



Lab	
HW	
Until	

การบ้านปฏิบัติการ 7

1D Lists and Tuples Part II (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

- 4 คะแนน (Lab07_1_6XXXXXXX.py) ให้เขียนฟังก์ชัน `corner_frame(n: int) -> str` ($n \geq 2$) เพื่อคืนค่า String แทนรูปสี่เหลี่ยมผืนผ้าแสดงในรูปด้านล่าง โดยระหว่างตัวเลขจะต้องคั่นด้วย อักขระ space (' ') หนึ่งตัวเสมอและจะต้องจบแต่ละบรรทัด (รวมถึงบรรทัดสุดท้าย) ด้วย newline character ('\n')

Input	Output
4	<pre> 1 2 3 4 2 2 3 4 3 3 3 4 4 4 4 4 </pre>
7	<pre> 1 2 3 4 5 6 7 2 2 3 4 5 6 7 3 3 3 4 5 6 7 4 4 4 4 5 6 7 5 5 5 5 5 6 7 6 6 6 6 6 6 7 7 7 7 7 7 7 7 </pre>

- 4 คะแนน (Lab07_2_6XXXXXXX.py) ให้เขียนฟังก์ชัน `square_frame(n: int, sep: str=' ') -> str` ($n \geq 3$) เพื่อแสดงผลกรอบสี่เหลี่ยมจัตุรัสแสดงด้านล่าง โดยจะต้องมีการ zero-padding ให้ตัวเลขมีความยาวหลักเท่ากันเสมอและ user สามารถระบุอักขระที่ใช้คั่น (sep) ระหว่างตัวเลขได้

Function Call	Output
<code>square_frame(3)</code>	<pre> 1 2 3 8 4 7 6 5 </pre>

<code>square_frame(4, '.')</code>	01.02.03.04 12.....05 11.....06 10.09.08.07
-----------------------------------	--

- 3) 4 คะแนน (HW07_1_6XXXXXXX.py) ให้เขียนฟังก์ชัน `print_polynomial(pc_list: list[tuple[int, float]], v: str) -> str` เพื่อคืนค่าสายอักขระ แทนพหุนามตัวแปรเดียว ตัวอย่างเช่น

$$6x^2 + 34x - 8$$

จะถูกแสดงในรูป

$$6x^2 + 34x - 8$$

โดย `pc_list` จะเป็น list ของ tuple ที่อยู่ในรูป (power, coefficient) หรือ (กำลัง, สัมประสิทธิ์) ทั้งนี้ power จะเป็นจำนวนเต็มที่ไม่เป็นลบและไม่มีค่าซ้ำกันในแต่ละพหุนาม ในขณะที่ coefficient เป็นจำนวนจริง และ `v` คืออักขระที่ใช้แทนตัวแปรในพหุนาม โดยให้แสดงสัมประสิทธิ์ให้สั้นที่สุดที่เป็นไปได้ (พิจารณาศึกษาการใช้ `%g` ในการ format)

Hint: พิจารณาการใช้ฟังก์ชัน `sorted()` เพื่อเรียงลำดับ tuple ภายใน list

Input	Output
<code>[(2, -6), (0, -8), (1, 34)]</code> 'x'	'-6x^2 + 34x - 8'
<code>[(2, -6), (0, -8), (1, 34)]</code> 'y'	'-6y^2 + 34y - 8'

- 4) 4 คะแนน (HW07_2_6XXXXXXX.py) ให้เขียนฟังก์ชัน `medal_allocation(list_a: list[int]) -> tuple[list[int], list[int], list[int]]` เพื่อคืนค่าการกำหนดเหรียญรางวัลในรูป Tuple of Lists ให้กับการแข่งขันเขียนโปรแกรม “โค้ดดิ้งโอลิมปิค” (Coding Olympiad) โดยพิจารณาจากคะแนนของผู้เข้าแข่งขันใน `list_a` ที่เป็น List ของจำนวนเต็มตั้งแต่ศูนย์ขึ้นไป โดยจะแบ่งประเภทของเหรียญเป็น Gold, Silver และ Bronze เหรียญละ 1 รางวัล ทั้งนี้ในแต่ละ List ที่คืนค่าจะเป็นคะแนนของผู้เข้าแข่งขันที่ได้เหรียญ เรียงตามลำดับจาก Gold, Silver และ Bronze และเป็น List ว่างหากไม่มีผู้ได้เหรียญในประเภทนั้น ๆ

กรณีมีผู้เข้าแข่งขันคะแนนเท่ากันและเข้าข่ายได้เหรียญรางวัล ก็จะได้รับเหรียญทุกคนและหักจากโควตาเหรียญประเภทที่ต่ำกว่าแทน เช่น หากมีผู้ได้คะแนนสูงสุดเท่ากันสองคน ก็จะแจก 2 เหรียญทอง และไม่แจกเหรียญเงิน แต่จะข้ามไปแจกเหรียญทองแดง หรือหากมีผู้เข้าแข่งขันได้คะแนนสูงสุด 5 คน ก็จะแจกเหรียญทอง 5 เหรียญ และงดให้เหรียญประเภทอื่น ๆ ทั้งนี้ในการแข่งขันจะมีผู้เข้าแข่งขันไม่น้อยกว่า 3 คนเสมอ และจะไม่พิจารณารางวัลให้ผู้เข้าแข่งขันที่ได้ 0 คะแนน

Input	Output
<code>[9, 8, 7, 6, 5, 4, 3, 2]</code>	<code>([9], [8], [7])</code>
<code>[9, 8, 7, 7, 6, 5, 4, 3, 2]</code>	<code>([9], [8], [7, 7])</code>
<code>[9, 9, 8, 7, 6, 5, 4, 3, 2]</code>	<code>([9, 9], [], [8])</code>
<code>[9, 9, 9, 9, 8, 7, 6, 5, 4, 3, 2]</code>	<code>([9, 9, 9, 9], [], [])</code>

5) 4 คะแนน (HW07_3_6XXXXXXXXX.py) ให้เขียนฟังก์ชันเพื่อคืนค่าคำอ่านในภาษาอังกฤษของจำนวนเต็มตามข้อกำหนดดังนี้

- 2 คะแนน ให้เขียนฟังก์ชัน `three_digits_to_word(n: int) -> str` เพื่อทำการคืนค่าคำอ่านในภาษาอังกฤษของจำนวนเต็ม n ($0 < n \leq 999$)
- 2 คะแนน ให้เขียนฟังก์ชัน `num_to_word(num: int) -> str` เพื่อทำการคืนค่าคำอ่านของจำนวนเต็ม num ($0 \leq num \leq 999999999999$) ความยาวไม่เกิน 12 หลัก โดยจะต้องเรียกใช้ฟังก์ชัน `three_digits_to_word()` จากข้อ a.

Hint:

- สามารถศึกษาการอ่านตัวเลขในภาษาอังกฤษได้จาก http://en.wikipedia.org/wiki/English_numerals
- พิจารณาเรียกใช้ฟังก์ชัน `divmod()`
- พิจารณาการใช้ **list** ในการแปลงตัวเลขให้เป็นคำอ่าน ดังแสดงด้านล่าง
- ตรวจสอบคำตอบได้จาก <https://www.calculatorsoup.com/calculators/conversions/numberstowords.php>

```
>>> 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'
```

Input	Output
14	fourteen
248	two hundred forty-eight
111	one hundred eleven
0	zero
42641323862	forty-two billion six hundred forty-one million three hundred twenty-three thousand eight hundred sixty-two

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>