



Lab	
HW	
Until	

### การบ้านปฏิบัติการ 3

#### Functions (20 คะแนน)

#### ข้อกำหนด

- การเรียกใช้ฟังก์ชันเพื่อการทดสอบ ต้องอยู่ภายใต้เงื่อนไข `if __name__ == '__main__':` เพื่อความสะดวกในการ import จาก Script อื่นๆ
- ไม่อนุญาตให้ใช้ Control Flow ต่าง ๆ เช่น `if` (Conditionals) หรือ `and`, `or` (Logical operations) หรือ `for`, `while` (Iterations), Recursions, หรือ Data Type อื่น ๆ ที่ยังไม่สอนในบทเรียน เช่น `range`, `list` หรือ `map` ในการแก้ปัญหา
- นักศึกษาสามารถสร้างฟังก์ชันย่อยต่าง ๆ เพิ่มเติมได้ตามความเหมาะสม
- ในข้อที่ระบุว่ามี **[Attachments]** ให้ Download ไฟล์ Template จาก Grader ลงมา implement (ปุ่ม Attachment บน Grader)
- ตัวอักษรเอียงสีน้ำเงิน ในตัวอย่างการ Run คือ User Input จาก keyboard (กรณีโจทย์กำหนดให้เขียนโปรแกรม)

**Hint:** ควรใช้ Statement `assert` เพื่อทำการทดสอบฟังก์ชันที่เขียนกับข้อมูลทดสอบหลายๆ ชุดโดยอัตโนมัติ (มีตัวอย่างใน Attachment ของ HW03\_1)

- 4 คะแนน (Lab03\_1\_6XXXXXXX.py) **[Attachments]** ให้เขียนโปรแกรมภาษา python เพื่อรับค่าพื้นที่ผิวของทรงกลมจาก User แล้วคำนวณปริมาตรของทรงกลมนั้น โดยต้องเขียน code ในฟังก์ชัน `find_r_from_surface_area(surface_area)` และ `sphere_volume(radius)` ทั้งนี้โปรแกรมที่ได้จะมีผลการ Run ดังแสดงด้านล่าง (ดูคำอธิบายทั้งหมดจาก Slide **เรื่อง Functions Part I** หน้า 15 - 20)

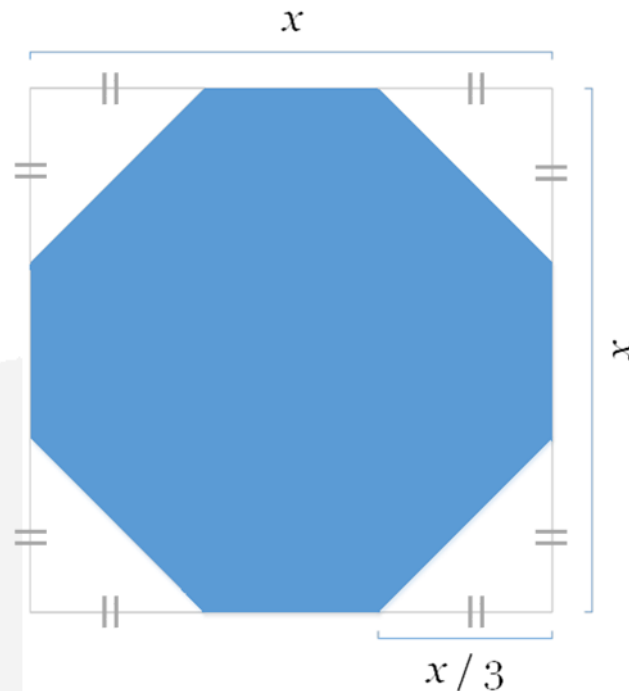
#### ตัวอย่างการ Run

```
input surface area: 50
volume = 33.25
```

#### การวิเคราะห์ปัญหา

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

2) **4 คะแนน** (Lab3\_2\_6XXXXXXX.py) ให้เขียนฟังก์ชัน `octagon_area(x)` เพื่อคืนค่าพื้นที่รูปแปดเหลี่ยมตามที่  
เรากำหนดความยาว  $x$  ดังรูป (สำหรับปัญหาในข้อนี้ให้นักศึกษาคิด test case เอง)



- การวิเคราะห์ปัญหา

• Input:		จำนวนข้อมูล	ชนิดข้อมูล
• Output:	(คี่ค่า)	จำนวนข้อมูล	ชนิดข้อมูล
• Output:	(แสดงค่า)	จำนวนข้อมูล	ชนิดข้อมูล

3) 4 คะแนน (HW03\_1\_6XXXXXXX.py) **[Attachments]** ให้เขียนฟังก์ชัน `nearest_odd(x)` เพื่อคืนค่าจำนวนคี่ที่ใกล้กับจำนวนจริง  $x$  ที่สุดโดยหาก  $x$  เป็นจำนวนคู่ ให้คืนค่าจำนวนคี่ที่น้อยกว่า  $x$  นอกจากนี้ไฟล์ที่ส่งจะต้องมีฟังก์ชัน `test_nearest_odd()` ที่ทำหน้าที่ทดสอบการทำงานของฟังก์ชัน `nearest_odd()` โดยใช้ **assert** ด้วย

**Hint:** จำนวนคี่คือจำนวนที่สามารถเขียนในรูป  $2n + 1$  หรือ  $2m - 1$  เมื่อ  $n, m$  เป็นจำนวนเต็ม

<u>Input</u>	<u>Output</u>
3	3
3.5	3
4	3
4.5	5

- การวิเคราะห์ปัญหา

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

- การวิเคราะห์ปัญหา

- 5) **4 คะแนน** (HW03\_3\_6XXXXXXX.py) ให้เขียนฟังก์ชัน `set_kth_digit(number, k, value)` เพื่อคืนค่าผลลัพธ์ของการเปลี่ยนค่าของหลักที่ระบุโดยตัวแปร  $k$  ของจำนวนเต็มบวก  $number$  ให้มีค่าเป็นตามที่กำหนดในตัวแปร  $value$  ( $0 \leq value \leq 9$ ) โดยกำหนดให้หลักที่อยู่ตำแหน่งขวาสุดคือหลักที่ 0 เช่น `set_kth_digit(2343, 2, 7)` จะเปลี่ยนหลักที่ 2 ของตัวเลข 2343 ให้เป็นเลข 7 และคืนค่า 2743 ทั้งนี้จะต้องมีการเรียกใช้ฟังก์ชัน `kth_digit()` จาก HW03\_2 และให้ถือว่า User จะไม่ใส่ตัวแปรในช่วงค่าที่ไม่ถูกต้อง (ไม่จำเป็นต้องตรวจสอบความถูกต้องของ Input)

<u>Input</u>	<u>Output</u>
2 <u>3</u> 43 2 7	27 <u>4</u> 3
5 <u>1</u> 0 2	5 <u>2</u>
1 2 5	<u>5</u> 01

- การวิเคราะห์ปัญหา

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