

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) **15 คะแนน** (m1p1_6xxxxxxx.py) **[Attachment]** ให้เขียนฟังก์ชัน `is_right_triangle(a, b, c)` เพื่อคืนค่าผลลัพธ์การตรวจสอบว่า สามเหลี่ยมที่มีความยาวด้านระบุด้วยจำนวนเต็มบวก a b และ c เป็นสามเหลี่ยมมุมฉากหรือไม่ ด้วยทฤษฎีของพีทาโกรัส โดยถ้าเป็นสามเหลี่ยมมุมฉากฟังก์ชันจะคืนค่า Boolean **True** และหากไม่เป็นสามเหลี่ยมมุมฉากฟังก์ชันจะคืนค่า Boolean **False** ทั้งนี้ความยาวด้าน a b และ c จะสามารถสร้างเป็นสามเหลี่ยมได้เสมอ

Function Call**Output**

<code>is_right_triangle(3, 4, 5)</code>	True
<code>is_right_triangle(5, 3, 4)</code>	True
<code>is_right_triangle(5, 12, 13)</code>	True
<code>is_right_triangle(5, 13, 13)</code>	False

- 2) **15 คะแนน** (m1p2_6xxxxxxx.py) **[Attachment]** โดยปกติแล้วรหัสประจำตัวนักศึกษาในดินแดนล้านนาจะยาว 9 - 15 หลักแตกต่างกันไปตามการกำหนดของแต่ละสถานศึกษา แต่ไม่ว่าความยาวจะเป็นเท่าไร สองหลักแรกจะเป็นปีที่เข้าศึกษา และสามหลักท้ายจะเป็นเลขสายรหัสเสมอ

ให้เขียนฟังก์ชัน `transform_id(int_id)` เพื่อคืนค่า String ผลลัพธ์การแปลงจากรหัสนักศึกษา `int_id` (จำนวนเต็มบวกระหว่าง 9 - 15 หลัก) ให้อยู่ในรูปเลขสายรหัส (3 หลักสุดท้าย) ตามด้วย '-' และ ปีที่เข้า (2 หลักแรก)

Input**Output**

650241555	555-65
62345678911	911-62

- 3) **15 คะแนน** (m1p3_6xxxxxxx.py) **[Attachment]** ให้เขียนฟังก์ชัน `count_4n5(n)` เพื่อคืนค่าผลลัพธ์การนับจำนวนตั้งแต่ 1 ถึงจำนวนเต็มบวก n ($n > 1$) ที่มีทั้ง 4 และ 5 อยู่ในหลักใด ๆ เช่นในช่วง 1 - 45 จะมีแค่ 45 ที่ตรงตามข้อกำหนดนี้

Input**Output**

45	1
99	2

- 4) **15 คะแนน** (m1p4_6xxxxxxx.py) **[Attachment]** ให้เขียนฟังก์ชันที่มีการทำงานแบบ Destructive ชื่อ `log2_list(list_a)` เพื่อแทนที่แต่ละสมาชิกใน `list_a` ด้วยจำนวนจริงแทนค่า Log ฐาน 2 ของสมาชิกนั้น ๆ ทั้งนี้ `list_a` จะไม่เป็น List ว่าง และแต่ละสมาชิกจะเป็นจำนวนจริงบวกเสมอ โดยคำตอบที่ถูกจะต้องอยู่ภายใน $\epsilon=0.001$

Hint: ฟังก์ชันจะไม่มีการคืนค่า

Input	Output
[1, 2, 4]	[0.0, 1.0, 2.0]
[0.25, 0.5]	[-2.0, -1.0]

- 5) **20 คะแนน** (m1p5_6xxxxxxx.py) **[Attachment]** ให้เขียนฟังก์ชัน `reverse_cap(list_a)` เพื่อคืนค่า List ผลลัพธ์การแปลงแต่ละสมาชิกของ `list_a` (ที่ไม่เป็น List ว่าง) ให้อยู่ในรูป reverse capitalized กล่าวคือให้มีอักษรตัวแรกเป็นตัวพิมพ์เล็ก และ ตัวที่เหลือเป็นตัวพิมพ์ใหญ่ โดยสมาชิกของ `list_a` จะเป็น String ที่ไม่เป็น String ว่าง และประกอบด้วยตัวอักษรในภาษาอังกฤษ (A-Z และ a-z) เท่านั้น

Hint: ฟังก์ชันทำงานแบบ Non-destructive

Input	Output
['I', 'bought', 'two', 'bananas']	['i', 'BOUGHT', 'tWO', 'bANANAS']
['a', 'man', 'and', 'a', 'dog']	['a', 'mAN', 'aND', 'a', 'dOG']

- 6) **20 คะแนน** (m1p5_6xxxxxxx.py) **[Attachment]** ให้เขียนฟังก์ชัน `harmonic_mean(list_a)` เพื่อคืนค่า List ผลลัพธ์การคำนวณค่าเฉลี่ยฮาร์โมนิกของสมาชิกใน `list_a` โดยที่ `list_a` ไม่เป็น List ว่างสมาชิกของ `list_a` เป็นจำนวนจริงที่ไม่เป็น 0 ทั้งนี้คำตอบที่ถูกจะต้องอยู่ภายใน $\epsilon=0.001$

ค่าเฉลี่ยฮาร์โมนิก (Harmonic Mean หรือ H.M.) ของทุกจำนวนใน `list_a` = $[x_1, x_2, x_3, \dots, x_n]$ คำนวณได้จากสูตร

$$H.M. = \frac{n}{\frac{1}{x_1} + \frac{1}{x_2} + \frac{1}{x_3} + \dots + \frac{1}{x_n}}$$

Hint: ฟังก์ชันทำงานแบบ Non-destructive

Input	Output
[1, 2, 2, 2]	1.6
[3, 3, 6]	3.6