

Part I: Basic Python Programming

100 คะแนน

File Header (กรณีไม่เขียน Header จะเสียคะแนน 5%)

```
#!/usr/bin/env python3
# ชื่อ (ไม่ต้องใส่นามสกุล)
# รหัสสนศ
# Sec00x
```

คำชี้แจง: ให้ถือว่า input ที่เข้ามาจะอยู่ใน Range ที่โจทย์กำหนดเสมอ (ไม่ต้องพยายามแก้ปัญหา input ที่อยู่นอก Range ดังกล่าว)

- 1) **20 คะแนน** (m1p1_6xxxxxxx.py) **[Attachment]** ให้เขียนฟังก์ชัน `is_isosceles(a, b, c)` เพื่อคืนค่าผลลัพธ์การตรวจสอบว่าสามเหลี่ยมที่มีความยาวด้านระบุด้วยจำนวนเต็มบวก a b และ c เป็นสามเหลี่ยมหน้าจั่ว (isosceles) หรือไม่ โดยนิยามของสามเหลี่ยมหน้าจั่วคือสามเหลี่ยมที่มีด้านที่มีความยาวเท่ากันอย่างน้อยสองด้าน หากเป็นสามเหลี่ยมหน้าจั่วฟังก์ชันจะคืนค่า Boolean **True** และหากไม่เป็นสามเหลี่ยมหน้าจั่วฟังก์ชันจะคืนค่า Boolean **False** ทั้งนี้ความยาวด้าน a b และ c จะสามารถสร้างเป็นสามเหลี่ยมได้เสมอ

Function Call

Output

<code>is_isosceles(3, 3, 3)</code>	True
<code>is_isosceles(5, 5, 4)</code>	True
<code>is_isosceles(7, 8, 9)</code>	False
<code>is_isosceles(6, 10, 12)</code>	False

- 2) **20 คะแนน** (m1p2_6xxxxxxx.py) **[Attachment]** ให้เขียนฟังก์ชัน `mask_code(PFE, color)` เพื่อคืนค่า string ข้อมูลรหัส (Code) สีน้ากากกรองอนุภาคขนาดเล็กที่จะพิมพ์ลงบนกล่อง โดยมีค่า *PFE* หรือ Particle Filtration Efficiency คือ ค่าตัวเลขประสิทธิภาพการกรองสิ่งปนเปื้อนอนุภาคขนาดเล็กของหน้ากากอนามัย และ *color* คือ สีของหน้ากากกรองอนุภาคขนาดเล็กซึ่งจะสามารถแสดงด้วยสีใดก็ได้ เช่น 'white', 'pink', 'yellow' โดยผลลัพธ์จะเริ่มแสดงตัวอักษร N และต่อด้วยค่า *PFE* แล้วจะคั่นด้วย hyphen ('-' หรือขีดกลาง) ระหว่างค่า *PFE* และ *color* โดยกำหนดให้ชื่อสีเพียงอักษร 2 ตัวแรก

Function Call

Output

<code>mask_code(95, 'white')</code>	'N95-wh'
<code>mask_code(97, 'black')</code>	'N97-bl'
<code>mask_code(90, 'brown')</code>	'N90-br'

- 3) **20 คะแนน** (m1p3_6xxxxxxx.py) **[Attachment]** ให้เขียนฟังก์ชัน `word_len(list_a)` ที่มีการทำงานแบบ Destructive เพื่อแทนที่แต่ละสมาชิกใน `list_a` ที่เป็น List ของ String ด้วยความยาวของ String ที่เป็นสมาชิกในแต่ละตำแหน่ง ทั้งนี้ `list_a` จะไม่เป็น List ว่างและมีสมาชิกเป็น String เสมอ

Hint: ฟังก์ชันจะไม่มีการทำงานคืนค่า

Function Call	Output
<pre>list_a = ['a', 'quick', 'brown', 'fox'] word_len(list_a) print(list_a)</pre>	[1, 5, 5, 3]
<pre>list_a = ['empty', 'or', '', 'full'] word_len(list_a) print(list_a)</pre>	[5, 2, 0, 4]

- 4) 20 คะแนน (m1p4_6xxxxxxx.py) **[Attachment]** ให้เขียนฟังก์ชัน `first_of_2(list_a, list_b)` เพื่อคืนค่า List ผลลัพธ์จากการเปรียบเทียบแต่ละสมาชิกของ `list_a` และ `list_b` ที่เป็น List ของ String โดยมีข้อกำหนดคือ สมาชิกของ List ผลลัพธ์ในตำแหน่งที่ n จะมีค่าเท่ากับค่าที่มาก่อนในพจนานุกรม เมื่อพิจารณาสมาชิกในตำแหน่งเดียวกันของ `list_a` และ `list_b` หรือหากค่าในตำแหน่งที่ n เป็นค่าเดียวกัน ให้ใช้ค่าจาก List ใดก็ได้ในตำแหน่งที่ n ใน List ผลลัพธ์ กำหนดให้ List ทั้งสองไม่เป็น List ว่าง และจะมีความยาวเท่ากันเสมอ โดยสมาชิกของ List ทั้งสองจะไม่เป็น String ว่างและประกอบด้วยตัวอักษรพิมพ์เล็กในภาษาอังกฤษ (a-z) เท่านั้น

Hint: ฟังก์ชันทำงานแบบ Non-destructive

Function Call	Output
<pre>first_of_2(['i', 'bought', 'me', 'a', 'boat'], ['a', 'man', 'and', 'a', 'dog'])</pre>	['a', 'bought', 'and', 'a', 'boat']

- 5) 20 คะแนน (m1p5_6xxxxxxx.py) **[Attachment]** ให้เขียนฟังก์ชัน `cumulative_sum_of_sq(list_a)` เพื่อคืนค่า List ผลลัพธ์ที่ได้จากการคำนวณโดยให้สมาชิกของ List ผลลัพธ์ในตำแหน่งที่ n มีค่าเท่ากับผลรวมของกำลังสองของสมาชิกของ `list_a` ในตำแหน่งแรกสุดจนถึงตำแหน่งที่ n เช่น หาก `list_a` คือ [1, 2, 3, 4] สมาชิกในตำแหน่ง index ที่ 1 จะมีค่าเท่ากับ $1^2 + 2^2 = 5$ ทั้งนี้ `list_a` จะไม่เป็น List ว่าง และมีสมาชิกทุกตัวเป็นจำนวนเต็ม

Hint: ฟังก์ชันทำงานแบบ Non-destructive

Input	Output
[1, 2, 3, 4]	[1, 5, 14, 30]
[3, 0, -1]	[9, 9, 10]

Python Tutor Visualizer: <http://10.10.10.11/visualize.html>

Grader: <http://10.10.10.10>