

# นายณัฐวิษณุ ขิแปงคณ 64090500404

สอบกลางภาควิชา CSS 112

การเขียนโปรแกรมคอมพิวเตอร์

เต็ม หนึ่งร้อยคะแนน คะแนนแต่ละข้อจะอยู่ในวงเล็บ

ข้อสอบมี 7 หน้า มี5ข้อใหญ่ ข้อA,B,C,D,Eข้อละ20คะแนน

## A. โค้ดข้างล่างจะพิมพ์คำตอบใด

1.

```
score = 0
def update_score(new_score):
    score = new_score
update_score(100)
print(score)
```

ก. 0 ข. 100

2.

```
def outer():
...     x = 0
...     def inner():
...         nonlocal x
...         x=100
...         print(x)
...
>>> outer()
```

ก. 0 ข. 100

3.

```
def outer():
...     x = 0
...     def inner():
...         nonlocal x
...         x=100
...         inner()
...         print(x)
...
>>> outer()
```

ก. 0 ข. 100

4.

```
>>> def func(y):
...     y += (3,)
...
>>> x=(1,2)
>>> func(x)
>>> print(x)
```

ก. (1,2,3)      ข. (1,2)

5. 

```
>>> x=[1,2]
>>> def func(y):
...     y += ['something else']
>>> func(x)
>>> print(x)
```

ก. [1,2]      ข. [1,2,'something else']

6. 

```
>>> x=[1,2]
>>> def func(y):
...     y = ['something else']
>>> func(x)
>>> print(x)
```

ก. [1,2]      ข. ['something else']

7. การประกาศฟังก์ชันนี้จะเกิดข้อผิดพลาดหรือไม่

```
>>> def func2(a=32,b=1,c):
...     c= 5
```

ก. เกิดข้อผิดพลาด      ข. ไม่เกิดข้อผิดพลาด

8. การประกาศฟังก์ชันนี้จะเกิดข้อผิดพลาดหรือไม่

```
>>> def func2(c,b=1,a=32):
...     c= 5
```

ก. เกิดข้อผิดพลาด      ข. ไม่เกิดข้อผิดพลาด

9 ข้อใดคือผลการprintด้านล่าง

```
>>>inbox = ['chairs', 'cats', 'water']
>>>global animals
>>>animals = ['dogs','cats','birds','insects','rats']
>>>def check_animals(box):
...     for item in inbox2:
...         if item in animals:
...             item+=' are animals'
>>>check_animbals(inbox)
>>>print(inbox)
```

ก. ['chairs', 'cats', 'water']      ข. ['chairs', 'cats are animals', 'water']

10. ฟังก์ชันนี้

```
x = [1, 2, 'something else']
```

```
>>> def add_all_list(*k):
...     li = list()
...     for k_ in k:
...         li.append(str(k_) + '3')
...     return li
```

เราเรียกฟังก์ชันด้านบนแบบไหน

ก. `print(add_all_list(*x))` ข. `print(add_all_list(x))`

ถึงจะได้ผลดังนี้

```
['13', '23', 'ss3']
```

## B. จงเขียนโปรแกรม

1. กำหนดคลิสต์ของตัวเลขมาให้ เช่น

```
li = [0.2, -1000, 1000, 33.21, -101.12, 0.01, 212, 0.4, -0.3, -100]
```

เราต้องการให้ทุกค่า  $i$  ที่อยู่ระหว่าง  $-0.5$  ถึง  $0.5$  ของ  $li$  มีค่าเป็น  $10000+i$  (คำว่าอยู่ระหว่างแปลว่าค่า  $i = -0.5, 0.5$  ไม่ได้อยู่ระหว่าง) เช่นคลิสต์ด้านบนจะได้ผล

```
[10000.2, -1000, 1000, 33.21, -101.12, 10000.01, 212, 10000.4, 9999.7, -100]
```

จงเขียนโปรแกรมโดยใช้บรรทัดเดียว และต้องใช้ `lambda function` เท่านั้น

```
li = [0.2, -1000, 1000, 33.21, -101.12, 0.01, 212, 0.4, -0.3, -100]
print(list(map(lambda i: i + 10000 if -0.5 < i < 0.5 else i, li)))
```

2. ทำข้อหนึ่ง โดยใช้ `list comprehension` บรรทัดเดียว

```
li = [0.2, -1000, 1000, 33.21, -101.12, 0.01, 212, 0.4, -0.3, -100]
print([i + 10000 if -0.5 < i < 0.5 else i for i in li])
```

3. กำหนดคลิสต์ของตัวเลขมาให้ เช่น

```
li = [0.2, -1000, 1000, 33.21, -101.12, 0.01, 212, 0.4, -0.3, -100]
```

เราต้องการลบทุกค่า  $i$  ที่อยู่ระหว่าง  $-0.5$  ถึง  $0.5$  ของ  $li$  โดยยังคงลำดับของค่าอื่นๆในคลิสต์ (คำว่าอยู่ระหว่างแปลว่าค่า  $i = -0.5, 0.5$  ไม่ได้อยู่ระหว่าง) เช่นคลิสต์ด้านบนจะได้ผล `[1000, 33.21, 212]`

จงเขียนโปรแกรมโดยใช้บรรทัดเดียว และต้องใช้ `lambda function` เท่านั้น

```
li = [0.2, -1000, 1000, 33.21, -101.12, 0.01, 212, 0.4, -0.3, -100]
print(list(filter(lambda x: x >= 0.5 or x <= -0.5, li)))
```

#### 4. ทำข้อสามโดยใช้ list comprehension บรรทัดเดียว

```
li = [0.2, -1000, 1000, 33.21, -101.12, 0.01, 212, 0.4, -0.3, -100]
print([x for x in li if x <= -0.5 or x >= 0.5])
```

#### 5. จงเขียนโปรแกรมเพื่อตรวจสอบว่าในกล่องมีสัตว์หรือไม่ โดยเติมโค้ดลงใต้ฟังก์ชัน check\_animals

```
>>> global animals
>>> animals = ['dogs', 'cats', 'birds', 'insects', 'rats']
>>> inbox1 = ['cars', 'keys', 'computers']
>>> inbox2 = ['chairs', 'cats', 'water']
>>> def check_animals(box):
    # ข้อสอบนักเรียนทำตรงนี้
```

ผลที่คาดว่าจะได้

```
>>> kk=check_animals(inbox2)
>>> kk
'cats'
>>> kk=check_animals(inbox1)
>>> kk
'not found'
>>>
```

```
global animals
animals = ['dogs', 'cats', 'birds', 'insects', 'rats']
inbox1 = ['cars', 'keys', 'computers']
inbox2 = ['chairs', 'cats', 'water']
def check_animals(box):
    found = False
    for animal in box:
        if animal in animals:
            return animal
        found = True
    if not found:
        return ("not found")

kk = check_animals(inbox1)
print(kk)
```

### C. Dict Comprehension

1)

สวิตช์โยกชนิดหนึ่งมีสถานะแค่ เปิด หรือ ปิด. ถ้าสวิตช์อยู่ในสถานะ เปิด แล้ว สัญญาณขาออกจะเป็น1 ถ้าสวิตช์อยู่ในสถานะปิดแล้ว สัญญาณขาออกจะเป็น0. สวิตช์โยกจะถูกรับรู้ว่าถูกโยกถ้าสัญญาณเปลี่ยนจาก 0ไป1 หรือ 1ไป0 มีเช่นนั้น จะถูกรับรู้ว่าสวิตช์โยกไม่ได้ถูกโยก

ให้สัญญาณขาออกเป็นลำดับของ0,กับ1หลายๆตัว จงหาจำนวนครั้งที่สวิตช์ถูกโยก เช่น 100111001 มีจำนวนการโยกคือ 1-0, 0-1, 1-0, 0-1 รวม 4 ครั้ง

```
a = input()
count = 0
for i in range(len(a)-1):
    if a[i] != a[i+1]:
        count += 1
print(count)
```

2) จงนำคำตอบข้อหนึ่งมาเขียนฟังก์ชันเพื่อนับจำนวนการโยกของสวิตช์แล้วใส่ใน **dictionary** ใหม่ชื่อ **temp\_count** โดยใช้ **dictionary comprehension** จากข้อมูลดิบที่มาในรูป **dictionary** โดยใช้คีย์เดิม ตัวอย่างเช่น

```
temp={'t1':1010101,'t2':1100110110}
```

```
temp_count= somefunc(temp)
```

จะได้ผลลัพธ์

```
>>>print(temp_count)
```

```
{'t1': 6, 't2': 5}
```

```
def somefunc(temp):
    def count(a):
        a = str(a)
        count = 0
        for i in range(len(a)-1):
            if a[i] != a[i+1]:
                count += 1
        return count
    x = { e : count(temp[e]) for e in temp }
    return x
```

```
temp = {'t1': 1010101, 't2': 1100110110}
temp_count = somefunc(temp)
print(temp_count)
```

D.

<https://docs.python-guide.org/writing/gotchas/>

1) จงอธิบายพร้อมยกตัวอย่าง default mutable arguments ที่ทำให้เกิด bug พร้อมอธิบายวิธีป้องกัน

**bug**

```
def append_to(element, to=[]):
    to.append(element)
    return to
```

สลับ Object ทุกครั้งที่เรียก function ให้ออกมา

**edit**

```
def append_to(element, to=None):
    if to is None:
        to = []
    to.append(element)
    return to
```

2) จงอธิบายว่าเมื่อไหร่ควรใช้ Generator. Generator มีประโยชน์อย่างไร

Generator เป็น Iterate ประเภทหนึ่งที่มา

Generate คิวของค่าที่เราเรียกแบบมาทีละตัว ใช้เวลาที่เรารอตัวมาจนจบ  
เพื่อเอาค่าออกมาทีละค่าเวลาตัวมันกำลังจะผ่านไปเรื่อยๆ

3) จงอธิบายความแตกต่างระหว่าง Iterator กับ Generator

Iterator เป็น Object ที่พบบ่อยใน Python เช่น list, tuple, set, และ dict. เราสามารถใช้ iter() เพื่อสร้าง iterator ได้. Generator คือฟังก์ชันที่ใช้ yield เพื่อสร้างค่าและจะทำงานไปทีละขั้นตอนในลำดับที่เรากำหนด

E. Nested List to dict comprehension. จงใช้ dict comprehension เพื่อแก้ nested list โดยยกเว้น key ที่มีค่าว่าง เช่น

```
a = [['k', [1, 2, 3]], ['j', [-1, -2, -3, -4]], ['m', []]]
```

ให้เป็น

```
{'k': [1, 2, 3], 'j': [-1, -2, -3, -4]}
```

```
a = [['k', [1, 2, 3]], ['j', [-1, -2, -3, -4]], ['m', []]]
q = {i[0]: i[1] for i in a if len(i[1]) > 0}
print(q)
```