



HW+Lab	
D+2	

## การบ้านปฏิบัติการ 10

### Iterations - Part I (20 คะแนน)

#### ข้อกำหนด

- การเรียกใช้ฟังก์ชันเพื่อการทดสอบ ต้องอยู่ภายใต้เงื่อนไข `if __name__ == '__main__':` เพื่อความสะดวกในการ import จาก Script อื่น ๆ
- ไม่อนุญาตให้ใช้ Data Type อื่น ๆ ที่ยังไม่สอนในบทเรียน เช่น `dict` หรือ `set` ในการแก้ปัญหา
- นักศึกษาสามารถสร้างฟังก์ชันย่อยต่าง ๆ เพิ่มเติมได้ตามความเหมาะสม
- ในข้อที่ระบุว่ามี **[Attachments]** ให้ Download ไฟล์ Template จาก Grader ลงมา implement

**Hint:** การใช้ Statement **assert** เพื่อทำการทดสอบฟังก์ชันที่เขียนกับข้อมูลทดสอบหลายๆ ชุดโดยอัตโนมัติ

- 1) 4 คะแนน (Lab10\_1\_6XXXXXXX.py) ก่อต้องการส่งข้อความลับหาดาว เพื่อป้องกันไม่ให้ตัวรู้ก็้อยจึงเลือกส่งข้อความด้วยการเข้ารหัสอย่างง่าย ทุก ๆ เข้าก็้อยจะส่งตารางรหัส (code table) ให้ดาว โดยตารางรหัสจะเป็นตัวอักษรยาว  $n$  ตัว หลังจากนั้นก็้อยจะส่งข้อความเป็นตัวเลขหาดาวทีละ 1 บรรทัด โดยตัวเลขแต่ละตัว จะเป็น index ของตัวอักษรในตารางรหัส

ให้เขียนฟังก์ชัน `decode(code_table, text)` เพื่อช่วยดาวถอดรหัสข้อความจากก็้อยและแสดงผล โดยใช้ตาราง `code_table` ที่อยู่ในรูปของสายอักขระ และถอดรหัสข้อความ `text` ที่ประกอบด้วยข้อความที่ก็้อยส่งมาในลักษณะ สายอักขระของตัวเลขคั่นด้วยอักขระ space ซึ่งอาจมีมากกว่า 1 บรรทัด ทั้งนี้หากตัวเลขที่ส่งมาไม่สามารถแสดงผลได้ (เช่นกรณีอยู่นอกขอบเขตของตารางรหัส) ให้แสดงเป็นอักขระ underscore แทน '\_'

#### Function Call

#### Output

<code>decode("aceiklmr-", ''' 3 5 3 4 2 3 1 2 8 1 7 2 0 86 ''')</code>	<code>i like ice-crea_</code>
--	---------------------------------------

#### การวิเคราะห์ปัญหา

- Input: จำนวนข้อมูล \_\_\_\_\_ ชนิดข้อมูล \_\_\_\_\_
- Output: (แสดงค่า) จำนวนข้อมูล \_\_\_\_\_ ชนิดข้อมูล \_\_\_\_\_  
(คืนค่า) จำนวนข้อมูล \_\_\_\_\_ ชนิดข้อมูล \_\_\_\_\_

**Hint:** พิจารณาตรวจสอบคำตอบได้ที่ <https://www.wolframalpha.com/>

<u>Input</u>	<u>Output</u>
44.1875 2	101100.001100
0.99999999 2	0.111111
-3.1415 3	-10.010211
0.9375 16	0.F00000

- Input:                      จำนวนข้อมูล \_\_\_\_\_ ชนิดข้อมูล \_\_\_\_\_
- Output:        (แสดงค่า)     จำนวนข้อมูล \_\_\_\_\_ ชนิดข้อมูล \_\_\_\_\_
- (คืนค่า)        จำนวนข้อมูล \_\_\_\_\_ ชนิดข้อมูล \_\_\_\_\_

การหา gcd (ตัวหารร่วมมาก) ด้วยวิธีการหาตัวประกอบเฉพาะร่วม ทำได้โดยการนำตัวเลขทั้งสอง มาหา prime factor (ตัวประกอบเฉพาะ) ก่อน เช่น

$$\begin{array}{l} 180 = \underline{2} \times \underline{2} \times 3 \times \underline{3} \times 5 \\ 48 = \underline{2} \times \underline{2} \times 2 \times 2 \times 3 \end{array}$$

**Hint:** พิจารณาการใช้ Merge Algorithm

Function Call	Output
<pre>x = common_prime_factor(180, 48) print('-----') print(x)</pre>	<pre>----- [2, 2, 3]</pre>
<pre>x = common_prime_factor(180, 48, True) print('-----') print(x)</pre>	<pre>a: [2, 2, 3, 3, 5] b: [2, 2, 2, 2, 3] ----- [2, 2, 3]</pre>

- การวิเคราะห์ปัญหา

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

4) 4 คะแนน (HW10\_2\_6XXXXXXXXX.py) ให้เขียนฟังก์ชัน `eratosthenes(n, show_step=False)` เพื่อคืนค่า `list` ของจำนวนเฉพาะตั้งแต่ 2 ถึง จำนวนเต็มบวก  $n$  ตามวิธี 'Sieve of Eratosthenes' ในบทเรียนโดยมี Optional Parameter `show_step` เพื่อแสดง/ไม่แสดงขั้นตอนในแต่ละ Iteration ดังแสดงด้านล่าง

<u>Function Call</u>	<u>Output</u>
<pre>result = eratosthenes(20, True) print('----') print(result)</pre>	<pre>2: [2, 3, 5, 7, 9, 11, 13, 15, 17, 19] 3: [2, 3, 5, 7, 11, 13, 17, 19] ---- [2, 3, 5, 7, 11, 13, 17, 19]</pre>
<pre>result = eratosthenes(20) print('----') print(result)</pre>	<pre>---- [2, 3, 5, 7, 11, 13, 17, 19]</pre>

- การวิเคราะห์ปัญหา

- Input:                      จำนวนข้อมูล \_\_\_\_\_ ชนิดข้อมูล \_\_\_\_\_
- Output:        (แสดงค่า)     จำนวนข้อมูล \_\_\_\_\_ ชนิดข้อมูล \_\_\_\_\_  
                                   (คืนค่า)          จำนวนข้อมูล \_\_\_\_\_ ชนิดข้อมูล \_\_\_\_\_

- 5) 4 คะแนน (HW10\_3\_6XXXXXXX.py) ให้เขียนฟังก์ชัน `polynomial_addition(p1, p2)` เพื่อทำการบวกพหุนามตัวแปรเดียว 2 จำนวนตัวอย่างเช่น

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

$$b = -6x^2 + x + 2$$

จะได้ผลลัพธ์  $a + b = 35x - 6$  หรือ `[(1, 35), (0, -6)]` โดย  $p1$  และ  $p2$  จะเป็น list ของ tuple ที่อยู่ในรูป (กำลัง, สัมประสิทธิ์) หรือ (power, coefficient) โดย power จะเป็นจำนวนเต็มที่ไม่มากกว่าหรือเท่ากับ 0 ในขณะที่ coefficient เป็นจำนวนจริง ทั้งนี้ฟังก์ชันจะต้องทำงานแบบ Non-destructive และคืนค่า list ผลลัพธ์เรียงตามกำลังจากมากไปน้อย

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

Input	Output
<code>[(2, 6), (1, 34), (0, -8)]</code> <code>[(2, -6), (0, 2), (1, 1)]</code>	<code>[(1, 35), (0, -6)]</code>

• การวิเคราะห์ปัญหา

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

COMPUTER SCIENCE  
Chiang Mai University