

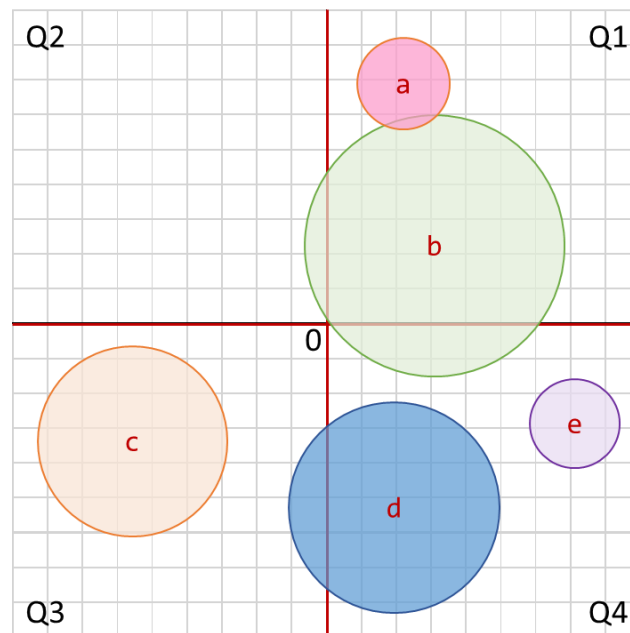


## การบ้านปฏิบัติการ 13

## Problem Solving and Algorithm Practice (20 คะแนน)

## ข้อกำหนด

- i. การเรียกใช้ฟังก์ชันเพื่อการทดสอบ ต้องอยู่ภายใต้เงื่อนไข `if __name__ == '__main__':` เพื่อความสะดวกในการ import จาก Script อื่นๆ
  - ii. นักศึกษาสามารถสร้างฟังก์ชันย่อยต่างๆ เพิ่มเติมได้ตามความเหมาะสม
- 1) 5 คะแนน (Lab13\_1\_6XXXXXXX.py) ให้เขียนฟังก์ชัน `count_segment(list_a)` เพื่อนับจำนวนส่วนของวงกลมที่อยู่ใน Quadrant ต่างๆ ที่ระบุด้วย `list_a` โดย `list_a` จะเป็น List ของ tuple ที่อยู่ในรูป  $(px, py, r)$  โดย  $px$  และ  $py$  คือพิกัดในแนวแกน  $x$  และแกน  $y$  ตามลำดับ และ  $r$  คือ รัศมีวงกลม ( $r \geq 0$ ) โดยฟังก์ชันจะคืนค่า tuple แทนจำนวนวงกลมหรือ Segment (ส่วนของวงกลม) ที่อยู่ใน Quadrant 1, 2, 3 และ 4 ตามลำดับ



เช่นจากรูปด้านบน ฟังก์ชันจะคืนค่า (2, 1, 2, 3)

InputOutput

<pre>[(2, 7, 1.5),  (3.2, 2.5, 4.06),  (-5.5, -4.5, 2.5),  (2, -5.2, 3),  (7.2, -2.8, 1.2)]</pre>	<pre># a # b # c # d # e</pre>	(2, 1, 2, 3)
---	--------------------------------	--------------

2) 5 คะแนน (Lab13\_2\_6XXXXXXXXX.py) ให้เขียนฟังก์ชัน `power_set(set_a)` เพื่อคืนค่า `list` ของ `set` ที่เป็น Power Set ทั้งหมดของ `set_a` โดยลำดับของผลลัพธ์ภายใน `list` ที่คืนค่าไม่จำเป็นต้องเหมือนตัวอย่างตามหลักวิชาคณิตศาสตร์ เซตกำลัง หรือ เพาเวอร์เซต (อังกฤษ: Power Set) เป็นเซตของเซตย่อยทั้งหมดของ `S` รวมทั้งเซตว่าง และเซต `S` เอง ถ้า `S` เป็นเซต `{a, b, c}` แล้วเซตย่อยของ `S` ได้แก่:

- `{}` (อาจเขียนแทนด้วย `∅` ได้แก่เซตว่าง)
- `{a}`
- `{b}`
- `{c}`
- `{a, b}`
- `{a, c}`
- `{b, c}`
- `{a, b, c}`

**Input****Output**

<code>{'a', 'b', 'c'}</code>	<code>[set(), {'a'}, {'b'}, {'c'}, {'a', 'b'}, {'a', 'c'}, {'b', 'c'}, {'a', 'b', 'c'}]</code>
------------------------------	--

3) 5 คะแนน (Lab13\_3\_6XXXXXXXXX.py) ให้เขียนฟังก์ชัน `word_count(text)` เพื่อนับจำนวนคำที่ปรากฏในสายอักขระ (String) `text` โดยฟังก์ชันจะ คืนค่าเป็น `dict` โดยมี `key` เป็นแต่ละคำที่ปรากฏใน `text` และมี `value` เป็นความถี่ ทั้งนี้ตัวอักษรที่อยู่ใน `key` จะต้องเป็นตัวอักษรพิมพ์เล็ก

**ข้อกำหนด**

- การนับความถี่จะเป็นแบบ Case Insensitive ('ant' และ 'Ant' ถือเป็นคำเดียวกัน)
- ข้อความในไฟล์จะเป็นภาษาอังกฤษมาตรฐานในรูปแบบที่ถูกต้อง (well-formed English)
- ไม่พิจารณาเครื่องหมายวรรคตอนต่างๆ เฉพาะที่ล้อมรอบคำ เช่น `!"#$%&'()*+,-./:;<=>?@[\\]^_`{|}~`
- เนื่องจาก `key` มีคุณสมบัติเป็น `set` ลำดับในการแสดงผลใน output จึงไม่จำเป็นต้องเหมือนตัวอย่าง

**Input****Output**

"He doesn't want to pay \$40,000 for a new car, but his wife doesn't seem to care."	<code>{'new': 1, 'but': 1, 'pay': 1, 'want': 1, 'seem': 1, 'care': 1, 'his': 1, '40,000': 1, 'wife': 1, 'a': 1, 'for': 1, 'car': 1, 'doesn't': 2, 'to': 2, 'He': 1}</code>
---	--

- 4) 5 คะแนน (Lab13\_4\_6XXXXXXX.py) ให้เขียนฟังก์ชัน `square_matrix(list_x)` เพื่อให้ List 2 มิติ `list_x` ที่มีสมาชิกเป็นจำนวนเต็มกลายเป็น matrix จัดรูปโดยเติม 0 เพื่อให้มีขนาด row และ column เท่ากัน โดยจะต้องคงทุก element ใน List เดิมไว้ และจำนวน 0 ที่เติมต้องเป็นจำนวนที่น้อยที่สุดที่เป็นไปได้ ทั้งนี้กำหนดให้ฟังก์ชันทำงานแบบ Destructive

Input	Output
[[2, 3, 4], [1, 2, 3]]	[[2, 3, 4], [1, 2, 3], [0, 0, 0]]
[[1, 2], [1, 2, 3], [1, 2], [1, 2], [1]]	[[1, 2, 0, 0, 0], [1, 2, 3, 0, 0], [1, 2, 0, 0, 0], [1, 2, 0, 0, 0], [1, 0, 0, 0, 0]]

### การส่งงาน

1. ลักษณะ/ลำดับข้อความของการรับค่า/แสดงผล จะต้องเป็นไปตามที่ระบุในตัวอย่างการ run
2. ไฟล์งานที่ส่ง จะต้องมีการแทรก comment ที่ต้นไฟล์ระบุข้อมูลดังนี้
 

```
#รหัสประจำตัวนศ
#ชื่อ นามสกุล
#section
#Labx_y ( x คือชื่อปฏิบัติการ และ y คือข้อที่ )
```
3. ไฟล์งานโปรแกรมที่ส่ง จะต้องมีการแทรก pseudocode เป็น comment ในแต่ละขั้นตอน
4. ให้เขียนโปรแกรมส่วนทดสอบ (นอกจากฟังก์ชันที่กำหนด) ไว้ในฟังก์ชัน `main()` และใช้คำสั่งด้านล่างก่อนตรวจสอบไฟล์บนเกรดเดอร์
 

```
if __name__ == '__main__':
    main()
```
5. ทดสอบไฟล์บนเกรดเดอร์การบ้าน ที่ <http://202.28.248.55:10111/> โดยใช้ username และ password ที่แจกให้ในคาบแลบ
6. Upload ไฟล์ source code ที่ผ่านการตรวจจากเกรดเดอร์การบ้านในข้อ 4 แล้วไปยัง website ที่ใช้ส่งการบ้าน <http://hw.cs.science.cmu.ac.th/> เลือกตาม section ที่นักศึกษาเรียน