



Lab	
HW	
Until	

การบ้านปฏิบัติการ 6

1D Lists and Tuples Part I (20 คะแนน)

ข้อกำหนด

- การเรียกใช้ฟังก์ชันเพื่อการทดสอบ ต้องอยู่ภายใต้เงื่อนไข `if __name__ == '__main__':` เพื่อให้สามารถ import ไปเรียกใช้งานจาก Script อื่น ๆ ได้อย่างเป็นมาตรฐาน
 - ไม่อนุญาตให้ใช้การทำซ้ำเช่น `for`, `while` (Iterations), Recursions, หรือ Data Type อื่น ๆ ที่ยังไม่สอนในบทเรียน เช่น `set` หรือ `dict` ในการแก้ปัญหา
 - ควรสร้างฟังก์ชันทดสอบทุกข้อ และพิจารณาสร้างฟังก์ชันย่อยต่าง ๆ เพิ่มเติมได้ตามความเหมาะสม
 - ระบบ grader จะไม่ตรวจให้คะแนนฟังก์ชัน `main()` และจะพิจารณาทดสอบเฉพาะฟังก์ชันที่ระบุชื่อในแต่ละโจทย์แต่ละข้อโดยตรง
 - ในข้อที่ระบุว่ามี **[Attachments]** ให้ Download ไฟล์ Template จาก Grader ลงมา implement
- 1) 4 คะแนน (Lab06_1_6XXXXXXX.py) **[Attachments]** ให้เขียนฟังก์ชัน `triangle(n: int) -> str` ($n \geq 3$) เพื่อคืนค่า String แทนรูปสามเหลี่ยมดังแสดงในรูปด้านล่าง โดยจะต้องจบแต่ละบรรทัดด้วย newline character (`'\n'`)

InputOutput

4	<pre> * * * * . * * * * *</pre>
7	<pre> * * * * . * * . . . * * * * * * *</pre>

- 2) 4 คะแนน (Lab06_2_6XXXXXXXXX.py) ให้เขียนฟังก์ชัน `dest_rotate_list(list_a: list[int], n: int) -> None` เพื่อทำการหมุน (Rotate) `list_a` ในทิศทางและตำแหน่งที่ระบุด้วยจำนวนเต็ม `n` โดยหาก `n` เป็นบวก จะต้องหมุน `list_a` ไปทางขวา `n` ตำแหน่ง หรือหาก `n` เป็นลบ ให้หมุน `list_a` ไปทางซ้ายทางซ้ายหาก `n` ตำแหน่ง โดยฟังก์ชันจะทำงานแบบ **Destructive** (ไม่มีการคืนค่า และเปลี่ยนแปลง List เดิมให้เป็น List ที่ rotate แล้ว)

Input	Output
[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]

- 3) 4 คะแนน (HW06_1_6XXXXXXXXX.py) ให้เขียนฟังก์ชัน `uniform(line: str) -> str` เพื่อคืนค่าผลลัพธ์จากการแปลง String `line` ให้อยู่ในรูปตัวพิมพ์ใหญ่หรือพิมพ์เล็กทั้งหมด ขึ้นอยู่กับว่าชนิดตัวอักษรแบบใดมีมากกว่า หากมีจำนวนเท่ากัน ให้ยึดชนิดของตัวอักษรตัวแรกในคำเป็นเกณฑ์ แล้วแปลงอักษรทั้งหมดให้เป็นชนิดนั้น โดยพิจารณานับเฉพาะอักขระที่เป็นตัวอักษรในภาษาอังกฤษ (a-z และ A-Z)

Input	Output
HaPpY	HAPPY
cOdINg	coding
coMP scI!!!	comp sci!!!

- 4) 4 คะแนน (HW06_2_6XXXXXXXXX.py) [Attachments] ก่อต้องการส่งข้อความลับหาดาว เพื่อป้องกันไม่ให้ตัวรู้ ก่อจึงเลือกส่งข้อความด้วยการเข้ารหัสอย่างง่าย ทุก ๆ เซ้าก่อกจะส่งตารางรหัส (code table) ให้ดาว โดย ตารางรหัสจะเป็นตัวอักษรยาว `n` ตัว หลังจากนั้นก่อกจะส่งข้อความเป็นตัวเลขหาดาวทีละ 1 บรรทัด โดยตัวเลขแต่ละตัว จะเป็น index ของตัวอักษรในตารางรหัส
- เพื่อช่วยดาวถอดรหัสข้อความจากก่อก ให้ศึกษาการทำงานของฟังก์ชัน `decode(code_table, text)` (สังเกตการเรียกใช้ผ่าน lambda) และเขียนฟังก์ชัน `decode_helper(code_table, str_index)` เพื่อให้ฟังก์ชัน `decode()` คืนค่า string ผลลัพธ์ที่ได้จากการถอดรหัสสำเร็จ โดยใช้ตาราง `code_table` ที่อยู่ในรูปของสายอักขระ และถอดรหัสข้อความ `text` ที่ประกอบด้วยข้อความที่ก่อกส่งมาในลักษณะ สายอักขระของตัวเลขคั่นด้วยอักขระ space ซึ่งอาจมีมากกว่า 1 บรรทัด และแต่ละบรรทัดจะจบด้วยเครื่องหมายจุด ('.') ทั้งนี้หากตัวเลขที่ส่งมาไม่สามารถแสดงผลได้ (เช่นกรณีอยู่นอกขอบเขตที่แสดงผลได้ของตารางรหัส) ให้แสดงเป็นอักขระ underscore แทน '_'

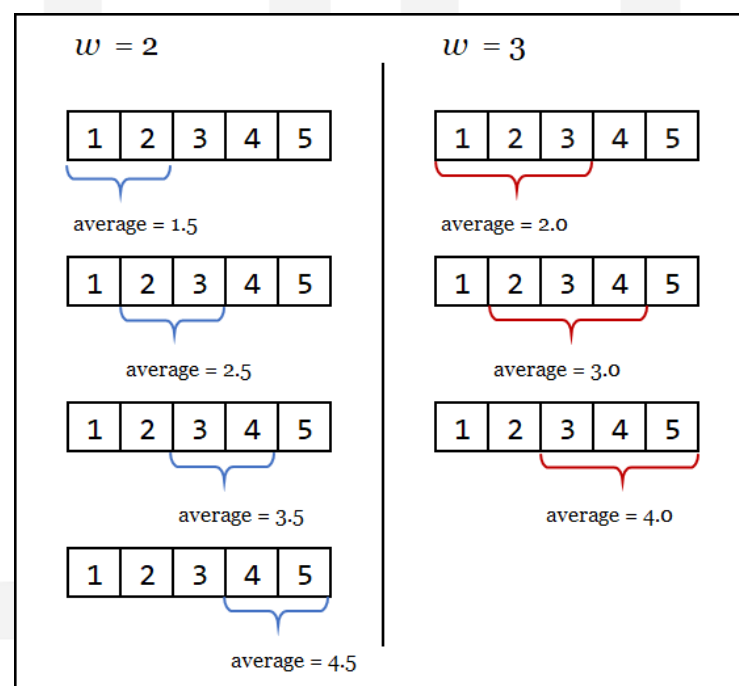
Function Call

```
decode("aceiklmr-",'''
3 .
5 3 4 2 .
3 1 2 8 1 7 20 86 .
''')
```

Output

```
i
like
ice-crea
```

- 5) 4 คะแนน (HW06_3_6XXXXXXXXX.py) ให้เขียนฟังก์ชัน `moving_average(list_x: list[float], win_size: int) -> list[float]` เพื่อคืนค่า List ของค่าเฉลี่ยเคลื่อนที่ (Moving Average) ของ element ใน `list_a` ซึ่งเป็น List ของจำนวนจริง โดยคำนวณตาม window size ที่กำหนดด้วยจำนวนเต็มบวก w ดังรูปด้านล่าง และหากไม่สามารถคำนวณ Moving Average ด้วย w ที่ระบุได้ ให้คืนค่า `[]`

**Input**

```
[1, 2, 3, 4, 5]
2
```

```
[1, 2, 3, 4, 5]
3
```

Output

```
[1.5, 2.5, 3.5, 4.5]
```

```
[2.0, 3.0, 4.0]
```

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>