_m_sort(list_x) print('-----') print(list_x)</pre>	<pre>----- [1, 2, 3, 4, 5, 6, 7, 9]</pre>

- 5) 4 คะแนน (HW14_3_6XXXXXXXXX.py) ให้เขียนฟังก์ชัน `radix_word(list_x: list[str], show_steps=False)` เพื่อทำการเรียงลำดับสมาชิกใน List ของ String `list_x` ตามลำดับคำในพจนานุกรม โดยใช้ Radix Sort Algorithm โดยมี Optional Parameter `show_steps` เพื่อแสดง/ไม่แสดงขั้นตอนในแต่ละ Iteration และกำหนดให้ฟังก์ชันทำงานแบบ **Destructive** ทั้งนี้ คำที่ใช้เรียงลำดับจะเป็นตัวอักษรพิมพ์เล็กในภาษาอังกฤษเท่านั้น

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