

Basic Python Programming

100 คะแนน

File Header (กรณีไม่เขียน Header จะเสียคะแนน 5%)

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

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

- 1) 15 คะแนน (m1p1_6xxxxxxx.py) [Attachment] ให้เขียนฟังก์ชัน `generate_p_triple(m: int, n: int) -> None` เพื่อแสดงผล ชุดจำนวนเต็มพีทาโกรีส (Pythagorean Triple) ที่คำนวณได้จากสูตรของยูคลิด (Euclid) เมื่อกำหนดจำนวนเต็มบวก m และ n ($m > n$)
- สูตรของยูคลิดสำหรับการสร้างชุดจำนวนเต็มพีทาโกรีส เป็นวิธีหนึ่งในการสร้างชุดของจำนวนเต็มบวกสามจำนวน (a, b, c) ที่สอดคล้องกับทฤษฎีบทพีทาโกรัส $a^2 + b^2 = c^2$ สูตรนี้มีดังนี้:

$$a = m^2 - n^2, b = 2mn, c = m^2 + n^2$$

โดยที่ m และ n เป็นจำนวนเต็มบวกใดๆ ที่ $m > n$ ทั้งนี้ฟังก์ชันจะแสดงจำนวนเต็มบวก 3 จำนวน (a, b, c) ดังกล่าวในบรรทัดเดียวกันคั่นด้วยอักขระว่าง (' ') และสามารถเลือกแสดงผลตัวเลขในลำดับใดก็ได้

Function Call

Output

<code>generate_p_triple(2, 1)</code>	3 4 5
<code>generate_p_triple(3, 2)</code>	5 12 13

- 2) 15 คะแนน (m1p2_6xxxxxxx.py) [Attachment] ปี 2024 เป็นปีนักษัตรมังกรไม้ (Wood Dragon) ตามหลักโหราศาสตร์จีน ชื่อที่เป็นมงคลสำหรับเด็กที่เกิดในปีนี้ จะต้องขึ้นต้นหรือลงท้ายด้วยสระในภาษาอังกฤษ (aeiou) อย่างใดอย่างหนึ่งเท่านั้น (ตำแหน่งอื่นๆ ในชื่อจะมีสระหรือไม่ก็ได้)
- ให้เขียนฟังก์ชัน Boolean `Lucky_name(name: str) -> bool` เพื่อคืนค่าผลลัพธ์การตรวจสอบว่า String `name` เป็นชื่อมงคลหรือไม่ โดยให้คืนค่า True ถ้าเป็นชื่อมงคล และ False ถ้าไม่ใช่ชื่อมงคล ทั้งนี้ `name` จะประกอบด้วยตัวอักษรในภาษาอังกฤษ [A-Za-z] เท่านั้น

Input

Output

Eric	True
Tony	False
Bruce	True
Eva	False

- 3) 15 คะแนน (m1p3_6xxxxxxx.py) [Attachment] ตัวเลขมงคลสำหรับปี 2024 ที่เป็นปีนักษัตรมังกรไม้ (Wood Dragon) จะต้องเป็นเลขจำนวนเต็มบวกที่หารด้วย 7 ลงตัวแต่ไม่มี 7 ในหลักใดๆ
- ให้เขียนฟังก์ชัน `count_Lucky_num(start: int, stop: int) -> int` เพื่อคืนค่าผลลัพธ์การนับจำนวนตัวเลขมงคลในช่วง `[start, stop)` ($0 < start \leq stop$) เช่นในช่วง 1 – 15 จะมีแค่ 14 ที่เข้าเงื่อนไข

Function Call

Output

<code>count_lucky_num(1, 14)</code>	0
<code>count_lucky_num(1, 15)</code>	1
<code>count_lucky_num(14, 22)</code>	2

- 4) 15 คะแนน (m1p4_6xxxxxxx.py) [Attachment] ให้เขียนฟังก์ชัน `find_min(list_a: list[int]) -> int` เพื่อคืนค่าผลลัพธ์จากการหาค่าต่ำสุดของ `list_a` ที่ไม่เป็น List ว่างของจำนวนเต็มใดๆ ทั้งนี้ไม่อนุญาตให้ใช้ฟังก์ชัน `min()` `max()` หรือ `sorted()`

Hint: ฟังก์ชันทำงานแบบ Non-destructive

Input

Output

[0, 1, 2, 3]	0
[1, 0, 2, 3]	0

- 5) 20 คะแนน (m1p5_6xxxxxxx.py) [Attachment] ให้เขียนฟังก์ชัน `median(list_a: list[int]) -> float` เพื่อคืนค่าผลลัพธ์จากการหาค่ามัธยฐานของสมาชิกใน `list_a` ที่ไม่เป็น List ว่างของจำนวนเต็มใดๆ
- มัธยฐานของรายการข้อมูลขนาดจำกัด สามารถหาได้โดยการเรียงลำดับข้อมูลจากน้อยไปมาก (หรือมากไปน้อยก็ได้) แล้วถือเอาตัวเลขที่อยู่ตรงกลางเป็นค่ามัธยฐาน ถ้าหากจำนวนสิ่งที่สังเกตการณ์เป็นจำนวนคู่ ทำให้ค่าที่อยู่ตรงกลางมีสองค่า เราหาค่าเฉลี่ย (mean) ของสองจำนวนนั้นเพื่อให้ได้มัธยฐานเพียงหนึ่งเดียว
- ทั้งนี้ไม่อนุญาตให้ `import` โมดูลเพิ่มเติมนอกจากโมดูลที่สอนในบทเรียน และคำตอบที่ถูกจะต้องอยู่ภายใน `epsilon=0.001`

Hint: ฟังก์ชันทำงานแบบ Non-destructive

Input

Output

[1, 2, 3]	2
[3, 2, 1, 0]	1.5

- 6) 20 คะแนน (m1p6_6xxxxxxx.py) [Attachment] ให้เขียนฟังก์ชัน `adjacent_sum(list_a: list[int]) -> list[int]` เพื่อคืนค่า List ผลลัพธ์จากการหาผลบวกของสมาชิกที่อยู่ติดกันของ `list_a` ที่เป็น List ของจำนวนเต็มใดๆ และมีสมาชิกอย่างน้อย 2 ตัวเสมอ

Hint:

- ฟังก์ชันทำงานแบบ Non-destructive
- ความยาวของ List ผลลัพธ์จะน้อยกว่าความยาวของ `list_a` อยู่ 1 เสมอ

Input**Output**

[0, 1, 2]	[1, 3]
[2, 4, 6, 8]	[6, 10, 14]

Python Tutor Visualizer: <http://10.10.10.11/visualize.html>Grader: <http://10.10.10.10>**Access Validation**

85 คะแนน (m2p1_6XXXXXXX.py)

File Header (กรณีไม่เขียน Header จะเสียคะแนน 5%)

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

การตรวจสอบความถูกต้องของผู้มีสิทธิ์ผ่านเข้าประตูห้องแห่งความลับทางการสื่อสารแห่งหนึ่ง กำหนดให้ผู้ที่ผ่านเข้ามาต้องยืนยันข้อมูลด้วยการรับข้อความจำนวน 2 ข้อความ โดยที่ข้อความสามารถประกอบด้วยตัวอักษรในภาษาอังกฤษทั้งตัวพิมพ์ใหญ่และตัวพิมพ์เล็ก (A-Z, a-z), ตัวเลข (0-9), เครื่องหมาย space และ tab, เครื่องหมาย punctuation ได้แก่ !"#\$%^&'()*+,-./:;<?=@[\]^_`{|}~ จากนั้นจะต้องทำการตรวจสอบว่าข้อความ 2 ข้อความให้ ชุดตัวอักษรที่เหมือนกันหรือไม่ โดยไม่พิจารณาจำนวนหรือตำแหน่งที่ปรากฏ รวมทั้งไม่พิจารณาตัวพิมพ์ใหญ่หรือตัวพิมพ์เล็ก (case insensitive) หาก "ใช่" ให้คืนค่า **True** ถ้า "ไม่ใช่" ให้คืนค่า **False** ตัวอย่างเช่น

ข้อความแรก	ข้อความที่สอง	ผลการตรวจสอบ
your reader	You are ready.	เหมือนกัน (True)
Hungry (56)laundry puPpy #@&	puy dry hun	ไม่เหมือนกัน (False)
fun RuN marathon	FoUrth man	เหมือนกัน (True)
Sator,ArepoandTenet_#Opera23and0Rotas	Rota send_\$&port	เหมือนกัน (True)
Goo{d} 3M\'orning78	iron2*#35 God2&	ไม่เหมือนกัน (False)
4i2r3o{l3n}6mtG7o8d9	Goolm#tdM4498orn23ing	เหมือนกัน (True)

หน้าที่ของคุณคือให้เขียนฟังก์ชัน Boolean `same_letters(str1: str, str2: str) -> bool` เพื่อคืนค่าผลการพิจารณาความเหมือนของ ชุดตัวอักษร ในข้อความ `str1` และข้อความ `str2` หากเหมือนกัน ให้คืนค่า **True** ถ้าไม่เหมือนกัน ให้คืนค่า **False** ทั้งนี้ `str1` และ `str2` จะประกอบด้วย ตัวอักษร อย่างน้อย 1 ตัวเสมอ

Input**Output**

your reader You are ready.	True
WeICome weIdoNe	False
annA bell Balleen	True
4i2r3o{l3n}6mtG7o8d9 Goolm#tdM4498orn23ing	True
reST334Ro0m344home56 256889HBbro455om55s	False

คำอธิบาย Test Case

มีทั้งหมด 17 Test Case คิดคะแนนเป็น 5 คะแนนต่อ Test Case

- Test Case ที่ 1 ข้อความประกอบด้วยตัวอักษรตัวพิมพ์เล็กเท่านั้น และมีความยาวเท่ากัน

- Test Case ที่ 2 ข้อความประกอบด้วยตัวอักษรตัวพิมพ์เล็กและตัวพิมพ์ใหญ่และความยาวเท่ากัน
- Test Case ที่ 3-4 ข้อความประกอบด้วยตัวอักษรทั้งตัวเล็กและตัวใหญ่ และตัวเลข
- Test Case ที่ 5-6 ข้อความประกอบด้วยตัวอักษรตัวพิมพ์เล็กและตัวพิมพ์ใหญ่, space และ Tab
- Test Case ที่ 7-8 ข้อความประกอบด้วยตัวอักษรตัวพิมพ์เล็กและตัวพิมพ์ใหญ่, punctuation
- Test Case ที่ 9-10 ข้อความประกอบด้วยตัวอักษรตัวพิมพ์เล็กและตัวพิมพ์ใหญ่, ตัวเลข, space และ Tab
- Test Case ที่ 11-12 ข้อความประกอบด้วยตัวอักษรตัวพิมพ์เล็กและตัวพิมพ์ใหญ่, space, Tab และ punctuation
- Test Case ที่ 13-15 ข้อความประกอบด้วยตัวอักษรตัวพิมพ์เล็กและตัวพิมพ์ใหญ่, ตัวเลข, space, Tab และ punctuation
- Test Case ที่ 16-17 ข้อความประกอบด้วยตัวอักษรตัวพิมพ์เล็กและตัวพิมพ์ใหญ่, ตัวเลข, space, Tab และ punctuation และมีความยาว 50-150 อักขระ

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

Grader: <http://10.10.10.10>

Finding Modes (FM)

100 คะแนน (m3p1_6XXXXXXX.py)

File Header (กรณีไม่เขียน Header จะเสียคะแนน 5%)

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

การหาฐานนิยม (Finding Modes)

ฐานนิยม (Mode) คือ ค่าที่ปรากฏบ่อยครั้งที่สุด (มีความถี่มากที่สุด) ในชุดข้อมูล ในการสอบ อาจารย์อยากทราบว่า คะแนนของนักศึกษามีฐานนิยมที่ใดบ้าง เพื่อศึกษาต่อไปว่า นักศึกษาได้คะแนนมากในช่วงนี้เป็นเพราะเหตุใด ตัวอย่างเช่น ชุดคะแนน [40, 15, 40, 20, 30, 30, 40, 80, 40] จะมีค่าฐานนิยมคือ 40 เพราะปรากฏอยู่ 4 ครั้งในชุดคะแนน

ทั้งนี้ ในตัวเลขชุดหนึ่งอาจมีฐานนิยมมากกว่า 1 ค่าได้ เช่น ชุดคะแนน [30, 20, 30, 20, 20, 100, 50, 100, 100] มีค่าฐานนิยม 2 ค่า คือ 20 และ 100 แต่ละค่าปรากฏอยู่ 3 ครั้ง ในชุดคะแนน

หน้าที่ของคุณคือให้เขียนฟังก์ชัน `find_mode(score_list: list[int]) -> list[int]` เพื่อคืนค่า List ผลลัพธ์จากการคำนวณฐานนิยม โดยฟังก์ชันจะรับ `score_list` ซึ่งเป็น List ของคะแนนสอบซึ่งเป็นเลขจำนวนเต็มที่ไม่เป็น List ว่าง แล้วคืน List จำนวนเต็มซึ่งเป็นค่าฐานนิยมของคะแนนใน `score_list` เรียงลำดับจากน้อยไปหามาก ทั้งนี้ไม่อนุญาตให้ import โมดูลเพิ่มเติมนอกจากโมดูลที่สอนในบทเรียน

Function Call

Output

<code>find_mode([100, 50, 50, 80, 70])</code>	<code>[50]</code>
<code>find_mode([30, 20, 30, 20, 20, 100, 50, 100, 100])</code>	<code>[20, 100]</code>
<code>find_mode([-20, 50, 50, 80, 70, -20])</code>	<code>[-20, 50]</code>
<code>find_mode([40, 20, 30, 10])</code>	<code>[10, 20, 30, 40]</code>

คำอธิบาย Test Case

ใน Case 1 – 7 `score_list` จะมีค่าฐานนิยมเพียง 1 ค่า

1. Case 1 มีคะแนนใน `score_list` ไม่เกิน 5 ค่า คะแนนอยู่ในช่วง 0 ถึง 20
2. Case 2 มีคะแนนใน `score_list` ไม่เกิน 10 ค่า คะแนนอยู่ในช่วง 0 ถึง 100
3. Case 3 – 5 มีคะแนนใน `score_list` ไม่เกิน 1,000 ค่า คะแนนอยู่ในช่วง 0 ถึง 100
4. Case 6 มีคะแนนใน `score_list` ไม่เกิน 1,000 ค่า คะแนนอยู่ในช่วง 0 ถึง 1,000
5. Case 7 มีคะแนนใน `score_list` ไม่เกิน 1,000 ค่า คะแนนอยู่ในช่วง -1,000 ถึง 1,000
6. Case 8 – 10 มีคะแนนใน `score_list` ไม่เกิน 1,000 ค่า คะแนนอยู่ในช่วง -1,000 ถึง 1,000 และมีค่าฐานนิยมมากกว่า 1 ค่า

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

Grader: <http://10.10.10.10>