

# List Processing

ภาควิชาวิศวกรรมคอมพิวเตอร์  
จุฬาลงกรณ์มหาวิทยาลัย

๒๕๖๒

# List

List คือ รายการของข้อมูล

[ ] ← empty list

[2, 3, 5, 7, 11, 13, 17, 19, 23]

["SU", "MO", "TU", "WE", "TH", "FR", "SA"]

[ ["Ranee", "Campen"],

["Roy Marn", "Plerng Boon",

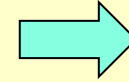
"Bubphe Sanniwat", "Krong Kam"] ]

# Basic List Operations

- Length `len([1,2,3])`
- Concatenation `[1] + [2,3] → [1,2,3]`
- Repetition `[0,0] * 3 → [0,0,0,0,0,0]`
- Indexing `x[k]`
- Slicing `x[start : stop : step]`

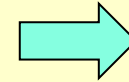
# แบบฝึกหัด: เลขใดหายไป

0110-02234-9877-0112



5, 6

9 8 7 6 5 4 3 2 1 0



None

```
d = input()
counts = [0] * 10
for c in d:
    if "0" <= c <= "9":
        counts[int(c)] += 1
missing = ""

if missing == "":
    print("None")
else:
    print(missing)
```

ต้องการให้ counts[k]  
เก็บจำนวนที่มีเลข k ปรากฏใน input

หยิบทีละตัวใน input  
ถ้าเป็นเลข ก็ให้เพิ่ม  
จำนวนที่พบอีก 1

นำเลขที่มี counts ของ  
เลขนั้นเป็น 0 มาต่อให้  
สตริง missing

# Basic List Methods

append, remove, insert, pop, index, sort

```
d = []
for i in range(5):
    d.append(10*i)          # [0,10,20,30,40]
d.remove(20)               # [0,10,30,40]
d.insert(2,99)             # [0,10,99,30,40]
d.insert(-1,7)            # [0,10,99,30,7,40]
x = d.pop(1)              # [0,99,30,7,40], x = 10
x = d.pop(-3)             # [0,99,7,40], x = 30
j = d.index(99)           # j = 1
d.sort()                  # [0,7,40,99]
b = 40 in d               # b = True
```

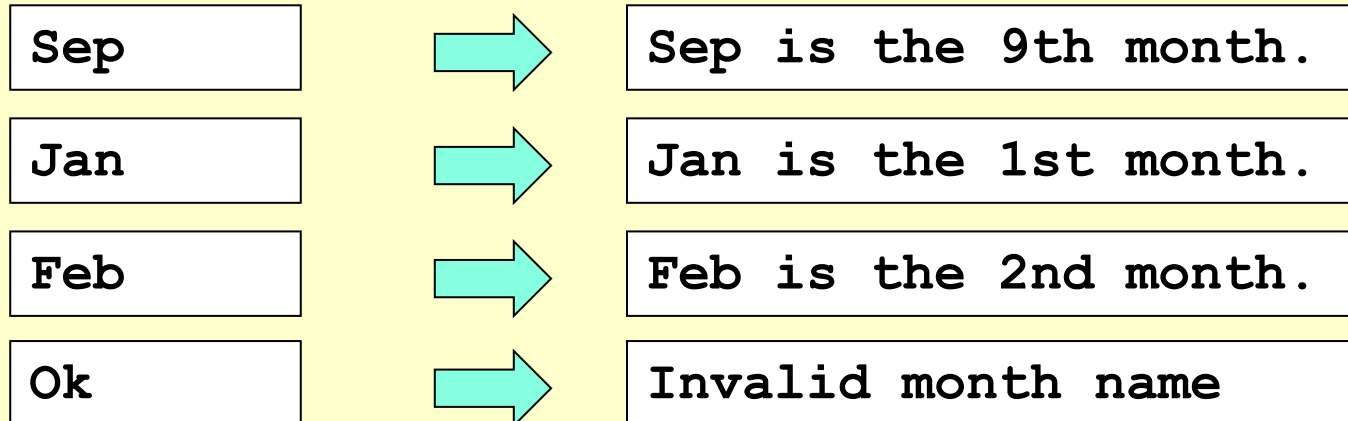
ต่อท้ายลิสต์

d.remove(e), d.index(e) ถ้าหา e ไม่พบใน d จะเกิดข้อผิดพลาด

# ตัวอย่าง: หาเลขเดือนจากชื่อเดือน

```
months = ["Jan", "Feb", "Mar", "Apr", "May", "Jun",  
          "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"]  
t = input()  
if t in months:  
    k = months.index(t)  
    th = "th"  
    if k == 0: th = "st"  
    if k == 1: th = "nd"  
    print(t, "is the", str(k+1)+th+" month.")  
else:  
    print("Invalid month name")
```

ก่อนใช้ index ต้อง  
มั่นใจว่ามีข้อมูลที่จะค้น



# แบบฝึกหัด: ชื่อเล่นอะไร ชื่อจริงอะไร

เขียนโปรแกรม รับชื่อจริงแสดงชื่อเล่น  
รับชื่อเล่นแสดงชื่อจริง โดยไม่ต้องใช้คำสั่ง if

Robert  $\longleftrightarrow$  Dick

William  $\longleftrightarrow$  Bill

James  $\longleftrightarrow$  Jim

John  $\longleftrightarrow$  Jack

Margaret  $\longleftrightarrow$  Peggy

Edward  $\longleftrightarrow$  Ed

Sarah  $\longleftrightarrow$  Sally

Andrew  $\longleftrightarrow$  Andy <http://www.cc.kyoto-su.ac.jp/~trobb/nicklist.html>

# รูปแบบการรับข้อมูลเก็บในลิสต์

- ระบุจำนวน ตามด้วยข้อมูลบรรทัดละตัว

```
4
10.2
11.5
20.0
22.5
```

- รับข้อมูลบรรทัดละตัว มีค่าระบุทั้งหมด

```
10.2
11.5
20.0
22.5
-1
```

- รับรายการข้อมูลในบรรทัดเดียว

```
10.2 11.5 20.0 22.5
```



# แบบที่ 1: ระบุนำนวน ตามด้วยข้อมูลบรรทัดละตัว

input

4

10.2

11.5

20.0

22.5

## แบบที่ 2: รับข้อมูลบรรทัดละตัว มีค่าระบุทั้งหมด

input

10.2  
11.5  
20.0  
22.5  
-1

# แบบที่ 3: รับรายการข้อมูลในบรรทัดเดียว

`input`

`10.2 11.5 20.0`

# ข้อแนะนำ: การต่อท้ายลิสต์

```
data.append( x )
```



```
data = data + [ x ]
```



```
data += [ x ]
```



# แบบฝึกหัด: เพิ่มหลัง-เพิ่มหน้า

input	
	4
เพิ่มหลัง	1
หน้า	2
หลัง	3
หน้า	4
หลัง หน้า หลัง หน้า หลัง	11 12 13 14 15
หน้า	21
หลัง	22
หน้า	23
หลัง	24
หน้า	25
	-1

เพิ่มข้อมูลเป็นจำนวน  
ตามที่บอกในบรรทัดแรก

← เพิ่มข้อมูลทุกตัวในบรรทัดนี้

เพิ่มข้อมูลทุกตัวเจอ -1

output

[25, 23, 21, 14, 12, 4, 2, 1, 3, 11, 13, 15, 22, 24]

# รูปแบบการหยิบข้อมูลที่ละตัวในลิสต์มาประมวลผล

```
for e in d:  
    ...
```



หยิบข้อมูลที่ละตัว  
ในลิสต์มาใช้งาน

```
for i in range(len(d)) :  
    e = d[i] * i  
    ...
```



ถ้าต้องการทั้ง  
ตำแหน่งและข้อมูล

```
for e in d:  
    i = d.index(e)  
    ...
```




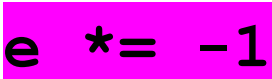
อย่าเขียนแบบนี้  
ซ้ำเปล่า ๆ  
ใช้แบบที่ 2


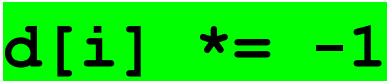
## ตัวอย่าง: นับจำนวนนิสิตวิชาฯ ในลิสต์

```
student_ids = input().split()
# ["6131022921", "6240123026", "6130021221", ...]
c = 0
for sid in student_ids:
    if sid[-2:] == "21":
        c += 1
print(c)
```

```
student_ids = input().split()
c = 0
for i in range(len(student_ids)):
    if student_ids[i][-2:] == "21":
        c += 1
print(c)
```

# ถ้าต้องการเปลี่ยนข้อมูลบางตัวในลิสต์

```
...  
# เปลี่ยนจำนวนติดลบทั้งหมดในลิสต์ d ให้เป็นบวก  
for e in d:  
    if e < 0:  
  e *= -1 # ค่าใน e เปลี่ยน แต่ไม่เปลี่ยนค่าใน d
```

```
...  
# เปลี่ยนจำนวนติดลบทั้งหมดในลิสต์ d ให้เป็นบวก  
for i in range(len(d)) :  
     if d[i] < 0:  
         d[i] *= -1 # เปลี่ยนค่าใน d ตามที่ต้องการ
```



# รูปแบบการหีบข้อมูลที่ติดกันในลิสต์มาประมวลผล

```
for i in range(len(d)-1):  
    left = d[i]  
    right = d[i+1]  
    ...
```

```
for i in range(1, len(d)):  
    left = d[i-1]  
    right = d[i]  
    ...
```

# ตัวอย่าง: ลำดับเพิ่มขึ้นหรือไม่

Input		Output
4 6 9 10 14 34 45 99	➡	True
2 3 4 5 5 5 5 5 99	➡	False

```
d = input().split()
is_increasing = True
for i in range(len(d)-1):
    if int(d[i]) >= int(d[i+1]):
        is_increasing = False
        break
print(is_increasing)
```

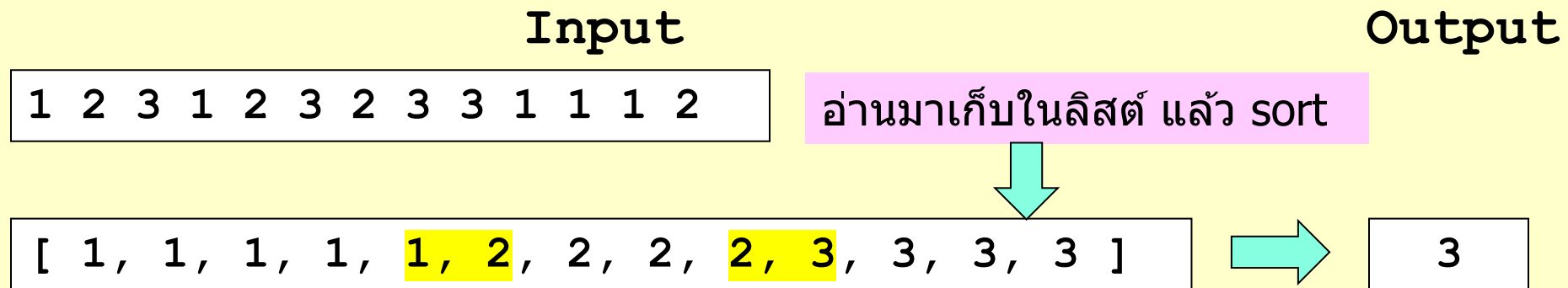
# แบบฝึกหัด: นับจำนวน "ยอด" (น้อย มาก น้อย)

Input																Output		
1	2	3	4	8	2	4	5	9	10	8	9	10	11	15	18	12	20	3
1	2	3	4	5	6	10	11	12	13	45	79	80	81	89	90			0

```
x = input().split()
d = []
for e in x:
    d.append(int(e))
count = 0

print(count)
```

# แบบฝึกหัด: ชุดข้อมูลมีแตกต่างกันกี่ตัว



```
x = input().split()
d = []
for e in x:
    d.append(int(e))
d.sort()
count = 1
```

วงวนเพิ่ม count ทุกครั้ง ที่พบตัวติดกันต่างกัน

```
print(count)
```

# รูปแบบการค้นข้อมูลในลิสต์

```
[ 'KTB' , 18.9 , 'BBL' , 176.5 , 'SCB' , 132.0 , 'KBANK' , 172.0 ]
```

```
[ 'KTB' , 'BBL' , 'SCB' , 'KBANK' ]
```

```
[ 18.9 , 176.5 , 132.0 , 172.0 ]
```

```
[ [ 'KTB' , 18.9 ] , [ 'BBL' , 176.5 ] , [ 'SCB' , 132.0 ] , [ 'KBANK' , 172.0 ] ]
```

# รูปแบบการค้นข้อมูลในลิสต์: กรณีลิสต์แบบง่าย

กรณีเป็นลิสต์ที่เก็บข้อมูลพื้นฐาน (ไม่ใช่ลิสต์ซ้อนลิสต์)

```
# x เป็นลิสต์ อยากค้นว่ามี c ใน x ไหม ?  
# x = [3,1,4,5,2,4,7], c = 5  
if c in x:  
    ...
```

```
# x เป็นลิสต์ อยากรู้ว่า c เก็บใน index ใดใน x  
if c in x:  
    j = x.index(c)
```

# รูปแบบการค้นข้อมูลในลิสต์: กรณีเก็บหลายลิสต์

เช่น มีลิสต์ `sid` เก็บเลขประจำตัว และลิสต์ `gpa` เก็บเกรด  
เกรดของ `sid[k]` เก็บอยู่ที่ `gpa[k]`

```
sid = [6131001021, 6130020221, ...]
```

```
gpa = [3.8, 3.7, ...]
```

ต้องการค้น ID ที่สนใจว่า ได้เกรดเท่าไร ?

1

```
if ID in sid:  
    j = sid.index(ID)  
    print("ID =", ID, gpa[j])  
else:  
    print("Not found")
```

# รูปแบบการค้นข้อมูลในลิสต์: กรณีลิสต์ซ้อนลิสต์

เช่น ลิสต์ที่เก็บ [ID, gpa]

data = [[6131001021, 3.8], [6130020221, 3.7], ...]

ต้องการค้น ID ที่สนใจว่า ได้เกรดเท่าไร ?

2

```
gpa = -1
for e in data : # e เป็นลิสต์ที่มี 2 ช่อง
    if e[0] == ID:
        gpa = e[1]
        break
if gpa == -1:
    print("Not found")
else:
    print("ID =", ID, gpa)
```



# รูปแบบการค้นข้อมูลในลิสต์: กรณีลิสต์ซ้อนลิสต์

เช่น ลิสต์ที่เก็บ [ID, gpa]

data = [[6131001021, 3.8], [6130020221, 3.7], ...]

ต้องการค้น ID ที่สนใจว่า ได้เกรดเท่าไร ?

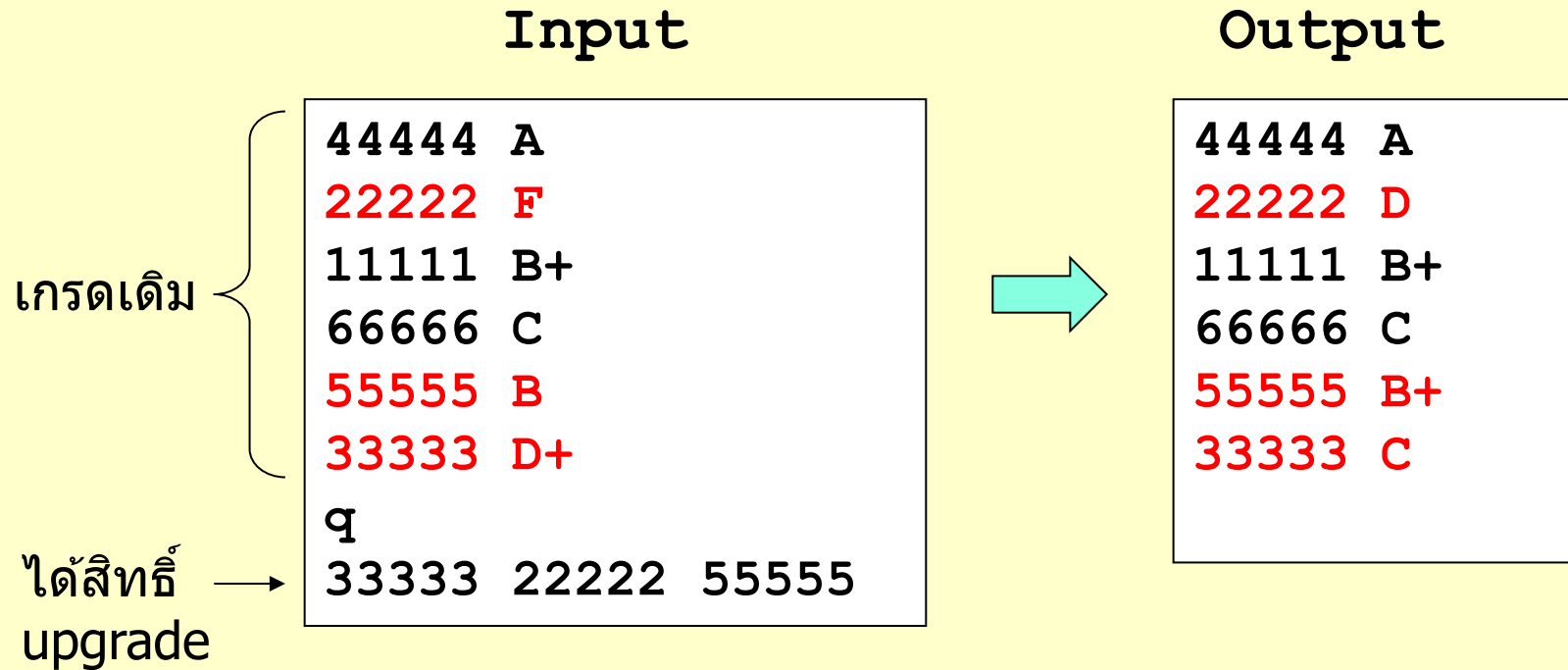
3

```
gpa = -1
for [sid, g] in data :
    if sid == ID:
        gpa = g
        break
if gpa == -1:
    print("Not found")
else:
    print("ID =", ID, gpa)
```

ดีกว่าแบบก่อนนี้

# แบบฝึกหัด: upgrade

เขียนโปรแกรมปรับเกรดเพิ่มให้นักเรียนที่ได้รับสิทธิ์คนละหนึ่งประจุ



# การเรียงลำดับข้อมูล

[90, 22, 44, 70, 20, 51, 12, 39, 67, 56, 88, 31, 72, 77, 53]

`d.sort()`

[12, 20, 22, 31, 39, 44, 51, 53, 56, 67, 70, 72, 77, 88, 90]

# การเรียงลำดับข้อมูล

[10, 10, 11, 12, 11, 13, 11, 11, 12, 10, 13, 10, 10, 13, 13]

**d.sort()**

[10, 10, 10, 10, 10, 11, 11, 11, 11, 12, 12, 13, 13, 13, 13]

# การเรียงลำดับข้อมูล: กรณียาง่าย

```
x = [3,1,4,5,2,4,7]
x.sort() # x = [1,2,3,4,4,5,7]
```

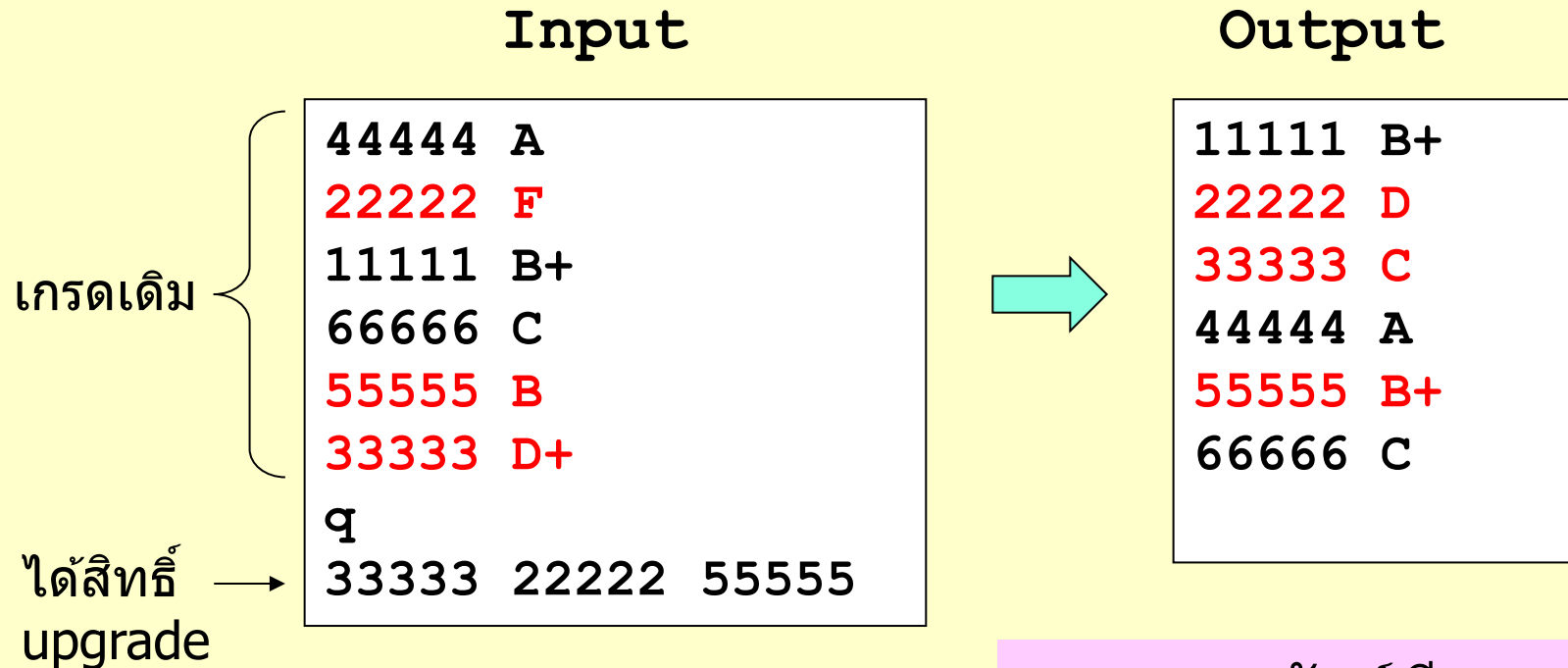
```
x = ["Tom", "Ann", "Don"]
x.sort() # x = ["Ann", "Don", "Tom"]
```

```
x = [["A",9], ["C",1], ["A",1]]
x.sort() # x = [["A",1], ["A",9], ["C",1]]
```

เรียงตามตัวแรกก่อน ถ้าตัวแรกเท่ากัน ให้เรียงตามตัวที่สอง

# แบบฝึกหัด: upgrade

เขียนโปรแกรมปรับเกรดเพิ่มให้นักเรียนที่ได้รับสิทธิ์คนละหนึ่งประจุ



แสดงผลลัพธ์เรียงตามเลข  
ประจำตัวจากน้อยไปมาก

# การเรียงลำดับข้อมูล: กรณีซับซ้อน

เรียงตามตัวแรกก่อน ถ้าตัวแรกเท่ากัน ให้เรียงตามตัวที่สอง

```
x = ["A",9], ["C",1], ["A",1]
x.sort() # x = ["A",1], ["A",9], ["C",1]
```

เรียงตามตัวที่สองก่อน ถ้าเท่ากัน ให้เรียงตามตัวแรก

```
x = ["A",9], ["C",1], ["A",1]
t = []
for [a1,a2] in x:
    t.append([a2,a1]) # สร้างลิสต์ใหม่ เก็บสลับตำแหน่ง
# t = [9,"A"], [1,"C"], [1,"A"]
t.sort()
# t = [1,"A"], [1,"C"], [9,"A"]
for i in range(len(t)): # สลับตำแหน่งกลับ ไปเก็บในลิสต์เดิม
    x[i][0],x[i][1] = t[i][1],t[i][0]
# x = ["A",1], ["C",1], ["A",9]
```

# ตัวอย่าง: เรียงลำดับสตริงตามความยาวสตริง

["ABC", "ABCD", "XYZ", "OK"]

เรียงแล้วได้

["OK", "ABC", "XYZ", "ABCD"]

```
x = ["ABC", "ABCD", "XYZ", "OK"]
t = []
for e in x:
    t.append([len(e), e])
# t = [[3, "ABC"], [4, "ABCD"], [3, "XYZ"], [2, "OK"]]
t.sort()
# t = [[2, "OK"], [3, "ABC"], [3, "XYZ"], [4, "ABCD"]]
for i in range(len(t)):
    x[i] = t[i][1]
# x = ["OK", "ABC", "XYZ", "ABCD"]
```

สร้างลิสต์ใหม่ เก็บความยาวด้วย

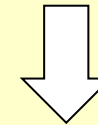


# แบบฝึกหัด: จุดใดใกล้จุดกำเนิดเป็นอันดับสาม

```
n = int(input())
points = []
for k in range(n):
    d = input().split()
    x = float(d[0])
    y = float(d[1])
```

อ่านพิกัดต่าง ๆ เก็บใส่ลิสต์

5  
2.0 2.0  
2.0 3.0  
2.0 2.0  
2.0 5.0  
-1 -5.0

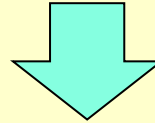


#2: (2.0, 3.0)

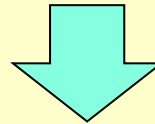
กำหนดให้พิกัดที่ได้รับทั้งหมดมีระยะถึงจุดกำเนิดไม่เท่ากันเลย

# List $\leftrightarrow$ String ด้วย split & join

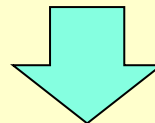
```
"ABC DEF GHI JK".split()
```



```
["ABC", "DEF", "GHI", "JK"]
```

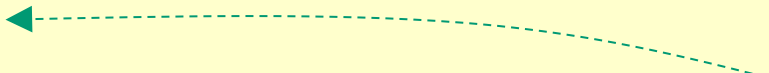


```
" , ".join( ["ABC", "DEF", "GHI", "JK"] )
```



```
"ABC, DEF, GHI, JK"
```

# split แยกสตริงออกเป็นลิสต์

- **`x = s.split()`** 
  - แยกสตริง `s` ออกเป็นส่วน ๆ ได้เป็นลิสต์ (แยกด้วยช่องว่าง)
  - `s = "11 2 33"`
  - `s.split()` ได้ `["11", "2", "33"]`
- **`x = s.split( sep )`**
  - แยกสตริง `s` ออกเป็นส่วน ๆ ได้เป็นลิสต์ ใช้ `sep` เป็นตัวแยก
  - `s = "11:2: 33"`
  - `s.split(":")` ได้ `["11", "2", " 33"]`

```
"a,,,b".split(",")   ได้ ["a", "", "", "b"]  
"a  b".split(" ")   ได้ ["a", "", "", "b"]  
"a  b".split()      ได้ ["a", "b"]  
"a,,,b".split(",,") ได้ ["a", ",b"]
```

# join นำสตริงในลิสต์มาต่อกัน

- `s = sep.join( x )`
  - `sep` เป็น string
  - `x` เป็น list of strings
  - `sep.join(x)` ได้ผลเป็น string ที่สร้างจากการนำ string แต่ละตัวใน `x` มาต่อกัน โดยมี `sep` เป็นตัวคั่น

```
x = ["A", "BC", "DEF", "GH"]  
s1 = "".join(x)    # "ABCDEF GH"  
s2 = ",".join(x)    # "A,BC,DEF,GH"  
s3 = "X".join(x)    # "AXBCXDEFXGH"
```

## ตัวอย่าง:

```
x = [-1,0,3,300,-2,39,50,12,400,-100]
# เลือกและแสดงข้อมูลใน x เฉพาะที่มีค่า 0 ถึง 255
y = []
for e in x:
    if 0 <= e <= 255:
        y.append( str(e) )
# y = ["0", "3", "39", "50", "12"]
print( " -> ".join(t) )
```

0 -> 3 -> 39 -> 50 -> 12

# แบบฝึกหัด: Collatz Problem

```
n = int(input())
while n != 1:
    if n%2 == 0:
        n //= 2
    else:
        n = 3*n + 1
```

อยากรู้ว่า  $n$  เปลี่ยนแปลง  
อย่างไรจนกลายเป็น 1

input

10



output

10->5->16->8->4->2->1


## ข้อควรระวัง: การแก้ไขลิสต์ภายในวงวน

```
x = [1,1,2,3,1]
for e in x:
    if e == 1:
        x.remove(e)
print(x)
```

รอบที่ 1	x = [1,1,2,3,1]	ได้ e = 1, x.remove(1)
รอบที่ 2	x = [1,2,3,1]	ได้ e = 2
รอบที่ 3	x = [1,2,3,1]	ได้ e = 3
รอบที่ 4	x = [1,2,3,1]	ได้ e = 1, x.remove(1)
	x = [2,3,1]	ผิด

# ข้อควรระวัง: การแก้ไขลิสต์ภายในวงวน

```
x = [1,1,2,3,1]
for i in range(len(x)):
    if x[i] == 1:
        x.pop(i)
```



```
x = [1,1,2,3,1]
for i in [0,1,2,3,4]:
    if x[i] == 1:
        x.pop(i)
```

i = 0, x = [1,1,2,3,1]	x.pop(0)
i = 1, x = [1,2,3,1]	
i = 2, x = [1,2,3,1]	
i = 3, x = [1,2,3,1]	x.pop(3)
i = 4, x = [1,2,3]	เจ๊ง ไม่มี x[4]



# ข้อควรระวัง: การแก้ไขลิสต์ภายในวงวน

```
x = [1,1,2,3,1]
i = 0
while i < len(x):
    if x[i] == 1:
        x.pop(i)
    i += 1
print(x)
```

```
x = [1,1,2,3,1]
i = 0
while i < len(x):
    if x[i] == 1:
        x.pop(i)
    else:
        i += 1
print(x)
```

ถูก

```
i = 0, x = [1,1,2,3,1]    x.pop(0)
i = 1, x = [1,2,3,1]
i = 2, x = [1,2,3,1]
i = 3, x = [1,2,3,1]    x.pop(3)
                        x = [1,2,3]    ผิด
```

# ข้อควรระวัง: การแก้ไขลิสต์ภายในวงวน

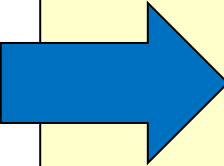
```
x = [1,3]
for e in x:
    if e == 1:
        x.insert(0,9)
```

รอบที่ 1	x = [1,3]	ได้ e = 1, x.insert(0,9)
รอบที่ 2	x = [9,1,3]	ได้ e = 1, x.insert(0,9)
รอบที่ 3	x = [9,9,1,3]	ได้ e = 1, x.insert(0,9)
รอบที่ 4	x = [9,9,9,1,3]	ได้ e = 1, x.insert(0,9)
... ทำงานไม่สิ้นสุด		

สรุป: อย่าแก้ไขลิสต์ภายในวงวนเลย

# อยากแก้ไขลิสต์ภายในวงวน → สร้างลิสต์ใหม่

```
x = [1,1,2,3,1]
for e in x:
    if e == 1:
        x.remove(e)
print(x)
```



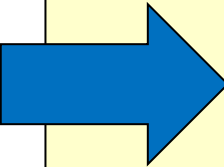
```
x = [1,1,2,3,1]
y = []
for e in x:
    if e != 1:
        y.append(e)
x[:] = y
```

`x[:]` คือที่เก็บในลิสต์ `x` ตั้งแต่ซ้ายถึงขวาสุด

`x[:] = y` คือ copy ข้อมูลใน `y` ไปเก็บในลิสต์ `x` (`x` และ `y` เป็นคนละลิสต์)

`x = y` ทำให้ตัวแปร `x` ไปเป็นตัวเดียวกับลิสต์ `y`

```
x = [1,3]
for e in x:
    if e == 1:
        x.insert(0,9)
```



```
x = [1,3]
y[:] = x
for e in x:
    if e == 1:
        y.insert(0,9)
```