

การทำงานแบบวงวน

ภาควิชาวิศวกรรมคอมพิวเตอร์

จุฬาลงกรณ์มหาวิทยาลัย

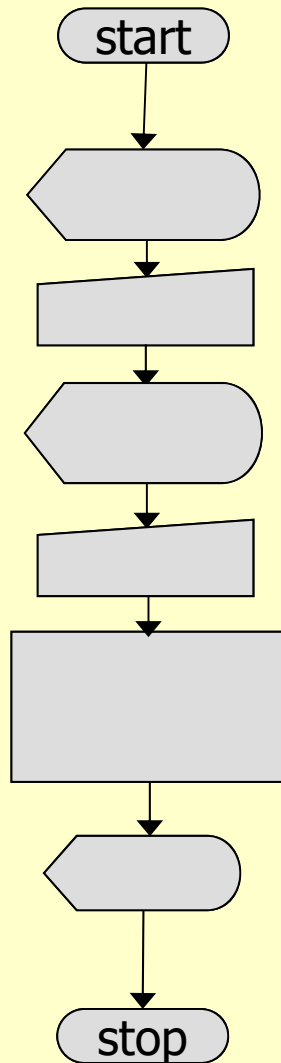
๒๕๖๒

หัวข้อ

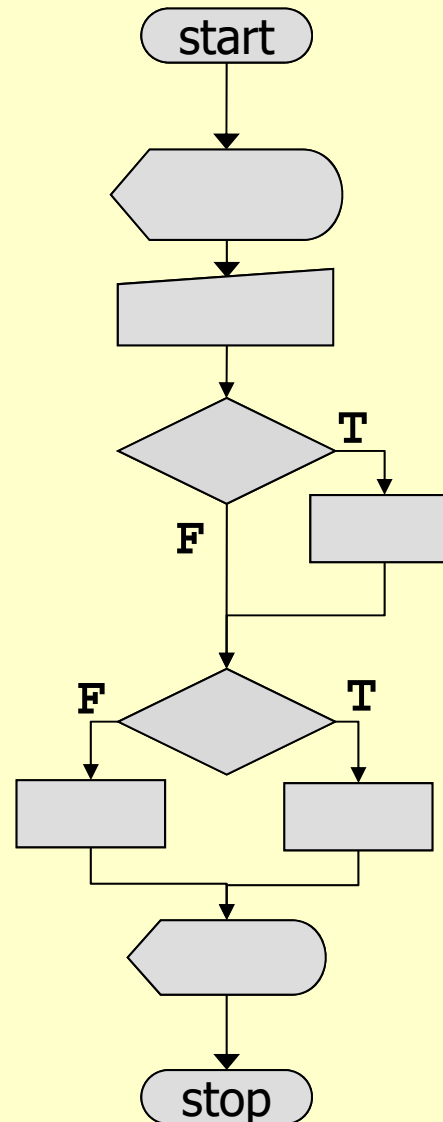
- while
- for
 - for k in range(...)
 - for ch in string
 - for elem in a_list
- break
- nested loops

ผังงาน (Flowchart)

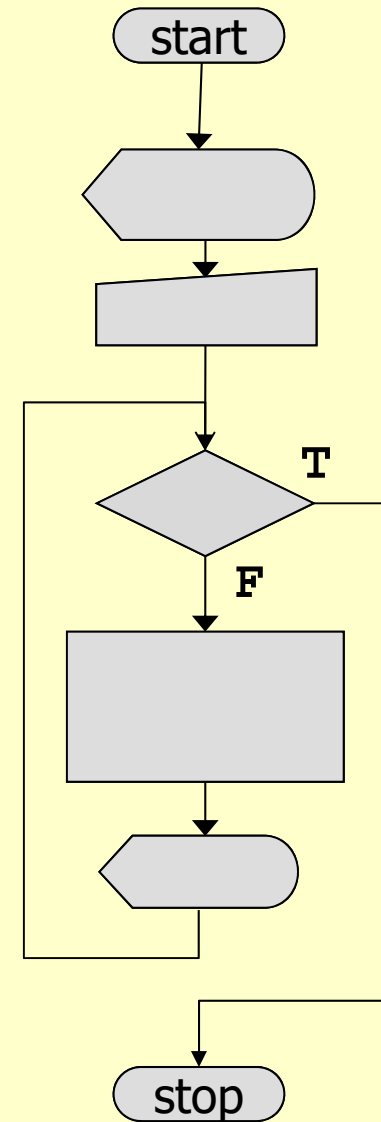
แบบลำดับ



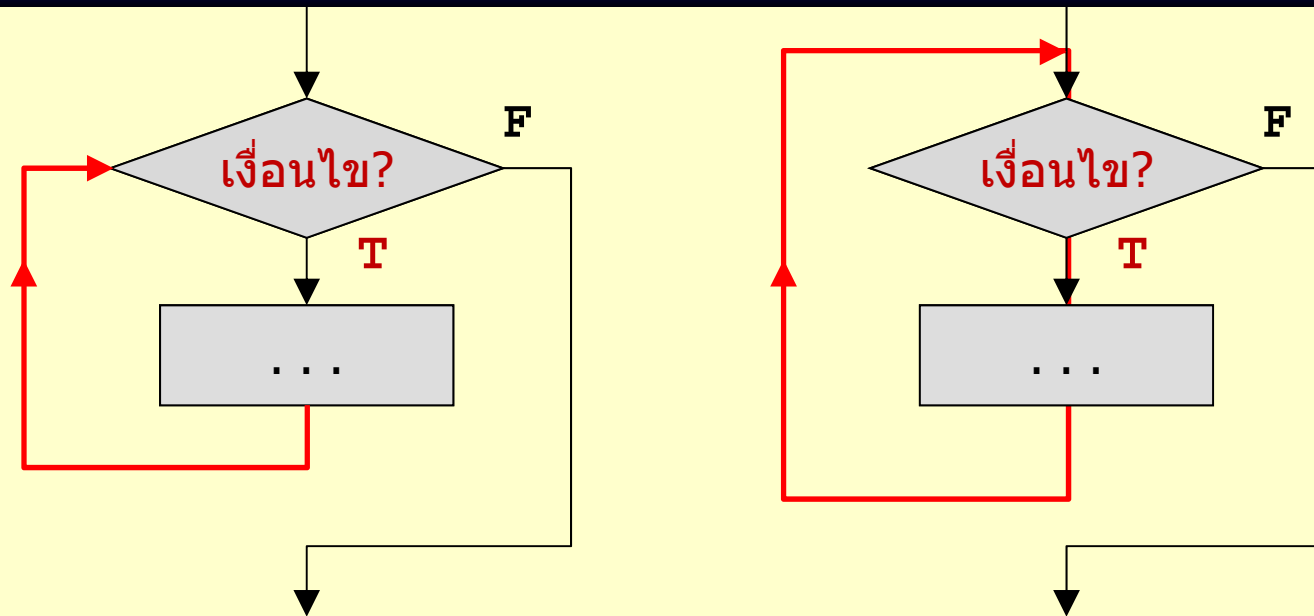
เลือกทำ



วงวน



วงวน : while



ต้องมีเครื่องหมาย :

while เงื่อนไข :

กลุ่มคำสั่งที่ทำงานเมื่อเงื่อนไขเป็นจริง

คำสั่งในกลุ่มต้องเยื้องเข้าไปทางขวา ให้ตรงกันหมด

ตัวอย่าง:

$$\sum_{k=0}^4 (2k - 1)^2$$

1

```
s = 0
s += (2*0 - 1)**2
s += (2*1 - 1)**2
s += (2*2 - 1)**2
s += (2*3 - 1)**2
s += (2*4 - 1)**2
print(s)
```

2

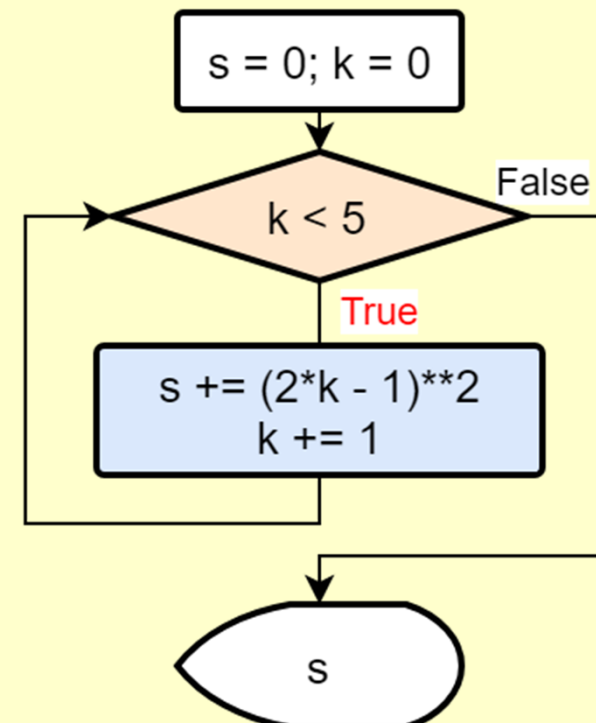
```
s = 0; k = 0
s += (2*k - 1)**2; k += 1
s += (2*k - 1)**2; k += 1
s += (2*k - 1)**2; k += 1
s += (2*k - 1)**2; k += 1
s += (2*k - 1)**2;
print(s)
```

3

```
s = 0; k = 0

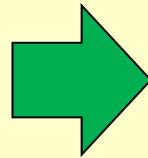
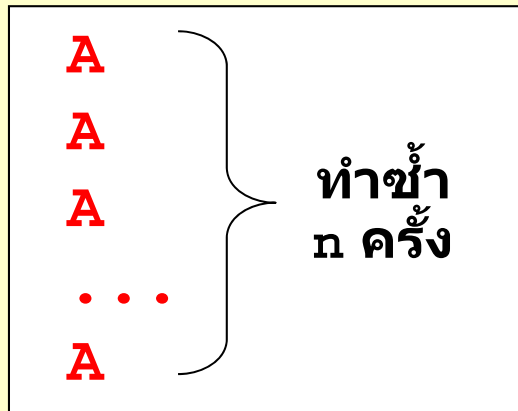
while k < 5 :
    s += (2*k - 1)**2
    k += 1

print(s)
```



หมุนซ้ำ 5 รอบ k เริ่มที่ 0 แต่ละรอบเพิ่ม 1
เมื่อ k เป็น 5 เงื่อนไขเป็นเท็จ ออกจากวงวน

รูปแบบ: วงวนเพื่อทำชุดคำสั่งซ้ำ ๆ



```
k = 0  
while k < n:  
    A  
    k += 1
```

```
k = 1  
while k <= n:  
    A  
    k += 1
```

ตัวอย่าง: หาค่าน้อยสุดจากข้อมูล 5 ตัว

```
min_v = float(input())
v = float(input())
if v < min_v:
    min_v = v
v = float(input())
if v < min_v:
    min_v = v
v = float(input())
if v < min_v:
    min_v = v
v = float(input())
if v < min_v:
    min_v = v
print("min =", min_v)
```

```
min_v = float(input())
k = 0
while k < 4:
    v = float(input())
    if v < min_v:
        min_v = v
    k += 1
print("min =", min_v)
```

ตัวอย่าง: หาค่าน้อยสุดจากข้อมูลหลายตัว

```
n = int(input())
min_v = float(input())
k = 0
while k < n-1:
    v = float(input())
    if v < min_v:
        min_v = v
    k += 1
print("min =", min_v)
```

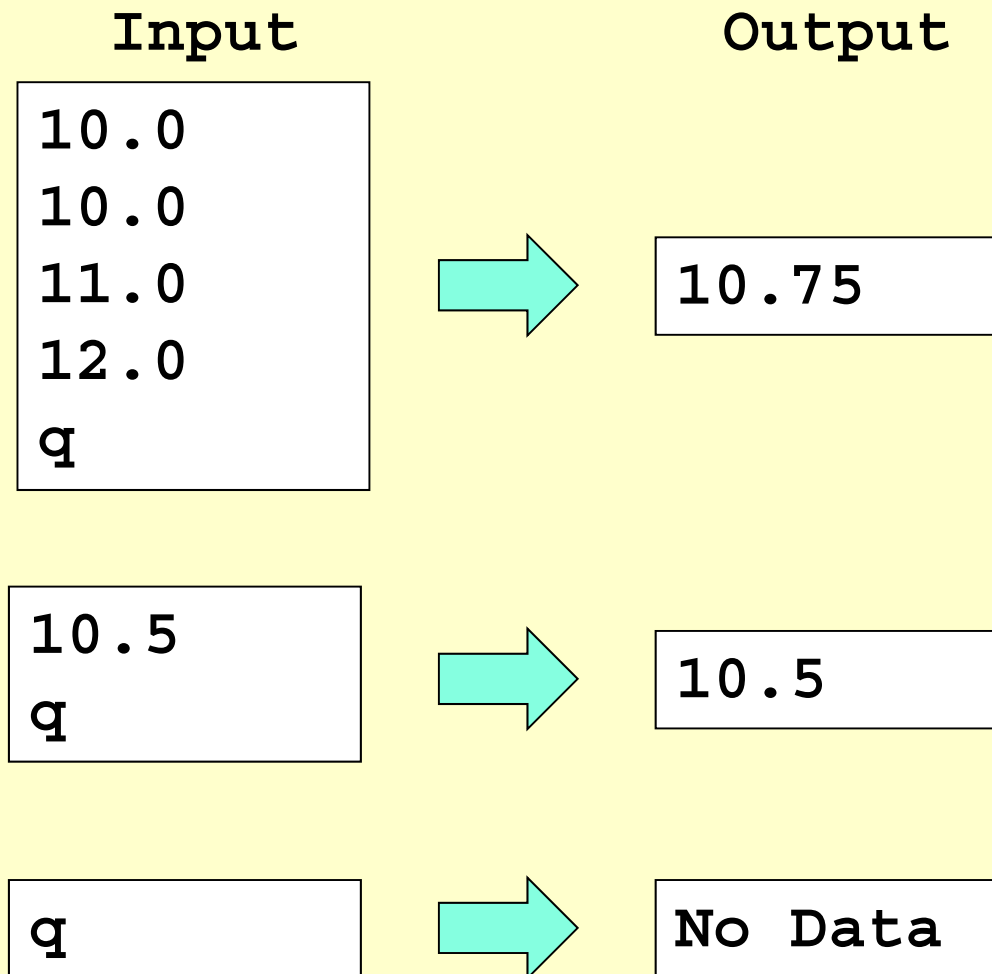
```
min_v = float(input())
d = input()
while d != "q":
    v = float(d)
    if v < min_v:
        min_v = v
    d = input()
print("min =", min_v)
```

4 ← บอกก่อนว่า
จะต้องรับกี่ตัว

10.0
11.2
15.5
12.4

10.0
11.2
15.5
12.4
q ← รับจนจบที่ q

แบบฝึกหัด: หาค่าเฉลี่ยจากข้อมูลหลายตัว



ตัวอย่าง: หาค่ารากที่สองด้วยวิธี bisection

- ต้องการหารากที่สองของ a
- ให้ $L = 0, U = a$
- ค่าตอบอยู่ใน $[L, U]$
- $x =$ จุดกึ่งกลางของช่วง
- ทำซ้ำล่างนี้ซ้ำ ถ้า $x^2 \neq a$
 - ถ้า $x^2 > a$:
เปลี่ยนช่วงเป็น $[L, x]$
 - ถ้า $x^2 < a$:
เปลี่ยนช่วงเป็น $[x, U]$
 - $x =$ จุดกึ่งกลางของช่วง

a = 25		
L	U	x
0	25	12.5
0	12.5	6.25
0	6.25	3.125
3.125	6.25	4.6875
4.6875	6.25	5.46875
4.6875	5.46875	5.078125
4.6875	5.078125	4.882813
4.882813	5.078125	4.980469
4.980469	5.078125	5.029297
4.980469	5.029297	5.004883

$$x = \frac{L + U}{2}$$

ตัวอย่าง: หาค่ารากที่สองด้วยวิธี bisection

- ต้องการหารากที่สองของ a
- ให้ $L = 0, U = a$
- คำตอบอยู่ใน $[L, U]$
- x = จุดกึ่งกลางของช่วง
- ทำซ้ำลํางนี้ซ้ำ ถ้า $x^2 \neq a$
 - ถ้า $x^2 > a$:
เปลี่ยนช่วงเป็น $[L, x]$
 - ถ้า $x^2 < a$:
เปลี่ยนช่วงเป็น $[x, U]$
 - x = จุดกึ่งกลางของช่วง

```
a = float(input())
L = 0; U = a
x = (L + U) / 2
while x**2 != a:
    if x**2 > a:
        U = x
    else:
        L = x
    x = (L + U) / 2
print(x)
```

ตัวอย่าง: หาค่ารากที่สองด้วยวิธี bisection

```
a = float(input())
L = 0; U = a
x = (L + U) / 2
while x**2 != a:
    if x**2 > a:
        U = x
    else:
        L = x
    x = (L + U) / 2
print(x)
```

มีปัญหา

```
a = float(input())
L = 0; U = a
x = (L + U) / 2
while ยังไม่ใกล้กัน(x**2, a):
    if x**2 > a:
        U = x
    else:
        L = x
    x = (L + U) / 2
print(x)
```

a กับ b ยังไม่ใกล้กันเมื่อ $|a - b| > \epsilon \max(a, b)$

ให้ a กับ b เป็นบวก, ϵ และ 10^{-9}

```
abs(a-b) > 1e-9*max(a,b)
```

แบบฝึกหัด: หา $\log_{10} a$ ด้วย bisection

เขียนโปรแกรมรับ a เพื่อหา $\log_{10} a$ ด้วย bisection
โดยที่ $1 \leq a \leq 600$

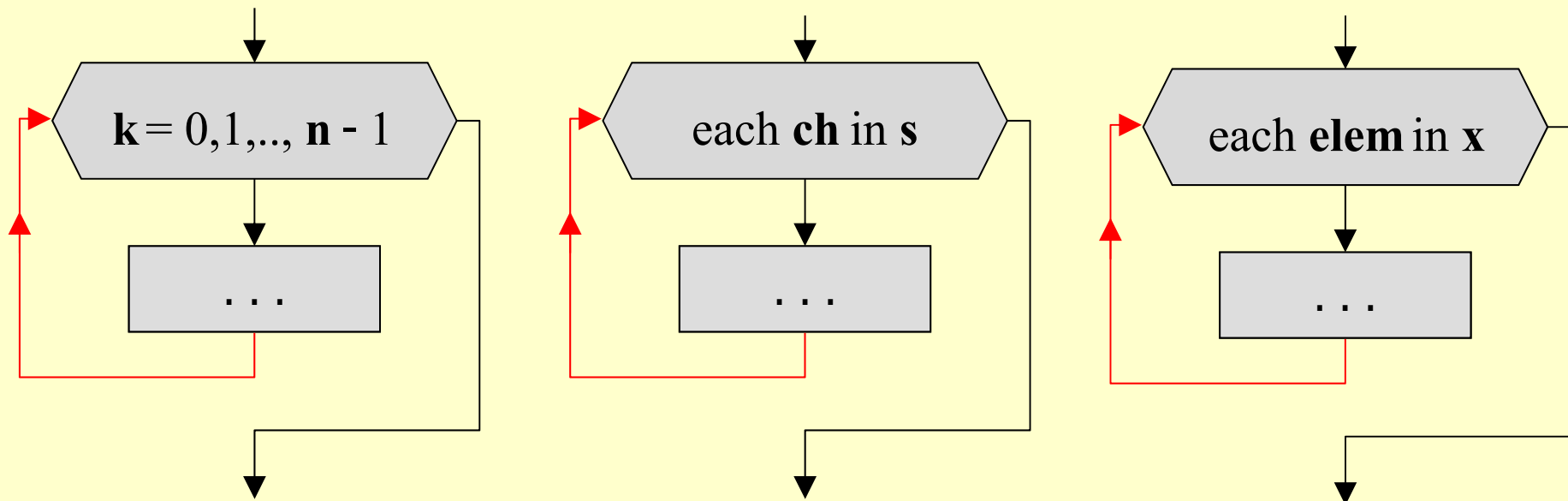
ถ้าต้องการให้ท้าทายขึ้น ก็ให้ a เกิน 600 มาก ๆ ได้

วงวน for (3 รูปแบบ)

```
for i in range(start, stop, step) :  
    ...
```

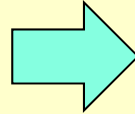
```
for each_char in a_string :  
    ...
```

```
for each_elem in a_list :  
    ...
```



แบบที่ 1: for k in range(start, stop, step)

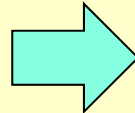
```
k = 0
while k < 100 :
    ...
    k += 1
```



```
for k in range(100) :
    ...
```

k = 0, 1, 2, ..., 99

```
k = 5
while k < 100 :
    ...
    k += 1
```

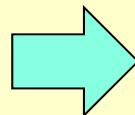


```
for k in range(5, 100) :
```

...

k = 5, 6, 7, ..., 99

```
k = 4
while k < 100 :
    ...
    k += 2
```

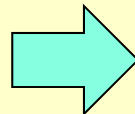


```
for k in range(4, 100, 2) :
```

...

k = 4, 6, 8, ..., 98

```
k = 100
while k > 0 :
    ...
    k += -1
```



```
for k in range(100, 0, -1) :
```

...

k = 100, 99, 98, ..., 1

อ่านเข้าใจได้ง่ายกว่า

ตัวอย่าง: while กับ for

```
s = 0
k = 0
while k < 5 :
    s += (2*k - 1)**2
    k += 1

print(s)
```

```
s = 0
for k in range(0,5,1):
    s += (2*k - 1)**2

print(s)
```

```
min_v = float(input())
k = 0
while k < 4:
    v = float(input())
    if v < min_v:
        min_v = v
    k += 1

print("min =", min_v)
```

```
min_v = float(input())
for k in range(4):
    v = float(input())
    if v < min_v:
        min_v = v

print("min =", min_v)
```


ตัวอย่าง: μ และ σ

```
N = int(input())
x = [0.0]*N
for i in range(N):
    x[i] = float(input())

s = 0
for i in range(N):
    s += x[i]
mean = s / N

s2 = 0
for i in range(N):
    s2 += (x[i]-mean)**2
sd = (s2/n)**0.5

print(mean, sd)
```

input

5
10
11
13
10
12

$$\mu = \frac{\sum_{i=1}^N x_i}{N}$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \mu)^2}{N}}$$

ตัวอย่าง: Dot Product $u \cdot v$

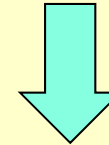
```
x = input().split()
u = [0.0]*len(x)
for i in range(len(x)) :
    u[i] = float(x[i])

x = input().split()
v = [0.0]*len(x)
for i in range(len(x)) :
    v[i] = float(x[i])

dot = 0
for i in range(len(x)) :
    dot += u[i]*v[i]

print( dot )
```

1	2	0	2	1
2	2	1	2	2



12.0

แบบฝึกหัด: ตรวจสอบคำตอบ

```
sol = input() # e.g., ABBBAAABCCBABABDCCDA  
ans = input() # e.g., ABBBAABBCCBAABBDCCDB
```

นับว่า sol กับ ans มีตัวตรงกันกี่ตัว

แบบที่ 2: for each_character in a_string

```
s = "Loop is Live"  
for c in s:  
    print(c)
```

หยิบตัวอักษรจาก s ใส่ c
รอบละหนึ่งตัวจากซ้ายไปขวา

	output
L	
o	
o	
p	
i	
s	
L	
i	
v	
e	

ตัวอย่าง: นับจำนวนตัวเลขในสตริง

```
s = input()
digit_counts = 0
for i in range(len(s)):
    if "0" <= s[i] <= "9":
        digit_counts += 1
print(digit_counts)
```

```
s = input()
digit_counts = 0
for ch in s:
    if "0" <= ch <= "9":
        digit_counts += 1
print(digit_counts)
```

แบบนี้ง่ายกว่า

ตัวอย่าง: สร้างสตริงที่ไม่มีเครื่องหมาย ([{ }])

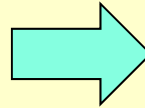
```
s = input()
result = ""
for ch in s:
    if ch not in "([{}])":
        result += ch
print(result)
```

หยิบออกมาทีละตัว
ถ้าไม่ใช่ ([{ }])
ก็ต่อเพิ่มให้กับสตริงผลลัพธ์

แบบฝึกหัด: [] กับ ()

Input

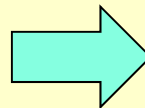
[x + (y - z)]



Output

(x + [y - z])

Programming



Programming

สร้างสตริงใหม่ที่
แทนวงเล็บ () ด้วย [] และ
แทนวงเล็บ [] ด้วย ()

แบบที่ 3: for each_element in a_list

```
x = [0,1,1,2,3,5,8,13,21]
for e in x:
    print(e)
```

หยิบข้อมูลแต่ละตัวจากลิสต์ **x** ใส่ **e**
รอบละหนึ่งตัวจากซ้ายไปขวา

	output
0	
1	
1	
2	
3	
5	
8	
13	
21	

ตัวอย่าง: หาค่าเฉลี่ย

```
x = input().split()
s = 0
for e in x:
    s += float(e)
avg = s/len(x)
print("Average =", avg)
```

Input

10 20 30 20 10

`input().split()`

`["10", "20", "30", "20", "10"]`

Output

Average = 18.0

หยิบมาทีละตัว แปลงเป็น
float แล้วหาค่าเฉลี่ย

Tips: ใช้ได้กับสตริงและลิสต์

หยิบทีละตัวจากซ้ายไปขวา

```
for e in x :  
    ...
```

หยิบทีละตัวจากขวาไปซ้าย

```
for e in x[::-1] :  
    ...
```

หยิบทีละตัวจากซ้ายไปขวา ไม่เอาตัวสุดท้าย

```
for e in x[:-1] :  
    ...
```

หยิบเฉพาะตัว index คี่จากซ้ายไปขวา

```
for e in x[1::2] :  
    ...
```

แบบฝึกหัด: นับจำนวน the และ The

The word "the" is one of the most common words in English.

" () , . ' ↓ เปลี่ยนเครื่องหมายวรรค
ตอนด้วยช่องว่าง

The word the is one of the most common words in English

↓ `split()`

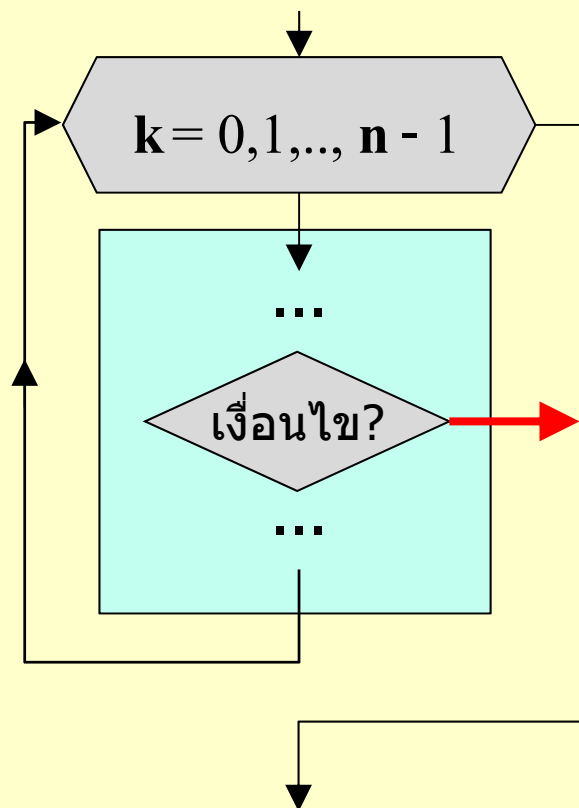
["The", "word", "the", "is", "one", "of", "the",
"most", "common", "words", "in", "English"]

↓ นับ **the** กับ **The**

3

break : คำสั่งเพื่อให้ออกจากวงวน for

```
for k in range(n):  
    ...  
    if เงื่อนไข :  
        break  
    ...
```

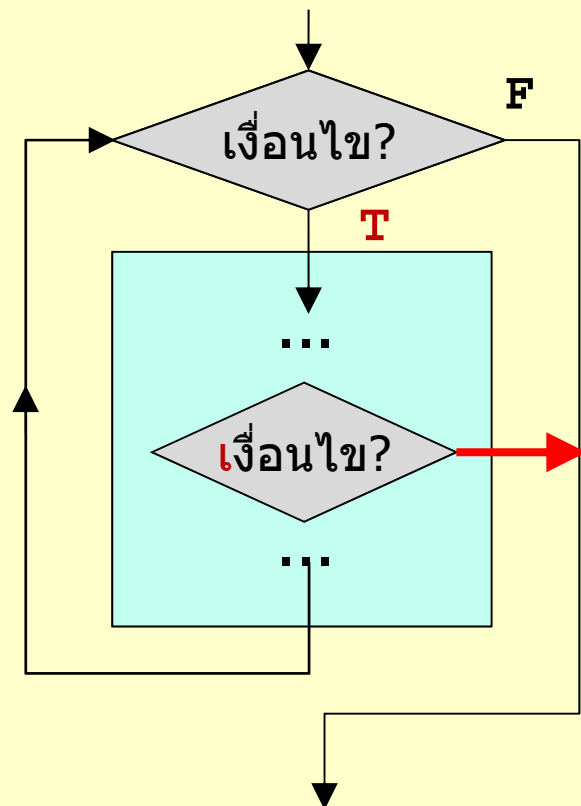


```
n = int(input())  
for k in range(2, n+1):  
    if n % k == 0:  
        break  
if k == n:  
    print("Prime")  
else:  
    print("Composite")
```

รับจำนวนเต็ม n
หา k ที่หาร n ลงตัว $k = 2, 3, \dots, n$
เมื่อออกนอกวงวน
ถ้า k เท่ากับ n → เป็นจำนวนเฉพาะ
ไม่เช่นนั้น → เป็นจำนวนประกอบ

break : คำสั่งเพื่อให้ออกจากวงวน while

```
while เงื่อนไข:  
    ...  
    if เงื่อนไข :  
        break  
    ...
```



```
t = input()  
s = 0; n = 0  
while t != "q":  
    s += float(t)  
    n += 1  
    t = input()  
print( "Average =", s/n )
```

10.0
10.0
11.0
12.0
q

```
s = 0; n = 0  
while True:  
    t = input()  
    if t == "q":  
        break  
    s += float(t)  
    n += 1  
print( "Average =", s/n )
```

10.0
10.0
11.0
12.0
q

while True คือวงวนที่หมุนไปเรื่อย ๆ

แบบฝึกหัด: เกมทายตัวเลข

```
Guess my number (0 to 99)
You have seven tries
50
Higher
75
Lower
57
Higher
68
Lower
62
Lower
60
Lower
59
You win
```

```
Guess my number (0 to 99)
You have seven tries
1
Higher
2
Higher
3
Higher
4
Higher
5
Higher
6
Higher
7
Higher
You lose, the number is 9
```

แบบฝึกหัด: เกมทายตัวเลข

```
import random
```

```
print("Guess my number (0 to 99)")
```

```
print("You have seven tries")
```

```
n = random.randint(0, 99)
```

สุ่มเลขระหว่าง 0 ถึง 99

```
for k in range(7):
```

```
    m = int(input())
```

ให้ลองทายอย่างมาก 7 ครั้ง