



## การบ้านปฏิบัติการ 10

## 1D Lists and Tuples (20 คะแนน)

## ข้อกำหนด

- i. การเรียกใช้ฟังก์ชันเพื่อการทดสอบ ต้องอยู่ภายใต้เงื่อนไข `if __name__ == '__main__':` เพื่อความสะดวกในการ import จาก Script อื่นๆ
- ii. ไม่อนุญาตให้ใช้ **recursion, set, dict** หรือเรื่องอื่นๆ ที่ยังไม่ได้กล่าวถึงในรายวิชาในการแก้ปัญหา
- iii. นักศึกษาสามารถสร้างฟังก์ชันย่อยต่างๆ เพิ่มเติมได้ตามความเหมาะสม

1) **4 คะแนน** (Lab10\_1\_6XXXXXXX.py) ให้เขียนฟังก์ชัน Boolean (ฟังก์ชันที่คืนค่า **True** หรือ **False** เท่านั้น) `is_anagram(str1, str2)` เพื่อตรวจสอบว่า `str1` และ `str2` เป็น Anagram (อะนาแกรม) ซึ่งกันและกันหรือไม่ โดยการพิจารณาจะเป็นแบบ case-insensitive และ ไม่พิจารณาเครื่องหมายวรรคตอน สัญลักษณ์พิเศษ ตัวเลข หรือ อักขระว่างต่าง ๆ

คำสลับอักษร หรือ Anagram คือข้อความเกิดจากการนำตัวอักษรในอีกข้อความหนึ่งมาเรียงสลับที่กัน เช่น "Tom Marvolo Riddle" เป็น Anagram ของ "I am Lord Voldemort" หรือ "Eleven plus two" เป็น Anagram ของ "Twelve plus one"

<u>Input</u>	<u>Output</u>
Anagram Nag a ram	True
I am Lord Voldemort Tom Marvolo Riddle	True
Eggse Geese	False
Nissan Insane	False

- การวิเคราะห์ปัญหา

• Input: จำนวนข้อมูล \_\_\_\_\_ ชนิดข้อมูล \_\_\_\_\_

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



4) **4 คะแนน** (Lab10\_4\_6XXXXXXXX.py) ให้เขียนฟังก์ชัน `dest_rotate_list(list_a, n)` เพื่อทำการหมุน (Rotate) `list_a` ไปทางขวา `n` ตำแหน่ง (หรือทางซ้ายหาก `n` เป็นลบ) โดยฟังก์ชันจะทำงานแบบ **Destructive** (ไม่มีการคืนค่า และเปลี่ยนแปลง List เดิมให้เป็น List ที่ rotate แล้ว)

ข้อกำหนด: ไม่อนุญาตให้เรียกใช้ฟังก์ชัน `dest_rotate_list()` จาก Lab09\_3 หรือสร้าง List ชั่วคราวที่มีขนาดเกิน 2 element ขึ้นมาใช้พักข้อมูลในขณะดำเนินการ

<u>Input</u>	<u>Output</u>
[1, 2, 3, 4] 1	[4, 1, 2, 3]
[1, 2, 3, 4] 105	[4, 1, 2, 3]
[1, 2, 3, 4] -1	[2, 3, 4, 1]

- การวิเคราะห์ปัญหา

● Input:                      จำนวนข้อมูล \_\_\_\_\_ ชนิดข้อมูล \_\_\_\_\_

●Output: (แสดงค่า) จำนวนข้อมูล\_\_\_\_\_ชนิดข้อมูล\_\_\_\_\_

(คืบค่า)      จำนวนข้อมูล \_\_\_\_\_ ชนิดข้อมูล \_\_\_\_\_

5) **5 คะแนน** (Lab10\_5\_6XXXXXXX.py) ให้เขียนฟังก์ชันเพื่อทำการคืนค่าคำอ่านในภาษาอังกฤษของจำนวนเต็มตามข้อกำหนดดังนี้

a. **2 คะแนน** ให้เขียนฟังก์ชัน `three_digits_to_word(n)` เพื่อทำการคืนค่าคำอ่านในภาษาอังกฤษของจำนวนเต็ม  $n$  ( $0 < n \leq 999$ )

b. **3 คะแนน** ให้เขียนฟังก์ชัน `num_to_word(num)` เพื่อทำการคืนค่าคำอ่านของจำนวนเต็ม `num` ที่ไม่เป็นลบ ความยาวไม่เกิน 12 หลัก โดยจะต้องเรียกใช้ฟังก์ชัน `three digits to word()` จากข้อ a.

**Hint:**

- สามารถศึกษาการอ่านตัวเลขในภาษาอังกฤษได้จาก [http://en.wikipedia.org/wiki/English\\_numerals](http://en.wikipedia.org/wiki/English_numerals)
- พิจารณาเรียกใช้ฟังก์ชัน `divmod()`
- พิจารณาการใช้ `list` ในการแปลงตัวเลขให้เป็นคำอ่าน ดังแสดงด้านล่าง

```
>>> unit_list = ["", "one", "two", "three", "four", "five",
                  "six", "seven", "eight", "nine", "ten",
                  "eleven", "twelve", "thirteen", "fourteen", "fifteen",
                  "sixteen", "seventeen", "eighteen", "nineteen"]
>>> unit_list[17]
'seventeen'
```

