



Lab	
HW	
Until	

## การบ้านปฏิบัติการ 11

## Sets, Dictionaries and I/O Redirection (20 คะแนน)

ข้อกำหนด

- i. การเรียกใช้ฟังก์ชันเพื่อการทดสอบ ต้องอยู่ภายใต้เงื่อนไข `if __name__ == '__main__':` เพื่อให้สามารถ `import` ไปเรียกใช้งานจาก Script อื่น ๆ ได้อย่างเป็นมาตรฐาน
  - ii. ทุกข้อต้องมีการสร้างฟังก์ชัน `my_id()` โดยให้คืนค่าสายอักขระแทนเลขประจำตัวนักศึกษา 9 หลัก
- 1) 4 คะแนน (Lab11\_1\_6XXXXXXX.py) ให้เขียนฟังก์ชัน `word_count(text)` เพื่อคืนค่า dictionary ผลลัพธ์จากการนับจำนวนคำที่ปรากฏในสายอักขระ (String) `text` โดยฟังก์ชันจะ คืนค่าเป็น dict ที่มี `key` เป็นแต่ละคำที่ปรากฏใน `text` และมี `value` เป็นความถี่ ทั้งนี้ตัวอักษรที่อยู่ใน `key` จะต้องเป็นตัวอักษรพิมพ์เล็กเท่านั้น

ข้อกำหนด

- การนับความถี่จะเป็นแบบ Case Insensitive ('ant' และ 'Ant' ถือเป็นคำเดียวกัน)
- ข้อความในไฟล์จะเป็นภาษาอังกฤษมาตรฐานในรูปแบบที่ถูกต้อง (well-formed English)
- ไม่พิจารณาเครื่องหมายวรรคตอนต่าง ๆ เฉพาะที่ล้อมรอบคำ เช่น `!"#$%&'()*+,-./:;<=>?@[\\]^_`{|}~`
- เนื่องจาก `key` มีคุณสมบัติเป็น `set` ลำดับในการแสดงผลใน output จึงไม่จำเป็นต้องเหมือนตัวอย่าง

InputOutput

"He doesn't want to pay \$40,000 for a new car, but his wife doesn't seem to care."	{ 'new': 1, 'but': 1, 'pay': 1, 'want': 1, 'seem': 1, 'care': 1, 'his': 1, '40,000': 1, 'wife': 1, 'a': 1, 'for': 1, 'car': 1, "doesn't": 2, 'to': 2, 'he': 1}
---	--

- การวิเคราะห์ปัญหา

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

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

2) **4 คะแนน** (Lab11\_2\_6XXXXXXX.py) ให้เขียนฟังก์ชัน `matching_sum(t, target_value)` เพื่อคืนค่า `list` ของจำนวนเต็มสองจำนวนใน `tuple t` ที่มีผลรวมเท่ากับจำนวนเต็ม `target_value` ทั้งนี้สมาชิกใน `tuple t` จะเป็นจำนวนเต็มเท่านั้นและจะมีสมาชิกอย่างน้อย 1 ตัวเสมอ โดยหากไม่สามารถหาจำนวนสองจำนวนดังกล่าวได้ ให้คืนค่า `list` ว่าง และหากมีสมาชิกใน `t` มากกว่า 1 คู่ ที่สามารถเป็นคำตอบได้ ให้คืนค่าเพียงคำตอบเดียวเท่านั้น

**Hint:** การจับคู่สมาชิกทั้งหมด แล้วนำมาตรวจสอบผลบวกทีละคู่ จะทำให้เวลาบน Grader เกิน ควรแก้ปัญหาโดยใช้ **`collections`** ประเภท `set` หรือ `dict`

<u>Function Call</u>	<u>Output</u>
matching_sum((1,), 1)	[]
matching_sum((5, 2), 7)	[5, 2]
	[2, 5]
matching_sum((10, -1, 1, -8, 3, 1), 2)	[10, -8]
	[-8, 10]
	[-1, 3]
	[1, 1]
matching_sum((10, -1, 1, -8, 3, 1), 10)	[]

- การวิเคราะห์ปัญหา

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

3) 4 คะแนน (HW11\_1\_6XXXXXXX.py) ให้เขียนฟังก์ชัน `words_to_num(words)` เพื่อทำการคืนค่าจำนวนเต็มบวกที่คำนวณได้จากคำอ่านของจำนวนเต็ม `words` โดยผลลัพธ์จะมีความยาวไม่เกิน 12 หลัก

**Hint:**

- สามารถศึกษาการอ่านตัวเลขในภาษาอังกฤษได้จาก [http://en.wikipedia.org/wiki/English\\_numerals](http://en.wikipedia.org/wiki/English_numerals)
- สามารถใช้ฟังก์ชัน `num_to_word()` ที่เคยเขียนไว้ในสัปดาห์ก่อนๆ ในการทดสอบเช่น `assert word_to_num(num_to_word(5)) == 5`

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

A diagram showing a road layout. On the right, three buses are parked in a line, labeled 1, 2, and 3 from left to right. Bus 1 is yellow and blue, bus 2 is blue, and bus 3 is red and yellow. The road starts as a single path from the right, then splits into two parallel paths that form a loop, and finally merges back into a single path that ends in an arrow pointing to the left.

### Output

3	['1>2>3', '1>3>2', '2>1>3', '2>3>1', '3>1>2']
4	['1>2>3>4', '1>2>4>3', '1>3>2>4', '1>3>4>2', '1>4>2>3', '2>1>3>4', '2>1>4>3', '2>3>1>4', '2>3>4>1', '2>4>1>3', '3>1>2>4', '3>1>4>2', '3>4>1>2', '4>1>2>3']

- 5) 4 คะแนน (HW11\_3\_5XXXXXXX.py) ให้เขียนฟังก์ชัน runner\_up() เพื่อรับค่าคะแนนของนักศึกษาในห้องตามจำนวนนักศึกษาที่ระบุในบรรทัดแรก แล้วแสดงผลคะแนนที่สูงเป็นอันดับที่ 1, อันดับที่ 2 และ ค่าเฉลี่ยคะแนน (ทศนิยม 2 ตำแหน่ง) โดยหากไม่มีตำแหน่งที่ 2 ให้ แสดงคำว่า **None**

**ข้อกำหนด:** ไม่อนุญาตให้ใช้ตัวแปรประเภท **iterables** เช่น **list** หรือ **tuple** ในการเก็บข้อมูลที่ได้รับเพื่อการคำนวณ

**Hint:** ศึกษา Slide Input/Output Redirection เพื่อความสะดวกในการทดสอบข้อมูลนำเข้าจำนวนมาก

#### ตัวอย่างการ run 1

```
Total students: 3
Enter score:
13
12
13
---
Max score is: 13.00
Runner up is: 12.00
Average is: 12.67
```

#### ตัวอย่างการ run 2

```
Total students: 2
Enter score:
61
61
---
Max score is: 61.00
Runner up is: None
Average is: 61.00
```

#### ตัวอย่างการ run 3

```
Total students: 7
Enter score:
61
72
64
81
61
79
63
---
Max score is: 81.00
Runner up is: 79.00
Average is: 68.71
```

#### ตัวอย่างการ run 4

```
Total students: 1
Enter score:
13
---
Max score is: 13.00
Runner up is: None
Average is: 13.00
```

- 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/gdr111>