

FACULTY OF ENGINEERING
CHULALONGKORN UNIVERSITY
2110101 Computer Programming

YEAR I, First Semester, Final Examination, December 1, 2022, 8:30 – 11:30

ชื่อ-สกุล _____ เลขประจำตัว

--	--	--	--	--	--	--	--

 ห้องสอบ _____ เลขที่สอบ

--	--

หมายเหตุ

1. ข้อสอบมีจำนวน 12 แผ่น 12 หน้า
2. ไม่อนุญาตให้นำตำราและเอกสารใดๆ เข้าในห้องสอบ
3. ไม่อนุญาตให้ใช้เครื่องคำนวณแบบใด ๆ
4. ห้ามการหยิบยืมสิ่งใดๆ ทั้งสิ้น จากผู้สอบอื่นๆ เว้นแต่เจ้าหน้าที่ควบคุมการสอบจะหยิบยืมให้
5. ห้ามนำส่วนใดส่วนหนึ่งของข้อสอบและสมุดคำตอบออกจากห้องสอบ
6. ผู้เข้าสอบสามารถออกจากห้องสอบได้ หลังจากผ่านการสอบไปแล้ว 45 นาที
7. เมื่อหมดเวลาสอบ ผู้เข้าสอบต้องหยุดการเขียนใดๆ ทั้งสิ้น
8. นิสิตกระทำผิดเกี่ยวกับการสอบ ตามข้อบังคับจุฬาลงกรณ์มหาวิทยาลัย มีโทษ คือ พ้นสภาพการเป็นนิสิต หรือ ได้รับ สัญลักษณ์ F ในรายวิชาที่กระทำผิด และอาจพิจารณาให้ถอนรายวิชาอื่นทั้งหมดที่ลงทะเบียนไว้ในภาคการศึกษานี้

เขียนเลขประจำตัว เลขห้อง และที่นั่งสอบ
ในหน้าที่ 1, 9, 10, 11 และ 12
ห้ามแกะหรือแยกชุดข้อสอบโดยเด็ดขาด

ห้ามนิสิตพกโทรศัพท์และอุปกรณ์สื่อสารไว้กับตัวระหว่างสอบ หากตรวจพบจะถือว่า
นิสิตกระทำผิดเกี่ยวกับการสอบ อาจต้องพ้นสภาพการเป็นนิสิต หรือ ให้ได้รับ F และ
อาจพิจารณาให้ถอนรายวิชาอื่นทั้งหมดที่ลงทะเบียนไว้ในภาคการศึกษานี้

* รวบรวมรณรงค์การไม่กระทำผิดและการไม่ทุจริตการสอบที่คณะวิศวกรรมศาสตร์ *

ข้าพเจ้ายอมรับในข้อกำหนดที่กล่าวมานี้ ข้าพเจ้าเป็นผู้ทำข้อสอบนี้ด้วยตนเองโดยมิได้รับ
การช่วยเหลือ หรือให้ความช่วยเหลือ ในการทำข้อสอบนี้

ลงชื่อนิสิต.....

วันที่.....

คำสั่งพื้นฐาน	import math
<p><code>x = int(input())</code> รับข้อมูลจำนวนเต็ม 1 จำนวน</p> <p><code>x,y,z = [float(e) for e in input().split()]</code> รับข้อมูลจำนวนจริง 3 จำนวนในบรรทัดเดียวกัน คั่นด้วยช่องว่าง</p> <p><code>int(x), float(x), str(x):</code> คืนค่า <code>x</code> ที่ถูกเปลี่ยนประเภทข้อมูลเป็นจำนวนเต็ม จำนวนจริง และสตริง</p> <p><code>abs(n):</code> คืนค่าสัมบูรณ์ของ <code>n</code></p> <p><code>round(f):</code> คืนค่าจำนวนเต็มที่เกิดจากการปัดเศษจำนวนจริง <code>f</code> โดยถ้าเศษของ <code>f</code> มีค่าตั้งแต่ 0.5 จะปัดขึ้น ถ้าน้อยกว่า 0.5 จะปัดลง</p> <p><code>round(f,d):</code> คืนค่าจำนวนจริงที่เกิดจากการปัดเศษจำนวนจริง <code>f</code> โดยปัดให้มีจำนวนตัวเลขหลังจุดทศนิยม <code>d</code> หลัก</p> <p><code>range(start,stop [,step])</code> หรือ <code>range(stop):</code> คืนค่าเป็นลิสต์ของตัวเลขตามลำดับตั้งแต่ <code>start</code> ถึง <code>stop-step</code> และเพิ่มขึ้นทีละ <code>step</code> (ถ้าไม่ระบุ <code>start</code> จะมีค่า 0 และ <code>step</code> จะมีค่า 1)</p> <p><code>enumerate(L):</code> คืนลิสต์ของ tuple (index, element) ของแต่ละข้อมูลในลิสต์ <code>L</code></p> <p><code>len(a):</code> คืนค่าเป็นจำนวนข้อมูลใน <code>a</code> ซึ่ง <code>a</code> อาจเป็นลิสต์ ดิกชันนารี เซต ทูเปิล สตริง หรือ numpy array ก็ได้</p> <p><code>max(a), min(a):</code> คืนค่าที่มาก/น้อยที่สุดของข้อมูลใน <code>a</code> ซึ่ง <code>a</code> อาจเป็นลิสต์ ดิกชันนารี เซต ทูเปิล หรือสตริงก็ได้ (numpy array ใช้ <code>np.max(a), np.min(a)</code>)</p> <p>ถ้า <code>a</code> เป็นดิกชันนารี จะคืนค่ามาก/น้อยที่สุดของ key ของดิกชันนารี</p> <p><code>type(a):</code> คืนค่าประเภทของ <code>a</code> เช่น <code>type([1,2])</code> ได้ <code><class 'list'></code></p> <p><code>list(), dict(), tuple(), set():</code> สร้างลิสต์ว่าง ดิกชันนารีว่าง ทูเปิลว่าง เซตว่าง</p>	<p><code>math.exp(x):</code> คืนค่า <code>e</code> ยกกำลัง <code>x</code></p> <p><code>math.cos(x):</code> คืนค่า cosine ของ <code>x</code> เรเดียน</p> <p><code>math.sin(x):</code> คืนค่า sine ของ <code>x</code> เรเดียน</p> <p><code>math.sqrt(x):</code> คืนค่ารากที่สองของ <code>x</code></p> <p><code>math.log(x,base):</code> คืนค่าลอการิทึมของ <code>x</code> ฐาน <code>base</code></p> <p><code>math.degrees(x):</code> แปลงมุม <code>x</code> จากเรเดียนเป็นองศา</p> <p><code>math.radians(x):</code> แปลงมุม <code>x</code> จากองศาเป็นเรเดียน</p> <p><code>math.pi, math.e:</code> ค่าคงที่ <code>pi</code> และ <code>e</code></p>
	string s
	<p><code>s.lower():</code> คืนสตริงใหม่ที่มีค่าเหมือน <code>s</code> แต่เป็นตัวพิมพ์เล็กทั้งหมด</p> <p><code>s.upper():</code> คืนสตริงใหม่ที่มีค่าเหมือน <code>s</code> แต่เป็นตัวพิมพ์ใหญ่ทั้งหมด</p> <p><code>s.find(sub):</code> คืน index แรกสุดที่พบ <code>sub</code> ใน <code>s</code> ถ้าไม่พบคืนค่า -1</p> <p><code>s.find(sub,i):</code> คืน index แรกสุดที่พบ <code>sub</code> ใน <code>s</code> โดยเริ่มค้นที่ index <code>i</code></p> <p><code>s.count(sub):</code> คืนจำนวนครั้งที่ <code>sub</code> ปรากฏในสตริง <code>s</code></p> <p><code>s.split(sep):</code> คืนลิสต์ของสตริงที่แยกด้วย <code>sep</code> (หรือ space ถ้าไม่ระบุ)</p> <p><code>s.strip():</code> คืนสตริงใหม่ที่มีค่าเหมือน <code>s</code> แต่ตัด spaces หัวท้ายออก</p> <p><code>s.join(L):</code> คืนสตริงที่สร้างจากการนำแต่ละ element ในลิสต์ <code>L</code> มาต่อกัน โดยมี <code>s</code> เป็นตัวคั่นระหว่างข้อมูลที่ต่อกัน (<code>L</code> ต้องเป็นลิสต์ของสตริง)</p>
	import numpy as np
	<p><code>np.array(L):</code> คืนค่า numpy array ที่สร้างจากลิสต์ <code>L</code></p> <p><code>np.arange(start,stop,step):</code> คืนอาร์เรย์ 1 มิติของจำนวนที่มีค่าตาม <code>start,stop,step</code></p> <p><code>np.ones(shape):</code> คืนอาร์เรย์ที่มีค่า 1 ทั้งหมด มีขนาดตาม tuple shape</p> <p><code>np.zeros(shape):</code> คืนอาร์เรย์ที่มีค่า 0 ทั้งหมด มีขนาดตาม tuple shape</p> <p><code>np.identity(size):</code> คืนอาร์เรย์ขนาด <code>size x size</code> ซึ่งมีข้อมูลในแนวเส้นทแยงมุมเป็น 1 และค่าในตำแหน่งอื่น ๆ เป็น 0</p> <p><code>np.empty_like(a):</code> คืนอาร์เรย์ใหม่ที่มีขนาดเหมือน <code>a</code> แต่ไม่มีการกำหนดค่าข้อมูลในอาร์เรย์ใหม่</p> <p><code>np.add(a,b), np.subtract(a,b), np.multiply(a,b), np.divide(a,b):</code> คืนค่าอาร์เรย์ใหม่ที่เป็นผลบวกลบคูณหารแบบช่องต่อช่องของ <code>a</code> และ <code>b</code></p> <p><code>np.dot(a,b):</code> คืนอาร์เรย์ที่เป็นผลคูณแบบเมทริกซ์ของ <code>a</code> และ <code>b</code></p> <p><code>np.sin(a), np.cos(a), np.sqrt(a), np.abs(a):</code> คืนอาร์เรย์ที่มีค่าของข้อมูลในแต่ละตำแหน่งเป็นผลจากการเรียกฟังก์ชัน sine, cosine, sqrt, abs กับข้อมูลในอาร์เรย์ <code>a</code> ที่ตำแหน่งเดียวกัน</p> <p><code>np.max(a,axis), np.min(a,axis):</code> คืนอาร์เรย์ของค่ามาก/น้อยที่สุด ใน <code>a</code> ตาม axis ที่กำหนด</p> <p><code>np.argmax(a,axis), np.argmin(a,axis):</code> คืนอาร์เรย์ของ index ที่มีค่ามาก/น้อยที่สุดใน <code>a</code> ตาม axis ที่กำหนด ตัวอย่างเช่น</p> <pre>a = np.array([[2, 4, 6], [8, 10, 12]]) np.max(a) คืนค่า 12, np.argmax(a) คืนค่า 5 np.max(a,axis=0) คืนค่า array([8,10,12]) np.argmax(a,axis=0) คืนค่า array([1,1,1]) np.argmax(a,axis=1) คืนค่า array([2,2])</pre> <p><code>np.sum(), np.std(), np.mean():</code> มีการใช้งานเหมือน <code>np.max()</code></p> <p><code>np.ndenumerate(a):</code> คืนลิสต์ของ tuple (position,element) ของแต่ละข้อมูลใน <code>a</code> โดย position เป็น tuple ที่เก็บตำแหน่งของข้อมูล</p>
list L	
<p><code>L.append(e):</code> เพิ่ม <code>e</code> ไปที่ท้ายลิสต์ <code>L</code></p> <p><code>L.insert(index,e):</code> เพิ่ม <code>e</code> ไปที่ตำแหน่ง <code>index</code> ในลิสต์ <code>L</code></p> <p><code>L.pop(index):</code> ลบข้อมูลที่ตำแหน่ง <code>index</code> และคืนค่าข้อมูลที่ถูกลบ</p> <p><code>L.count(e):</code> คืนจำนวนครั้งที่ <code>e</code> ปรากฏในลิสต์ <code>L</code></p>	
dict D	
<p><code>D.items():</code> คืนลิสต์ของ tuple (key, value) ของดิกชันนารี <code>D</code></p> <p><code>D.keys():</code> คืนลิสต์ของ key ทั้งหมดของดิกชันนารี <code>D</code></p> <p><code>D.values():</code> คืนลิสต์ของ value ทั้งหมดของดิกชันนารี <code>D</code></p> <p><code>D.pop(k):</code> ลบข้อมูลใน <code>D</code> ที่มี key เป็น <code>k</code> และคืนค่า value ของ key นั้น</p> <p><code>D.update(D1):</code> เพิ่มข้อมูลจากดิกชันนารี <code>D1</code> เข้าไปใน <code>D</code></p>	
set S	
<p><code>S.add(e):</code> เพิ่ม <code>e</code> ในเซต <code>S</code></p> <p><code>S.difference(T):</code> คืนเซตใหม่ที่เท่ากับ <code>S-T</code></p> <p><code>S.discard(e):</code> ลบ <code>e</code> ออกจากเซต <code>S</code> ถ้าไม่มี <code>e</code> ใน <code>S</code> ก็ไม่ทำอะไร</p> <p><code>S.intersection(T):</code> คืนเซตใหม่ที่เท่ากับ <code>S ∩ T</code></p> <p><code>S.union(T):</code> คืนเซตใหม่ที่เท่ากับ <code>S ∪ T</code></p> <p><code>S.issubset(T):</code> ทดสอบว่า <code>S ⊆ T</code> หรือไม่</p> <p><code>S.issuperset(T):</code> ทดสอบว่า <code>S ⊇ T</code> หรือไม่</p> <p><code>S.pop():</code> ลบข้อมูลหนึ่งตัวออกจากเซต <code>S</code> และคืนค่าข้อมูลที่ถูกลบ</p> <p><code>S.update(T):</code> ให้ <code>S = S ∪ T</code></p>	

สำหรับคำถามปรนัย เลือกคำตอบที่ต้องการเพียงตัวเลือกเดียว และตอบในกระดาษคำตอบเท่านั้น

1. ค่า 10001 ในฐานสอง มีค่าเท่าใดในฐานสิบ
 - A) 13
 - B) 15
 - ☒ C) 17
 - D) 19
 - E) ไม่มีข้อใดถูก
2. ฮาร์ดแวร์ของเครื่องคอมพิวเตอร์ในปัจจุบันทำงานกับข้อมูลและคำสั่งที่ถูกเข้ารหัสแบบใด
 - ☒ A) รหัสเลขฐานสอง
 - B) ตัวอักษร
 - C) ภาษาไพทอน
 - D) ภาษาซี
 - E) ไม่มีข้อใดถูก
3. 1 byte มีขนาดเท่าใด
 - A) 8 tokens
 - B) 8 lines
 - C) 8 words
 - ☒ D) 8 bits
 - E) ไม่มีข้อใดถูก
4. Compiler รับ computer program source code มาประมวลผลเพื่อให้ได้อะไรเป็นผลลัพธ์
 - ☒ A) รหัสภาษาเครื่อง
 - B) รหัสภาษาซี
 - C) รหัสแอส
 - D) รหัสลับ
 - E) ไม่มีข้อใดถูก
5. ในเครื่องคอมพิวเตอร์ RAM คืออะไร
 - ☒ A) หน่วยความจำ
 - B) หน่วยถอดรหัส
 - C) หน่วยรับข้อมูล
 - D) หน่วยสื่อสาร
 - E) ไม่มีข้อใดถูก
6. ข้อใดคือผลลัพธ์ของ `a = {2:3, 4:5, 6:7}; print(a[2]+a[4]+a[6])`
 - A) 12
 - B) 13
 - C) 14
 - ☒ D) 15
 - E) ไม่มีข้อใดถูก
7. โปรแกรมทางขวานี้ทำงานแล้วจะแสดงอะไรเสมอ *dict keys ลิสต์ random?*
 - A) 15
 - B) 357
 - C) 1200
 - ☒ D) 200400600
 - E) ไม่มีข้อใดถูก

```
a = {'200': '3', '400': '5', '600': '7'}
s = ''
for e in a: s += e
print(s)
```
8. โปรแกรมทางขวานี้ทำงานแล้วจะแสดงอะไร
 - A) 16
 - B) 29
 - C) 30
 - ☒ D) 32
 - E) ไม่มีข้อใดถูก

```
d = {3:4, 4:5, 6:7}
for e in d:
    d[6] += e
print(d[3]+d[4]+d[6])
```
9. โปรแกรมทางขวานี้ทำงานแล้วจะแสดงอะไร
 - A) 11
 - B) 12
 - C) 13
 - ☒ D) 14
 - E) ไม่มีข้อใดถูก *Key Error*

```
d = dict()
for i in range(5): d[i] = i
for e in d:
    d[-1] += e
print(d[-1])
```
10. เมื่อโปรแกรมทางขวานี้ทำงาน คำสั่ง `print` จะแสดงผลกี่บรรทัด
 - A) 8 บรรทัด
 - B) 12 บรรทัด
 - ☒ C) 14 บรรทัด
 - D) 16 บรรทัด
 - E) ไม่มีข้อใดถูก

```
for i in range(1, 5):
    for j in range(i, 6):
        print(i, j)
```
11. เมื่อโปรแกรมทางขวานี้ทำงาน แล้วออกจากวงวนนอกสุดแล้ว `i` และ `j` มีค่าเท่าใด
 - A) 8 กับ 13
 - B) 9 กับ 12
 - ☒ C) 14 กับ 14
 - D) 14 กับ 7
 - E) ไม่มีข้อใดถูก

```
for i in range(8, 15):
    for j in range(i, 20):
        if (i+j)%7 == 0: break
        pass
```
12. คำสั่ง `[[x, x+1] for x in range(0, 2)]` ได้ผลเหมือนคำสั่งในข้อใด
 - A) `[0,1,1,2]`
 - B) `[0,1,1,2,2,3]`
 - ☒ C) `[[0,1],[1,2]]`
 - D) `[[0,1],[1,2],[2,3]]`
 - E) ไม่มีข้อใดถูก
13. ข้อใดคือผลลัพธ์ของโปรแกรมด้านขวา
 - ☒ A) 6
 - B) 5
 - C) 4
 - D) 3
 - E) ไม่มีข้อใดถูก

```
d = {3:1, 4:2, 5:3, 6:4}
x = [[v,k] for k,v in d.items()]
x.sort()
print(x[-1][-1])
```

14. ข้อใดคือผลลัพธ์ของโปรแกรมด้านขวา

- A) 10 B) 18 C) 28 D) 32 E) ไม่มีข้อใดถูก

```
d = {3:1, 4:2, 5:3, 6:4}
print(sum([e+d[e] for e in d]))
```

$4+6+8+10 = 28$

15. ให้ $x = \{1:4, 5:8, 2:6, 8:2\}$ ข้อใดเป็นผลของ `sum([e for e in x if x[e]>5])`

- A) 6 B) 7 C) 8 D) 14 E) ไม่มีข้อใดถูก

16. ให้ $x = \{1:[2,3,1], 2:[0,1,3], 4:[3,2]\}$ ข้อใดเป็นผลของคำสั่ง `x[4][1]`

- A) 0 B) 1 C) 2 D) 3 E) ไม่มีข้อใดถูก

17. ให้ $x = \{'B':\ 'C', 'A':\ 'B', 'D':\ 'A', 'C':\ 'D'\}$

ข้อใดเป็นผลของ `','.join([x[e] for e in sorted(x.keys())])`

- A) 'A,B,C,D' B) 'C,B,A,D' C) 'B,A,D,C' D) 'B,C,D,A' E) ไม่มีข้อใดถูก

18. ให้ $s1 = \{1,2,3,4\}$ และ $s2 = \{2,3,4,5\}$ คำสั่ง `s1.union(s2)` ทำให้ $s1$ เป็นดังข้อใด *ไม่ replace*

- A) $\{1,2,3,4,5\}$ B) $\{2,3,4\}$ C) $\{1,5\}$ D) $\{1,2,3,4\}$ E) ไม่มีข้อใดถูก

19. ถ้าเราต้องการให้บริการตอบคำถามว่า เลขท้ายสองตัวของสลากกินแบ่งที่กำหนดให้ ออกรางวัลในงวดใดบ้าง การเก็บข้อมูลในข้อใดทำให้ทำงานได้ตามที่ต้องการ

- A) list จำนวน 100 ช่อง ใช้ index แทนเลขท้ายสองตัว แต่ละช่องเก็บ set ของวันเดือนปีที่ออกรางวัล
B) list จำนวน 100 ช่อง ใช้ index แทนเลขท้ายสองตัว แต่ละช่องเก็บ list ของวันเดือนปีที่ออกรางวัล
C) dict ที่มี key เป็นเลขท้ายสองตัว และ value เป็น set ของวันเดือนปีที่ออกรางวัล
D) dict ที่มี key เป็นเลขท้ายสองตัว และ value เป็น list ของวันเดือนปีที่ออกรางวัล

~~E) ทุกข้อให้บริการได้ตามโจทย์~~

20. พิจารณาฟังก์ชัน c ทางขวานี้ คำสั่ง `c("123") + c("12341")` จะได้ค่าเท่าไร

- A) 4 B) 5 C) 6 D) 7 E) ไม่มีข้อใดถูก

```
def c(t):
    s = set()
    for e in t:
        s.add(e)
    return len(s)
```

21. ให้ t เป็น dict ที่มี key เป็นชื่อย่อจังหวัด และ value แทนจำนวนประชากรในจังหวัด

คำสั่งในข้อใดหาจังหวัดที่มีประชากรมากที่สุด (กำหนดให้จังหวัดที่มีประชากรมากที่สุดมีเพียงจังหวัดเดียว)

- A) `max(t.values())` B) `max([(t[k],k) for k in t])[-1]`
C) `sorted([(k,t[k]) for k in t])[-1][0]` D) `max([(v,t[v]) for v in t.values()])[1]`
E) ไม่มีข้อใดถูก

คำถาม 11 ข้อต่อไปนี้จะถือว่าได้ทำคำสั่ง `import numpy as np` แล้ว

ให้ $x = np.array([[4,3,2,1], [2,5,0,-4], [4,3,1,3]])$ จงตอบ 3 คำถามข้างล่างนี้

22. ข้อใดเป็นผลของ `np.max(x)`

- A) 1 B) 2 C) 3 D) 4 E) 5

23. ข้อใดเป็นผลของ `np.argmax(x[1])`

- A) 1 B) 2 C) 3 D) 4 E) 5

24. ข้อใดเป็นผลของ `np.sum(np.min(x,axis=0))` $\text{min}([2,3,0,-4]) = -4$

- A) 1 B) 2 C) 3 D) 4 E) -2

25. ข้อใดข้างล่างนี้ได้ผลเหมือนคำสั่งสร้างอาร์เรย์ `d = np.array([[1,0,0],[0,1,0],[0,0,1]])`
- A) `d = np.zeros((3,3), dtype=int)` B) `d = np.identity(3, dtype=int)`
 C) `d = np.diag_one(3, dtype=int)` D) `d = np.arrayone((3,3), dtype=int)`
 E) ไม่มีข้อใดถูก
26. ให้ `d = np.array([[1,2,3,4],[5,6,7,8]])` ข้อใดเป็นผลของ `d.shape`
- A) `[4,2]` B) `[2,4]` C) `(4,2)` D) `(2,4)` E) ไม่มีข้อใดถูก
27. ให้ `d` เป็น **numpy array** ขนาด `100x100` คำสั่งในข้อใดทำให้ข้อมูลในแถวแนวนอนหมายเลขคู่ทั้งหมดมีค่าเป็น 0 ทั้งหมด
- A) `d[:,2] = 0` B) `d[:2] = 0`
 C) `d[:,2] = 0` D) `d[:,2] = 0` E) ไม่มีข้อใดถูก
28. ให้ `a` และ `b` เป็น **numpy array** ของเลขจำนวนจริงหนึ่งมิติที่มีขนาดเท่ากัน หากเราให้ `a` และ `b` แทนเวกเตอร์ 2 ตัว คำสั่งในข้อใดไม่ใช้การหา dot product ของ vectors ทั้งสอง
- A) `np.sum(a*b)` B) `sum([a[i]*b[i] for i in range(len(a))])`
 C) `a.dot(b)` D) `np.dot(b,a)` E) หาได้เหมือนกันหมด
29. ให้ `d = np.array([[9,8,7,6]])` คำสั่งในข้อใดไม่ได้ ผลเป็น **numpy array** ขนาด `4x4`
- A) `d * d.T` B) `d.T * d`
 C) `np.dot(d.T, d)` D) `np.dot(d, d.T)` E) ได้หมดทุกข้อ
30. ให้ `a = np.array([[3,3],[4,4]])`; `b = np.array([[1,1,1],[2,2,2]])` แทนเมทริกซ์ 2 ตัว คำสั่งในข้อใดคือการคูณเมทริกซ์ `a x b`
- A) `a.multiply(b)` B) `np.multiply(a,b)`
 C) `a.dot(b)` D) `a.dot(b.T)` E) ได้หมดทุกข้อ
31. ให้ `x = np.array([[1,3,10,5,2,10,8,-8]])` เก็บพิกัด `x` ของจุดต่าง ๆ บนแกน `x` คำสั่งในข้อใดหาระยะทางของคู่จุดที่ห่างกันที่สุด
- A) `np.max(x - x.T)` B) `np.max(np.all_pair_distance(x))`
 C) `np.argmax(x) - np.argmin(x)` D) `np.max(np.dot(x,x.T))`
 E) ไม่มีข้อใดถูก
32. จากคลาส `A` ทางขวานี้ ให้ `x = [A(5), A(8), A(3)]` ถ้าทำคำสั่ง `x.sort()` ตามด้วย `print(x[0])` จะแสดงค่าอะไร
- A) 3 B) 5
 C) 8 D) เขียน class A ผิด จึง sort ไม่ได้
 E) ไม่มีข้อใดถูก
33. จากคลาส `A` ในข้อที่แล้ว (โดยตัดเมทอด `__lt__` และ `__str__` ออก) ให้ `x = A(99)` คำสั่งในข้อใดได้ผลต่างจากข้ออื่น
- A) `x.get_a() *= 3` B) `x.triple()`
 C) `x.a *= 3` D) `A.triple(x)` E) ได้ผลเหมือนกันหมดทุกข้อ
34. หากสั่งโปรแกรมข้างล่างนี้ (ทางซ้าย) ให้ทำงาน จะได้ผลดังแสดงทางขวา

```
class A :
    def __init__(self,a):
        self.a = a
    def __lt__(self,x):
        return x.a < self.a
    def __str__(self):
        return str(self.a)
    def get_a(self):
        return self.a
    def triple(self):
        self.a *= 3
```

```
class A :
    def __init__(self,x):
        self.x = x
    for e in [1,2.2,[1],"a",A(1)]:
        print(type(e))
```

```
<class 'int'>
<class 'float'>
<class 'list'>
<class 'str'>
<class '__main__.A'>
```

สรุปได้ว่า ข้อใดข้างล่างนี้ไม่ใช่ชื่อคลาสใน Python

- A) `int` B) `float` C) `str` D) `list` ~~E) เป็นชื่อคลาสทุกข้อ~~

35. ข้อใดข้างล่างนี้ไม่ใช่ชื่อเมทอด/ฟังก์ชัน ในคลาสที่เรียนในวิชานี้

- A) `sort` B) `append` C) `dot` ~~D) `shape`~~ E) เป็นชื่อเมทอด/ฟังก์ชันทุกข้อ

36. โปรแกรมหนึ่งทำงานได้อย่างถูกต้อง หากเราพบคำสั่ง `a = t.x(y)` ในโปรแกรมนี้ คำว่า `x` ของคำสั่งคืออะไร

- A) ชื่อคลาส ~~B) ชื่อเมทอด~~ C) ชื่อข้อมูลภายใน D) ชื่อตัวแปร E) ไม่มีข้อใดถูก

ศึกษาคลาส `Order` ฟังก์ชัน `get_total` และการทำงานของโปรแกรมข้างล่างนี้ จงตอบคำถาม 5 ข้อต่อไปนี้

```
class Order:
    def __init__(self):
        self.orderlines = []
    def add(self, name, price):
        self.orderlines.append( (name, price) )    # list of tuples
    def total(self):
        return sum([p for (n,p) in self.orderlines])
    def __lt__(self, rhs):
        return self.total() < rhs.total()
def get_total(orders):
    total = 0
    for od in orders:
        total += od.total()
    return total
o1 = Order(); o1.add("Congee",30); o1.add("Fried Rice",45); o1.add("Water",7)
o2 = Order(); o2.add("Papaya Salad",40); o2.add("Congee",30)
orders = [o1,o2,o1,o1]
```

37. หลังจากโปรแกรมข้างบนนี้ทำงาน ถ้าให้ทำคำสั่ง `o2.orderlines[1][0]` จะคืนผลอะไร

- A) "Papaya Salad" B) "Fried Rice" C) "Water" ~~D) "Congee"~~ E) ไม่มีข้อใดถูก

38. หลังจากโปรแกรมข้างบนนี้ทำงาน ถ้าให้ทำคำสั่ง `o1.total()` จะคืนผลอะไร

- A) 70 B) 75 ~~C) 82~~ D) 152 E) ไม่มีข้อใดถูก

39. หลังจากโปรแกรมข้างบนนี้ทำงาน ถ้าให้ทำคำสั่ง `o1 < o2` จะคืนผลอะไร

- A) `true` B) `false` C) `True` ~~D) `False`~~ E) ไม่มีข้อใดถูก

40. หลังจากโปรแกรมข้างบนนี้ทำงาน ถ้าให้ทำคำสั่ง `get_total(orders[2:3])` จะคืนผลอะไร

- ~~A) 70~~ B) 82 C) 152 D) 232 E) ไม่มีข้อใดถูก

ศึกษาคลาส `Menu` คลาส `Order` ฟังก์ชัน `get_total` และการทำงานของโปรแกรมข้างล่างนี้ จงตอบคำถาม 5 ข้อต่อไปนี้

```
class Menu:
    def __init__(self, name, price):
        self.name = name
        self.price = price

class Order:
    def __init__(self, date):
        self.date = date
        self.orderlines = []

    def add(self, menu, n):
        for i in range(n):
            self.orderlines.append(menu)
```



```
def total(self):
    return sum([menu.price for menu in self.orderlines])

def get_total(orders, date):
    return sum([od.total() for od in orders if od.date == date])

m = [ Menu("fried rice", 45), Menu("phat thai", 50),
      Menu("Congee", 30), Menu("papaya salad", 40) ]
o1 = Order("1/03/2016"); o1.add(m[0],2); o1.add(m[3],1)
o2 = Order("1/03/2016"); o2.add(m[1],2); o2.add(m[0],1)
o3 = Order("2/03/2016"); o3.add(m[1],1); o3.add(m[2],1)
o4 = Order("2/03/2016"); o4.add(m[2],5)
orders = []
orders.append(o1); orders.append(o2);
orders.append(o3); orders.append(o4)
```

41. หลังจากโปรแกรมข้างบนนี้ทำงาน ถ้าให้ทำคำสั่ง `o3.orderlines[1].price` จะคืนผลอะไร
- ☒ A) 30 B) 40 C) 45 D) 50 E) ไม่มีข้อใดถูก
42. หลังจากโปรแกรมข้างบนนี้ทำงาน ถ้าให้ทำคำสั่ง `o2.orderlines[1].price` จะคืนผลอะไร
- A) 30 B) 40 ☒ C) 45 D) 50 E) ไม่มีข้อใดถูก
43. หลังจากโปรแกรมข้างบนนี้ทำงาน ถ้าให้ทำคำสั่ง `o1.total()` จะคืนผลอะไร
- A) 85 B) 95 ☒ C) 130 D) 140 E) ไม่มีข้อใดถูก
44. หลังจากโปรแกรมข้างบนนี้ทำงาน ถ้าให้ทำคำสั่ง `m[0].price=40` แล้วคำสั่ง `o1.total()` จะคืนผลอะไร
- A) 80 B) 90 ☒ C) 120 D) 130 E) ไม่มีข้อใดถูก
45. หลังจากโปรแกรมข้างบนนี้ทำงาน ถ้าให้ทำคำสั่ง `get_total(orders, "2/03/2016")` จะคืนผลอะไร
- A) 85 B) 165 ☒ C) 230 D) 275 E) ไม่มีข้อใดถูก

กระดาษคำตอบวิชา 2110101 Computer Programming ภาคปลาย ปีการศึกษา 2565

วันที่ 1 ธันวาคม 2565 เวลา 8.30 - 11.30 น.

- | | A | B | C | D | E | | A | B | C | D | E |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1) | (A) | (B) | (C) | (D) | (E) | 31) | (A) | (B) | (C) | (D) | (E) |
| 2) | (A) | (B) | (C) | (D) | (E) | 32) | (A) | (B) | (C) | (D) | (E) |
| 3) | (A) | (B) | (C) | (D) | (E) | 33) | (A) | (B) | (C) | (D) | (E) |
| 4) | (A) | (B) | (C) | (D) | (E) | 34) | (A) | (B) | (C) | (D) | (E) |
| 5) | (A) | (B) | (C) | (D) | (E) | 35) | (A) | (B) | (C) | (D) | (E) |
| 6) | (A) | (B) | (C) | (D) | (E) | 36) | (A) | (B) | (C) | (D) | (E) |
| 7) | (A) | (B) | (C) | (D) | (E) | 37) | (A) | (B) | (C) | (D) | (E) |
| 8) | (A) | (B) | (C) | (D) | (E) | 38) | (A) | (B) | (C) | (D) | (E) |
| 9) | (A) | (B) | (C) | (D) | (E) | 39) | (A) | (B) | (C) | (D) | (E) |
| 10) | (A) | (B) | (C) | (D) | (E) | 40) | (A) | (B) | (C) | (D) | (E) |
| 11) | (A) | (B) | (C) | (D) | (E) | 41) | (A) | (B) | (C) | (D) | (E) |
| 12) | (A) | (B) | (C) | (D) | (E) | 42) | (A) | (B) | (C) | (D) | (E) |
| 13) | (A) | (B) | (C) | (D) | (E) | 43) | (A) | (B) | (C) | (D) | (E) |
| 14) | (A) | (B) | (C) | (D) | (E) | 44) | (A) | (B) | (C) | (D) | (E) |
| 15) | (A) | (B) | (C) | (D) | (E) | 45) | (A) | (B) | (C) | (D) | (E) |
| 16) | (A) | (B) | (C) | (D) | (E) | 46) | (A) | (B) | (C) | (D) | (E) |
| 17) | (A) | (B) | (C) | (D) | (E) | 47) | (A) | (B) | (C) | (D) | (E) |
| 18) | (A) | (B) | (C) | (D) | (E) | 48) | (A) | (B) | (C) | (D) | (E) |
| 19) | (A) | (B) | (C) | (D) | (E) | 49) | (A) | (B) | (C) | (D) | (E) |
| 20) | (A) | (B) | (C) | (D) | (E) | 50) | (A) | (B) | (C) | (D) | (E) |
| 21) | (A) | (B) | (C) | (D) | (E) | | | | | | |
| 22) | (A) | (B) | (C) | (D) | (E) | | | | | | |
| 23) | (A) | (B) | (C) | (D) | (E) | | | | | | |
| 24) | (A) | (B) | (C) | (D) | (E) | | | | | | |
| 25) | (A) | (B) | (C) | (D) | (E) | | | | | | |
| 26) | (A) | (B) | (C) | (D) | (E) | | | | | | |
| 27) | (A) | (B) | (C) | (D) | (E) | | | | | | |
| 28) | (A) | (B) | (C) | (D) | (E) | | | | | | |
| 29) | (A) | (B) | (C) | (D) | (E) | | | | | | |
| 30) | (A) | (B) | (C) | (D) | (E) | | | | | | |

ชื่อ-สกุล _____

เลขประจำตัวนิสิต

0	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)
1	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)
2	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)
3	(3)	(3)	(3)	(3)	(3)	(3)	(3)	(3)	(3)	(3)
4	(4)	(4)	(4)	(4)	(4)	(4)	(4)	(4)	(4)	(4)
5	(5)	(5)	(5)	(5)	(5)	(5)	(5)	(5)	(5)	(5)
6	(6)	(6)	(6)	(6)	(6)	(6)	(6)	(6)	(6)	(6)
7	(7)	(7)	(7)	(7)	(7)	(7)	(7)	(7)	(7)	(7)
8	(8)	(8)	(8)	(8)	(8)	(8)	(8)	(8)	(8)	(8)
9	(9)	(9)	(9)	(9)	(9)	(9)	(9)	(9)	(9)	(9)

ห้องสอบ

(0)	(0)	(0)
(1)	(1)	(1)
(2)	(2)	(2)
(3)	(3)	(3)
(4)	(4)	(4)
(5)	(5)	(5)
(6)	(6)	(6)
(7)	(7)	(7)
(8)	(8)	(8)
(9)	(9)	(9)

เลขที่นั่งสอบ

(0)	(0)
(1)	(1)
(2)	(2)
(3)	(3)
(4)	(4)
(5)	(5)
(6)	(6)
(7)	(7)
(8)	(8)
(9)	(9)

(ในกรณีที่เลขที่นั่งสอบเป็นเลขหลักเดียว ให้เติม เลข 0 ไว้ด้านหน้าให้ครบสองหลัก เช่น เลขที่นั่งสอบ 1 ให้เติมเป็น 01)

ข้าพเจ้ารับทราบกฎระเบียบการสอบของมหาวิทยาลัย และยินยอมรับบทลงโทษหากกระทำความผิดหรือร่วมกระทำความผิดในการสอบทุกประการ

ลงนาม

ใช้ดินสอ 2B ระบายในช่องวงกลมที่เป็นคำตอบที่เลือกให้เติมวง
ขอละหนึ่งคำตอบเท่านั้น (ข้อที่เลือกตอบมากกว่าหนึ่งตัวเลือก
จะได้ศูนย์คะแนน) อย่าพับหรือทำกระดาษคำตอบยับ และ
ห้ามขีดเขียนบริเวณอื่นใด ยกเว้นเฉพาะบริเวณที่กำหนดให้เท่านั้น

เขียนตอบข้อนี้ในหน้านี้ ด้วยลายมือที่ชัดเจนอ่านง่าย (หากเนื้อที่ไม่พอให้เขียนด้านหลังของกระดาษแผ่นนี้เท่านั้น)

เขียน # comment ประกอบด้วย เพื่อให้ผู้ตรวจสอบสามารถทำความเข้าใจโปรแกรมได้โดยง่าย

- จงเขียนฟังก์ชัน **zip(x, y)** ที่รับ **x** และ **y** เป็นลิสต์ของจำนวนเต็ม

ฟังก์ชันนี้คืนลิสต์ของจำนวนเต็ม ที่มีขนาดเท่ากับขนาดของ **x** และ **y** รวมกันโดยข้อมูลในลิสต์มาจากการสลับนำข้อมูลจากลิสต์ **x** และ **y** มาใส่ (โดยห้ามใช้ฟังก์ชัน **zip** ที่มีอยู่แล้วของไพทอน) เช่น

zip([-1,-2,-3], [10,20,30,40]) ได้ผลเป็น **[-1,10,-2,20,-3,30,40]**

zip([-1,-2,-3,-4,-5], [10,20]) ได้ผลเป็น **[-1,10,-2,20,-3,-4,-5]**

(10 คะแนน)

```
def zip(x,y):
    xIdx=0
    yIdx=0
    res=[]
    while xIdx<len(x) or yIdx<len(y): # loop until both x and y reach the ends
        if xIdx<len(x):
            res.append(x[xIdx])
            xIdx+=1
        if yIdx<len(y):
            res.append(y[yIdx])
            yIdx+=1
    return res
```

- จงเขียนฟังก์ชัน **reverse_digits(t)** ที่รับ **t** เป็นสตริง ฟังก์ชันนี้คืนสตริงใหม่ ที่มีอักขระทั้งหลายในลำดับที่เหมือนกับ **t** แต่ตัวที่เป็นตัวเลขจะกลับลำดับ

เช่น **reverse_digits('You are 16 going on 17. A3B53')** ได้ผลเป็น

'You are 35 going on 37. A1B61'

(10 คะแนน)

```
def reverse_digits(t):
    nums = [x for x in t if x.isnumeric()] # create list containing only numbers in reverse order
    idx = len(nums)-1
    ans = ""
    for c in t:
        if c.isnumeric():
            ans += nums[idx]
            idx -= 1
        else:
            ans += c
    return ans
```

เขียนตอบข้อนี้ในหน้านี้ ด้วยลายมือที่ชัดเจนอ่านง่าย (หากเนื้อที่ไม่พอให้เขียนด้านหลังของกระดาษแผ่นนี้เท่านั้น)

Numpy: โปรแกรมต่าง ๆ ในหน้านี้ ถือว่าได้เขียน `import numpy as np` ไว้แล้ว

จงเขียนฟังก์ชันที่ทำงานตามที่ระบุใน **comment** ใช้คำสั่งของ **numpy** ให้มากที่สุด ห้ามใช้คำสั่ง **for, while, recursive, list comprehension, map, reduce** (**ถ้าใช้จะไม่ได้คะแนน**)

```
import numpy as np
```

```
def nearest(D, x):
```

```
# รับ: D เป็นอาร์เรย์หนึ่งมิติของจำนวนจริง และ x เก็บจำนวนจริง
```

```
# คืน: ค่าของข้อมูลใน D ที่มีค่าใกล้กับ x ที่สุด
```

```
# เช่น: nearest(np.array([3.1, 2.2, 4.5, 9.0]), 2.1) ได้ 2.2
```

```
return D[np.argmin(np.abs(D-x))] # 4 คะแนน
```

```
#-----
def z_score(X):
```

```
# รับ: X เป็นอาร์เรย์หนึ่งมิติของจำนวนจริง
```

```
# คืน: อาร์เรย์หนึ่งมิติเก็บค่า z score ของข้อมูลใน X
```

```
# โดยที่  $Z_i = \frac{x_i - \bar{x}}{s}$ ,  $\bar{x}$  และ  $s$  คือค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐานของ X
```

```
return (X - np.mean(X)) / (np.std(X)) # 4 คะแนน
```

```
#-----
```

```
def transform(P, a, b, c, d):
```

```
# รับ: P เป็นอาร์เรย์สองมิติขนาดหลายแถวแต่ละแถวมี 2 ช่อง และจำนวนจริงอีก 4 จำนวน
```

```
# คืน: Q เป็นอาร์เรย์หนึ่งมิติขนาดเดียวกับ P
```

```
# ในแถวที่ k ของ Q:  $Q[k,0] = a * P[k,0] + b * P[k,1]$ 
```

```
# ในแถวที่ k ของ Q:  $Q[k,1] = c * P[k,0] + d * P[k,1]$ 
```

```
# เช่น P = np.array([[1, 2], [10, 20], [100, 200]]), a = 1, b = 2, c = 3, d = 4
```

```
# ได้ผล np.array([5, 11], [50, 110], [500, 1100])
```

```
M = np.array([a, b], [c, d])
```

```
return np.matmul(P, np.array([a, b], [c, d])) # 4 คะแนน
```

```
#-----
def is_magic(D):
```

```
# รับ: D เป็นอาร์เรย์สองมิติของจำนวนเต็ม ที่แทนเมทริกซ์จัตุรัส
```

```
# คืน: True หรือ False เพื่อบอกว่า D เป็นเมทริกซ์ magic square หรือไม่
```

```
# magic square คือเมทริกซ์จัตุรัสที่มีผลรวมของค่าใน
```

```
# ทุกแถว ทุกหลัก และตามแนวทแยงมุมทั้งสองแนว เท่ากันหมด ----->
```

```
n = D.shape[0]
```

or sum_ding_0

```
# 8 คะแนน
```

```
sum_row = np.sum(D, axis=1)
```

```
sum_col = np.sum(D, axis=0)
```

```
def get_sum_ding(A):
```

```
return np.sum(np.ndindex(A))
```

get an lambda A: np.sum(np.ndindex(A))

```
sum_ding_1 = get_sum_ding(D)
```

```
sum_ding_2 = get_sum_ding(D[:, ::-1]) # reverse columns to find another diagonal with the same row
```

```
def checkSame(A):
```

```
return np.sum(A[0, :] == A) == A.shape[0] # return true if all values are the same
```

```
return (checkSame(sum_row) and checkSame(sum_col) and sum_ding_1 == sum_row[0] and sum_ding_2 == sum_row[0])
```

8	11	14	1
13	2	7	12
3	16	9	6
10	5	4	15

magic square

เขียนตอบข้อนี้ในหน้านี้ ด้วยลายมือที่ชัดเจนอ่านง่าย (หากเนื้อที่ไม่พอให้เขียนด้านหลังของกระดาษแผ่นนี้เท่านั้น)

Class ให้เขียนตอบในช่องที่เว้นว่างของหน้านี้ : โจทย์นี้มี 2 คลาส **Grader** กับ **User** ล้อเลียนระบบเกรดเดอร์ที่เราใช้ในวิชานี้ จงศึกษาคลาสและการทำงานของคลาสทั้งสอง (ในหน้านี้และหน้าถัดไป) แล้วเติมชุดคำสั่งในเมทอดที่ยังเขียนไม่เสร็จในช่องที่เว้นว่างไว้

```
import random

class Grader:
    # ไว้สร้างออนเจกต์ของระบบ Grader แบบง่าย ๆ
    def __init__(self):
        # สร้างที่เก็บ user ต่าง ๆ ของระบบ grader
        self.users = []
    def new_user(self, user_id, user_name):
        # สร้าง user ใหม่ที่มีรหัส user_id และชื่อ user_name, บันทึกเก็บใน grader, แล้วก็คืน user ใหม่นี้เป็นผลลัพธ์
        user = User(user_id, user_name)
        self.users.append(user)
        return user
    def submit(self, user, problem_id, code):
        # รับการ submit โปรแกรม code ของโจทย์ปัญหา problem_id ส่งไปเก็บใส่ลิสต์ของ submissions ของ user
        if user not in self.users: return
        score = self.grade(problem_id, code) # เรียกใช้เมทอด grade เพื่อตรวจให้คะแนน code
        user.save_submission(problem_id, code, score)
    def grade(self, problem_id, code):
        # เกรดเดอร์ตัวนี้ถูกตั้งในสถานะทดสอบ ของจริงไม่ทำแบบคำสั่งข้างล่างนี้นะ !!!
        return random.randint(0,100) # สุ่มจำนวนเต็มระหว่าง 0 ถึง 100
    def top_users(self):
        # คืนลิสต์ของ user ทั้งหมด ที่มีคะแนนรวมมากที่สุด > user ลิสต์คะแนน score
        if len(self.users) == 0: return []
        u = sorted(self.users)[::-1] # u เก็บ user ที่มีคะแนนรวมเรียงจากมากไปน้อย
        for k in range(1, len(u)): # หาตำแหน่ง k น้อยสุดที่ตัวที่ k-1 กับตัวที่ k มีคะแนนรวมไม่เท่ากัน
            if u[k] < u[k-1]: return u[:k] # นั่นคือตัวที่ 0 ถึง k-1 มีคะแนนรวมเท่ากันหมด และเท่ากับคะแนนสูงสุด
        return u
#-----
class User:
    def __init__(self, user_id, user_name): # เก็บรหัส ชื่อ และ submissions ของโจทย์ปัญหาทั้งหลายของ user นี้
        self.id = user_id
        self.name = user_name
        self.submissions = dict() # {problem_id: [(code0,score0), (code1,score1), ...], ...}
    def save_submission(self, problem_id, code, score):
        # จัดเก็บการส่งโปรแกรม code ของโจทย์ปัญหา problem_id ที่ได้คะแนน score เก็บใส่ submissions
        self.submissions[problem_id].append((code, score))
    def total_score(self):
        # คืนผลรวมของคะแนนของทุก ๆ โจทย์ปัญหาที่ user คนนี้เคย submit
        # (คะแนนของโจทย์ปัญหาหนึ่ง คือคะแนนมากที่สุดของ submissions ต่าง ๆ ของโจทย์นั้น, ดูเมทอด score ในหน้าถัดไป)
        tot=0
        for problem_id in self.submissions:
            tot += self.score(problem_id)
        return tot
    def __lt__(self, rhs): # ใช้คะแนนรวมของ user เป็นตัวเปรียบเทียบ
        return self.total_score() < rhs.total_score()
```

สองเมทอดข้างล่างนี้ เป็นของคลาส User (เขียนต่อจากหน้าที่แล้ว)

```
def score(self, problem_id):
    # คืนคะแนนของ submission ของโจทย์ปัญหา problem_id (ถ้าเคยส่งหลายครั้ง ก็ใช้ครั้งที่ได้มากที่สุด)

    max_score = 0
    if problem_id in self.submissions:
        for code,sc in self.submissions[problem_id]:
            if sc > max_score: max_score = sc
    return max_score

def __str__(self):
    out = self.id + ': ' + self.name + ' [ '
    for pid in self.submissions:
        out += pid + ': ' + str(self.score(pid))+' '
    return out + ']'
```

#-----

ศึกษาตัวอย่างการใช้งาน

```
g = Grader()                                # สร้างระบบ grader

u1 = g.new_user('u001', 'ปรีก')              # สร้าง user ใหม่ 3 ราย
u2 = g.new_user('u002', 'แย้ม')
u3 = g.new_user('u003', 'ผิน')

g.submit(u1, 'p001', 'print("ok1")')         # u1 ขอ submit โปรแกรมของโจทย์ปัญหา p001
g.submit(u1, 'p001', 'print("ok2")')         # u1 ขอ submit โปรแกรมของโจทย์ปัญหา p001 อีกครั้ง
g.submit(u2, 'p001', 'print("ok3")')         # u2 ขอ submit โปรแกรมของโจทย์ปัญหา p001
g.submit(u2, 'p002', 'print("ok4")')         # u2 ขอ submit โปรแกรมของโจทย์ปัญหา p002
g.submit(u3, 'p002', 'print("ok5")')         # u3 ขอ submit โปรแกรมของโจทย์ปัญหา p002

tops = g.top_users()                         # ขอ user ทั้งหมดที่ได้คะแนนรวมมากที่สุด (อาจมีหลายรายที่ได้คะแนนมากที่สุด)
for u in tops:                               # หยิบ user ที่ได้คะแนนรวมมากที่สุด แต่ละรายมาแสดง
    print(u.id, u.name, ',', u.total_score(), ': ', u.submissions)
```