

TQC+ 程式語言 Python 第 1 類：基本程式設計

TQC+ 程式語言 Python 101 整數格式化輸出

請撰寫一程式，輸入四個整數，然後將這四個整數以欄寬為 5、欄與欄間隔一個空白字元，再以每列印兩個的方式，先列印向右靠齊，再列印向左靠齊，左右皆以直線 | (Vertical bar) 作為邊界。

```
a=eval(input(""))
b=eval(input(""))
c=eval(input(""))
d=eval(input(""))
```

範例輸入

```
85
4
299
478
```

```
print("|%5d %5d|"%(a,b))
print("|%5d %5d|"%(c,d))
print("|%-5d %-5d|"%(a,b))
print("|%-5d %-5d|"%(c,d))
```

範例輸出

```
| 85 4|
| 299 478|
|85 4|
|299 478|
```

TQC+ 程式語言 Python 102 浮點數格式化輸出

請撰寫一程式，輸入四個分別含有小數 1 到 4 位的浮點數，然後將這四個浮點數以欄寬為 7、欄與欄間隔一個空白字元、每列印兩個的方式，先列印向右靠齊，再列印向左靠齊，左右皆以直線 | (Vertical bar) 作為邊界。提示：輸出浮點數到小數點後第二位。

```
a = float(input(""))
b = float(input(""))
c = float(input(""))
d = float(input(""))
```

範例輸入

```
23.12
395.3
100.4617
564.329
```

```
print("|%7.2f %7.2f|"%(a,b))
print("|%7.2f %7.2f|"%(c,d))
print("|%-7.2f %-7.2f|"%(a,b))
print("|%-7.2f %-7.2f|"%(c,d))
```

範例輸出

```
| 23.12 395.30|
| 100.46 564.33|
|23.12 395.30|
|100.46 564.33|
```

TQC+ 程式語言 Python 103 字串格式化輸出

請撰寫一程式，輸入四個單字，然後將這四個單字以欄寬為 10、欄與欄間隔一個空白字元、每列印兩個的方式，先列印向右靠齊，再列印向左靠齊，左右皆以直線 | (Vertical bar) 作為邊界。

```
a=str(input(""))
b=str(input(""))
c=str(input(""))
d=str(input(""))
```

範例輸入

```
I
enjoy
learning
Python
```

```
print("|%10s %10s|"%(a,b))
print("|%10s %10s|"%(c,d))
print("|%-10s %-10s|"%(a,b))
print("|%-10s %-10s|"%(c,d))
```

範例輸出

```
|      I      enjoy|
| learning    Python|
|I          enjoy  |
|learning    Python  |
```

TQC+ 程式語言 Python 104 圓形面積計算

請撰寫一程式，輸入一圓的半徑，並加以計算此圓之面積和周長，最後請印出此圓的半徑 (Radius)、周長 (Perimeter) 和面積 (Area)。

提示 1：需 import math 模組，並使用 math.pi。

提示 2：輸出浮點數到小數點後第二位。

```
import math
Radius = eval(input(""))
Perimeter = 2 * math.pi * Radius
Area = math.pi * Radius * Radius
print("Radius = %.2f"%(Radius))
print("Perimeter = %.2f"%(Perimeter))
print("Area = %.2f"%(Area))
```

範例輸入1

10

範例輸入2

2.5

範例輸出1

Radius = 10.00
Perimeter = 62.83
Area = 314.16

範例輸出2

Radius = 2.50
Perimeter = 15.71
Area = 19.63

#圓周 = $2 * \text{math.pi} * \text{半徑}$

#圓面積 = $\text{math.pi} * \text{半徑} * \text{半徑}$

TQC+ 程式語言 Python 105 矩形面積計算

請撰寫一程式，輸入兩個正數，代表一矩形之寬和高，計算並輸出此矩形之高 (Height)、寬 (Width)、周長 (Perimeter) 及面積 (Area)。

提示：輸出浮點數到小數點後第二位。

```
Height = eval(input(""))
Width = eval(input(""))
Perimeter = 2 * (Height + Width)
Area = Height * Width
print("Height = %.2f"%(Height))
print("Width = %.2f"%(Width))
print("Perimeter = %.2f"%(Perimeter))
print("Area = %.2f"%(Area))
```

範例輸入

23.5
19

範例輸出

Height = 23.50
Width = 19.00
Perimeter = 85.00
Area = 446.50

TQC+ 程式語言 Python 106 公里英哩換算

假設一賽跑選手在 x 分 y 秒的時間跑完 z 公里，請撰寫一程式，輸入 x、y、z 數值，最後顯示此選手每小時的平均英哩速度 (1 英哩等於 1.6 公里)。

提示：輸出浮點數到小數點後第一位。

```
x = eval(input(""))
y = eval(input(""))
z = eval(input(""))
Speed = (z/1.6)/((60*x+y)/(60*60))
print("Speed = %.1f" %Speed)
```

範例輸入

10
25
3

範例輸出

Speed = 10.8

#速度 = 距離/時間

#速度 = $(\text{km}/1.6)/((60 * \text{min} + \text{sec})/(60 * 60))$

TQC+ 程式語言 Python 107 數值計算

請撰寫一程式，讓使用者輸入五個數字，計算並輸出這五個數字之數值、總和及平均數。

#提示：總和與平均數皆輸出到小數點後第1位。

```
m1 = eval(input(""))
m2 = eval(input(""))
m3 = eval(input(""))
m4 = eval(input(""))
m5 = eval(input(""))
Sum = m1 + m2 + m3 + m4 + m5
Average = Sum / 5
print(m1,m2,m3,m4,m5)
print("Sum = %.1f"%Sum)
print("Average = %.1f"%Average)
```

範例輸入1

```
20
40
60
80
100
```

範例輸入2

```
88.7
12
56
132.55
3
```

範例輸出1

```
20 40 60 80 100
Sum = 300.0
Average = 60.0
```

範例輸出2

```
88.7 12 56 132.55 3
Sum = 292.2
Average = 58.5
```

TQC+ 程式語言 Python 108 座標距離計算

請撰寫一程式，讓使用者輸入四個數字 x_1 、 y_1 、 x_2 、 y_2 ，分別代表兩個點的座標 (x_1, y_1) 、 (x_2, y_2) 。

計算並輸出這兩點的座標與其歐式距離。

提示1：歐式距離 = $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$
提示2：兩座標的歐式距離，輸出到小數點後第4位

```
x1 = eval(input(""))
y1 = eval(input(""))
x2 = eval(input(""))
y2 = eval(input(""))
d = ((x1 - x2)**2 + (y1 - y2)**2)**0.5
print("( { } , { } )".format(x1,y1))
print("( { } , { } )".format(x2,y2))
print("Distance = {:.4f}".format(d))
```

範例輸入

```
2
1
5.5
8
```

範例輸出

```
( 2 , 1 )
( 5.5 , 8 )
Distance = 7.8262
```

TQC+ 程式語言 Python 109 正五邊形面積計算

請撰寫一程式，讓使用者輸入一個正數 s ，代表正五邊形之邊長，計算並輸出此正五邊形之面積 (Area)。

#method1

```
import math
s = eval(input(""))
Area1 = (5 * math.pow(s,2)) / (4 * math.tan(math.pi / 5))
print("Area = {:.4f}".format(Area1))
```

提示1：建議使用import math模組的math.pow及math.tan
提示2：正五邊形面積的公式： $Area = (5 * s^2) / (4 * \tan(\pi/5))$
提示3：輸出浮點數到小數點後第四位。

範例輸入

```
5
```

範例輸出

```
Area = 43.0119
```

#method2

```
import math
s = eval(input(""))
Area = (5 * s**2) / (4 * math.tan(math.pi / 5))
print("Area = {:.4f}".format(Area))
```

TQC+ 程式語言 Python 110 正 n 邊形面積計算

請撰寫一程式，讓使用者輸入兩個正數 n 、 s ，代表正 n 邊形之邊長為 s ，計算並輸出此正 n 邊形之面積 (Area)。

提示1: 建議使用import math模組的math.pow及math.tan

提示2: 正 n 邊形面積的公式如下: $Area = (n * s^2) / (4 * \tan(\pi/n))$

提示3: 輸出浮點數到小數點後第四位

```
import math
n = eval(input(""))
s = eval(input(""))
Area = (n * math.pow(s,2)) / (4 * math.tan(math.pi / n))
print("Area = {:.4f}".format(Area))
```

範例輸入

8
6

範例輸出

Area = 173.8234

TQC+ 程式語言 Python 第 2 類：選擇敘述

TQC+ 程式語言 Python 201 偶數判斷

請使用選擇敘述撰寫一程式，讓使用者輸入一個正整數，然後判斷它是否為偶數 (even)。

```
n = eval(input(""))
if n % 2 == 0:
    print("%d is an even number." %n)
else:
    print("%d is not an even number." %n)
```

範例輸入1

56

範例輸出1

56 is an even number.

範例輸入2

21

範例輸出2

21 is not an even number.

TQC+ 程式語言 Python 202 倍數判斷

請使用選擇敘述撰寫一程式，讓使用者輸入一個正整數，然後判斷它是 3 或 5 的倍數，顯示【x is a multiple of 3.】或【x is a multiple of 5.】；若此數值同時為 3 與 5 的倍數，顯示【x is a multiple of 3 and 5.】；如此數值皆不屬於 3 或 5 的倍數，顯示【x is not a multiple of 3 or 5.】，將使用者輸入的數值代入 x。

#method1

```
n = eval(input(""))
if n % 3 == 0:
    if n % 5 == 0:
        print("%d is a multiple of 3 and 5." %n)
    else:
        print("%d is a multiple of 3." %n)
elif n % 5 == 0:
    print("%d is a multiple of 5." %n)
else:
    print("%d is not a multiple of 3 or 5." %n)
```

範例輸入1

55

範例輸出1

55 is a multiple of 5.

範例輸入2

36

範例輸出2

36 is a multiple of 3.

範例輸入3

92

範例輸出3

92 is not a multiple of 3 or 5.

範例輸入4

15

範例輸出4

15 is a multiple of 3 and 5.

#method2

```
n = eval(input())
if n % 3 == 0 and n % 5 == 0:
    print(n,"is a multiple of 3 and 5.")
elif n % 3 != 0 and n % 5 != 0:
    print(n,"is not a multiple of 3 or 5.")
elif n % 3 != 0 and n % 5 == 0:
    print(n,"is a multiple of 5.")
elif n % 3 == 0 and n % 5 != 0:
    print(n,"is a multiple of 3.")
```

TQC+ 程式語言 Python 203 閏年判斷

請使用選擇敘述撰寫一程式，讓使用者輸入一個西元年份，然後判斷它是否為閏年（leap year）或平年。其判斷規則為：每四年一閏，每百年**不閏**，但每四百年也一閏。

```
y = eval(input(""))
if y % 4 == 0 and y % 100 != 0 or y % 400 == 0:
    print("%d is a leap year." % y)
else:
    print("%d is not a leap year." % y)
```

範例輸入1

1992

範例輸出1

1992 is a leap year.

範例輸入2

2010

範例輸出2

2010 is not a leap year.

TQC+ 程式語言 Python 204 算術運算

請使用選擇敘述撰寫一程式，讓使用者輸入**兩個整數 a、b**，然後再輸入一算術運算子（+、-、*、/、//、%），輸出經過運算後的結果。

```
a = eval(input(""))
b = eval(input(""))
c = str(input(""))
if c == "+":
    print(a + b)
elif c == "-":
    print(a - b)
elif c == "*":
    print(a * b)
elif c == "/":
    print(a / b)
elif c == "//":
    print(a // b)
elif c == "%":
    print(a % b)
```

範例輸入

30
20
*

範例輸出1

600

TQC+ 程式語言 Python 205 字元判斷

請使用選擇敘述撰寫一程式，讓使用者輸入一個字元，判斷它是包括大、小寫的英文字母 (alphabet)、數字 (number)、或者其它字元 (symbol)。例如：a 為英文字母、9 為數字、\$ 為其它字元。

#method1

```
c = input()
if c >= "A" and c <= "Z":
    print("%s is an alphabet." %c)
elif c >= "a" and c <= "z":
    print("%s is an alphabet." %c)
elif c >= "0" and c <= "9":
    print("%s is a number." %c)
else:
    print("%s is a symbol." %c)
```

範例輸入1

P

範例輸出1

P is an alphabet.

範例輸入2

@

範例輸出2

@ is a symbol.

範例輸入3

7

範例輸出3

7 is a number.

#method2

```
c = input()
if c.isalpha():
    print(c,"is an alphabet.")
elif c.isdigit():
    print(c,"is a number.")
else:
    print(c,"is a symbol.")
```

TQC+ 程式語言 Python 206 等級判斷

請使用選擇敘述撰寫一程式，根據使用者輸入的分數顯示對應的等級。標準如下表所示：

分數	等級
80 ~ 100	A
70 ~ 79	B
60 ~ 69	C
<= 59	F

```
s = eval(input())
if 80 <= s:
    print("A")
elif 70 <= s:
    print("B")
elif 60 <= s:
    print("C")
else:
    print("F")
```

範例輸入

79

範例輸出

B

TQC+ 程式語言 Python 207 折扣方案

請使用選擇敘述撰寫一程式，要求使用者輸入購物金額，購物金額需大於 8,000（含）以上，並顯示折扣優惠後的實付金額。購物金額折扣方案如下表所示：

```
m = eval(input())
```

```
if m >= 38000:
    print(m * 0.7)
elif m >= 28000:
    print(m * 0.8)
elif m >= 18000:
    print(m * 0.9)
elif m >= 8000:
    print(m * 0.95)
else:
    print(m)
```

金 額	折 扣
8,000 (含) 以上	9.5折
18,000 (含) 以上	9折
28,000 (含) 以上	8折
38,000 (含) 以上	7折

範例輸入

12000

範例輸出

11400.0

TQC+ 程式語言 Python 208 十進位換算

請使用選擇敘述撰寫一程式，讓使用者輸入一個十進位整數 num($0 \leq \text{num} \leq 15$)，將 num 轉換成十六進位值。

#提示：轉換規則 = 十進位 0~9 的十六進位值為其本身，十進位 10~15 的十六進位值為 A~F。

```
num = eval(input())
```

```
if num <= 9:
    print(num)
elif num == 10:
    print("A")
elif num == 11:
    print("B")
elif num == 12:
    print("C")
elif num == 13:
    print("D")
elif num == 14:
    print("E")
elif num == 15:
    print("F")
```

範例輸入1

13

範例輸出1

D

範例輸入2

8

範例輸出2

8

TQC+ 程式語言 Python 209 距離判斷

請使用選擇敘述撰寫一程式，讓使用者輸入一個點的平面座標 x 和 y 值，判斷此點是否與點(5, 6)的距離小於或等於 15，如距離小於或等於 15 顯示【Inside】，反之顯示【Outside】。

```
import math
a = eval(input(""))
b = eval(input(""))
d = math.sqrt((a - 5)**2 + (b - 6)**2)
if d <= 15:
    print("Inside")
else:
    print("Outside")
```

提示：計算平面上兩點距離的公式： $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$

範例輸入1

7
20

範例輸入2

30
35

範例輸出1

Inside

範例輸出2

Outside

TQC+ 程式語言 Python 210 三角形判斷

請使用選擇敘述撰寫一程式，讓使用者輸入三個邊長，檢查這三個邊長是否可以組成一個三角形。若可以，則輸出該三角形之周長；否則顯示【Invalid】。

提示：檢查方法 = 任意兩個邊長之總和大於第三邊長。

```
a = eval(input(""))
b = eval(input(""))
c = eval(input(""))
if a + b > c and b + c > a and a + c > b:
    print(a + b + c)
else:
    print("Invalid")
```

範例輸入1

5
6
13

範例輸入2

1
1
1

範例輸出1

Invalid

範例輸出2

3

TQC+ 程式語言 Python 第 3 類：迴圈敘述

TQC+ 程式語言 Python 301 迴圈整數連加

請使用迴圈敘述撰寫一程式，讓使用者輸入兩個正整數 a 、 b ($a < b$)，利用迴圈計算從 a 開始連加到 b 的總和。例如：輸入 $a=1$ 、 $b=100$ ，則輸出結果為 5050 ($1 + 2 + \dots + 100 = 5050$)。

```
a = eval(input(""))
```

```
b = eval(input(""))
```

```
idx = 0
```

```
for i in range(a,b + 1):
```

```
    idx += i
```

```
print(idx)
```

範例輸入

66

666

範例輸出

219966

TQC+ 程式語言 Python 302 迴圈偶數連加

請使用迴圈敘述撰寫一程式，讓使用者輸入兩個正整數 a 、 b ($a < b$)，利用迴圈計算從 a 開始的偶數連加到 b 的總和。例如：輸入 $a=1$ 、 $b=100$ ，則輸出結果為 2550 ($2 + 4 + \dots + 100 = 2550$)。

```
a = eval(input(""))
```

```
b = eval(input(""))
```

```
idx = 0
```

```
for i in range(a,b + 1):
```

```
    if i % 2 == 0:
```

```
        idx += i
```

```
print(idx)
```

範例輸入

14

1144

範例輸出

327714

TQC+ 程式語言 Python 303 迴圈數值相乘

請使用迴圈敘述撰寫一程式，讓使用者輸入一個正整數 (< 100)，然後以三角形的方式依序輸出此數的相乘結果。 [提示：輸出欄寬為 4，且需靠右對齊。]

```
a = eval(input(""))
```

```
for i in range(1,a+1):
```

```
    for j in range(1,i+1):
```

```
        print("%4d"%(j*i),end="")
```

```
    print("")
```

範例輸入1

5

範例輸出1

```
1
2  4
3  6  9
4  8 12 16
5 10 15 20 25
```

TQC+ 程式語言 Python 304 迴圈倍數總和

請使用迴圈敘述撰寫一程式，讓使用者輸入一個正整數 a ，利用迴圈計算從 1 到 a 之間，所有 5 之倍數數字總和。

```
a = eval(input(""))
```

```
idx = 0
```

```
for i in range(1,a + 1):
```

```
    if i % 5 == 0:
```

```
        idx += i
```

```
print(idx)
```

範例輸入

21

範例輸出

50

TQC+ 程式語言 Python 305 數字反轉

請撰寫一程式，讓使用者輸入一個正整數，將此數值以反轉的順序輸出。

```
a = input()
print(a[::-1])
```

TQC+ 程式語言 Python 306 迴圈階乘計算

請使用迴圈敘述撰寫一程式，讓使用者輸入一個正整數 n ，利用迴圈計算並輸出 $n!$ 的值。

```
a = eval(input(""))
idx = 1
for i in range(2,a+1):
    idx *= i
print(idx)
```

範例輸入

15

範例輸出

1307674368000

TQC+ 程式語言 Python 307 乘法表

- (1) 請使用迴圈敘述撰寫一程式，要求使用者輸入一個正整數 n ($n < 10$)，顯示 $n * n$ 乘法表。
- (2) 每項運算式需進行格式化排列整齊，每個運算子及運算元輸出的欄寬為 2，而每項乘積輸出的欄寬為 4，皆靠左對齊不跳行。

```
a = eval(input(""))

for i in range(1,a+1):
    for j in range(1,a+1):
        print("%-2d* %-2d= %-4d"%(j,i,j*i),end="")
    print("")
```

範例輸入1

3

範例輸出1

1 * 1 = 1	2 * 1 = 2	3 * 1 = 3
1 * 2 = 2	2 * 2 = 4	3 * 2 = 6
1 * 3 = 3	2 * 3 = 6	3 * 3 = 9

TQC+ 程式語言 Python 308 迴圈位數加總

請使用迴圈敘述撰寫一程式，要求使用者輸入一個數字，此數字代表後面測試資料的數量。每一筆測試資料是一個正整數（由使用者輸入），將此正整數的每位數全部加總起來。

```
a = int(input())

for i in range(a):
    total = 0
    num = input("")
    for i in range(len(num)):
        total += int(num[i])
    print("Sum of all digits of {} is {}".format(num,total))
```

輸入與輸出會交雜如下，輸出的部份以粗體字表示 1

```
1
98765
Sum of all digits of 98765 is 35
```

輸入與輸出會交雜如下，輸出的部份以粗體字表示 2

```
3
32412
Sum of all digits of 32412 is 12
0
Sum of all digits of 0 is 0
769
Sum of all digits of 769 is 22
```

TQC+ 程式語言 Python 309 存款總額

請使用迴圈敘述撰寫一程式，提示使用者輸入金額（如 10,000）、年收益率（如 5.75），以及經過的月份數（如 5），接著顯示每個月的存款總額。 #提示：四捨五入，輸出浮點數到小數點後第二位。

舉例：

假設您存款\$10,000，年收益為 5.75%。

過了一個 month，存款會是： $10000 + 10000 * 5.75 / 1200 = 10047.92$

過了兩個 month，存款會是： $10047.92 + 10047.92 * 5.75 / 1200 = 10096.06$

過了三個月，存款將是： $10096.06 + 10096.06 * 5.75 / 1200 = 10144.44$

```
t = eval(input(""))
y = eval(input(""))
m = eval(input(""))
print("%s\t\t%s" % ("Month", "Amount"))
for i in range(m):
    t = t + (t * y / 1200)
    print("%3d\t\t%.2f" % (i+1, t))
```

範例輸入

```
50000
1.3
5
```

範例輸出

Month	Amount
1	50054.17
2	50108.39
3	50162.68
4	50217.02
5	50271.42

TQC+ 程式語言 Python 310 迴圈公式計算

請使用迴圈敘述撰寫一程式，讓使用者輸入正整數 n ($1 < n$)，計算以下公式的總和並顯示結果：

$$\frac{1}{1 + \sqrt{2}} + \frac{1}{\sqrt{2} + \sqrt{3}} + \frac{1}{\sqrt{3} + \sqrt{4}} + \dots + \frac{1}{\sqrt{n-1} + \sqrt{n}}$$

提示：輸出結果至小數點後四位。

```
import math
n = eval(input(""))
idx = 0
for i in range(2, n+1):
    idx += 1 / (math.sqrt(i-1) + math.sqrt(i))
print("%.4f" % idx)
```

範例輸入

```
8
```

範例輸出

```
1.8284
```

TQC+ 程式語言 Python 第 4 類：進階控制流程

TQC+ 程式語言 Python 401 最小值

請撰寫一程式，由使用者輸入十個數字，然後找出其最小值，最後輸出最小值。

```
l=[]
for i in range(10):
    l.append(eval(input()))
print(min(l))
```

TQC+ 程式語言 Python 402 不定數迴圈-最小值

請撰寫一程式，讓使用者輸入數字，輸入的動作直到輸入值為 9999 才結束，然後找出其最小值，並輸出最小值。

```
l=[]
while True:
    n = eval(input(""))
    if n == 9999:
        break
    l.append(n)
print(min(l))
```

TQC+ 程式語言 Python 403 倍數總和計算

請撰寫一程式，讓使用者輸入兩個正整數 a、b ($a \leq b$)，輸出從 a 到 b (包含 a 和 b) 之間 4 或 9 的倍數 (一行輸出十個數字、欄寬為 4、靠左對齊) 以及倍數之個數、總和。

```
a = eval(input())
b = eval(input())

idx = 0
l = []
for i in range(a,b+1):
    if i % 4 == 0 or i % 9 == 0:
        idx += 1
        l.append(i)
    print("%-4d"%i,end="")
    if idx % 10 == 0:
        print()
```

範例輸入1

```
5
55
```

範例輸出1

```
8  9  12 16 18 20 24 27 28 32
36 40 44 45 48 52 54
17
513
```

範例輸入2

```
4
9
```

範例輸出2

```
4  8  9
3
21
```

TQC+ 程式語言 Python 404 數字反轉判斷

請撰寫一程式，讓使用者輸入一個正整數，將此正整數以反轉的順序輸出，並判斷如輸入 0，則輸出為 0。

```
n = input("")
print(n[::-1])
```

TQC+ 程式語言 Python 405 不定數迴圈-分數等級

請撰寫一程式，以不定數迴圈的方式輸入一個正整數（代表分數），之後根據以下分數與 GPA 的對照表，印出其所對應的 GPA。假設此不定數迴圈輸入-9999則會結束此迴圈。標準如下表所示：

while True:

a = eval(input())

if a >= 90:

print("A")

elif a >= 80:

print("B")

elif a >= 70:

print("C")

elif a >= 60:

print("D")

elif a >= 0:

print("E")

elif a == -9999:

break

分 數 GPA

90 ~ 100 A

80 ~ 89 B

70 ~ 79 C

60 ~ 69 D

0 ~ 59 E

輸入與輸出會交雜如下，輸出的部份以粗體字表示

75
C
39
E
100
A
85
B
65
D
-9999

TQC+ 程式語言 Python 406 不定數迴圈-BMI 計算

請撰寫一程式，以不定數迴圈的方式輸入身高與體重，計算出 BMI 之後再根據以下對照表，印出 BMI 及相對應的 BMI 代表意義 (State)。假設此不定數迴圈輸入-9999則會結束此迴圈。標準如下表所示：

while True:

h = eval(input(""))

if h == -9999:

break

w = eval(input(""))

if w == -9999:

break

bmi = w/(h/100)**2

if bmi < 18.5:

s="under weight"

elif bmi >= 18.5 and bmi < 25:

s="normal"

elif bmi >= 25.0 and bmi < 30:

s="over weight"

else:

s="fat"

print("BMI: {:.2f}\nState: {}".format(bmi,s))

提示: $BMI = \text{體重}(kg) / \text{身高}^2(m)$ ，輸出浮點數到小數點後第二位。不需考慮男性或女性標準。

BMI值 代表意義

BMI < 18.5 under weight

18.5 <= BMI < 25 normal

25.0 <= BMI < 30 over weight

30 <= BMI fat

輸入與輸出會交雜如下，輸出的部份以粗體字表示

176
80
BMI: 25.83
State: over weight
170
100
BMI: 34.60
State: fat
-9999

TQC+ 程式語言 Python 407 不定數迴圈-閏年判斷

- (1) 請撰寫一程式，以不定數迴圈的方式讓使用者輸入西元年份，然後判斷它是否為閏年(leap year)或平年。其判斷規則如下：每四年**一閏**，每百年**不閏**，但每四百年也**一閏**。
- (2) 假設此不定數迴圈輸入-9999 則會結束此迴圈。

#method1

```
y = 0
```

```
while y != -9999:
```

```
    y = eval(input(""))
```

```
    if y != -9999:
```

```
        if y % 4 == 0 and y % 100 != 0:
```

```
            print(y,"is a leap year.")
```

```
        elif y % 400 == 0:
```

```
            print(y,"is a leap year.")
```

```
        else:
```

```
            print(y,"is not a leap year.")
```

#method2

```
year = eval(input())
```

```
while year!=-9999:
```

```
    if year%400==0:
```

```
        print(y,"is a leap year.")
```

```
    elif year%100==0:
```

```
        print(y,"is not a leap year.")
```

```
    elif year%4==0:
```

```
        print(y,"is a leap year.")
```

```
    else:
```

```
        print(y,"is not a leap year.")
```

```
    year = eval(input())
```

輸入與輸出會交雜如下，輸出的部份以粗體字表示

2017

2017 is not a leap year.

2000

2000 is a leap year.

2016

2016 is a leap year.

2009

2009 is not a leap year.

2018

2018 is not a leap year.

-9999

TQC+ 程式語言 Python 408 奇偶數個數計算

請撰寫一程式，讓使用者**輸入十個整數**，計算並輸出**偶數**和**奇數**的個數。

```
units = 0
```

```
tens = 0
```

```
for i in range(10):
```

```
    n = int(input(""))
```

```
    if n % 2 == 0:
```

```
        units += 1
```

```
    if n % 2 != 0:
```

```
        tens += 1
```

```
print("Even numbers: %d" %units)
```

```
print("Odd numbers: %d" %tens)
```

範例輸入

```
69
48
19
91
83
22
18
37
82
40
```

範例輸出

```
Even numbers: 5
Odd numbers: 5
```

TQC+ 程式語言 Python 409 得票數計算

某次選舉有兩位候選人，分別是 No.1: Nami、No.2: Chopper。請撰寫一程式，輸入五張選票，輸入值如為 1 即表示針對 1 號候選人投票；輸入值如為 2 即表示針對 2 號候選人投票，如輸入其他值則視為廢票。每次投完後需印出目前每位候選人的得票數，最後印出最高票者為當選人；如最終計算有相同的最高票數者或無法選出最高票者，顯示【=> No one won the election.】。

```
Nami_vote = 0
```

```
Chopper_vote = 0
```

```
Null_vote = 0
```

```
for i in range(5):
```

```
    vote = int(input(""))
```

```
    if vote == 1 :
```

```
        Nami_vote += 1
```

```
    elif vote == 2 :
```

```
        Chopper_vote += 1
```

```
    else:
```

```
        Null_vote += 1
```

```
    print("Total votes of No.1: Nami = %d"%Nami_vote)
```

```
    print("Total votes of No.2: Chopper = %d"%Chopper_vote)
```

```
    print("Total null votes = %d"%Null_vote)
```

```
if Nami_vote > Chopper_vote:
```

```
    print("=> No.1 Nami won the election.")
```

```
elif Nami_vote < Chopper_vote:
```

```
    print("=> No.2 Chopper won the election.")
```

```
else:
```

```
    print("=> No one won the election.")
```

輸入與輸出會交雜如下，輸出的部份以粗體字表示

```
2
Total votes of No.1: Nami = 0
Total votes of No.2: Chopper = 1
Total null votes = 0
1
Total votes of No.1: Nami = 1
Total votes of No.2: Chopper = 1
Total null votes = 0
8
Total votes of No.1: Nami = 1
Total votes of No.2: Chopper = 1
Total null votes = 1
2
Total votes of No.1: Nami = 1
Total votes of No.2: Chopper = 2
Total null votes = 1
2
Total votes of No.1: Nami = 1
Total votes of No.2: Chopper = 3
Total null votes = 1
=> No.2 Chopper won the election.
```

TQC+ 程式語言 Python 410 繪製等腰三角形

請撰寫一程式，依照使用者輸入的 n，畫出對應的等腰三角形。

```
n = int(input(""))
```

```
for i in range(1,n+1):
```

```
    for sp in range(n-i):
```

```
        print(" ",end="")
```

```
    for star in range(i*2-1):
```

```
        print("*",end="")
```

```
    print()
```

範例輸入

7

範例輸出

```
  *
 ***
*****
*****
*****
*****
*****
```


TQC+ 程式語言 Python 第 5 類：函式(Function)

TQC+ 程式語言 Python 501 訊息顯示

請撰寫一程式，呼叫函式 `compute()`，該函式功能為讓使用者輸入系別 (Department)、學號 (Student ID) 和姓名 (Name) 並顯示這些訊息。

```
def compute():
```

```
    Department = input()
```

```
    ID = input()
```

```
    Name = input()
```

```
    print("Department:", Department)
```

```
    print("Student ID:", ID)
```

```
    print("Name:", Name)
```

```
compute()
```

範例輸入

```
Information Management
123456789
Tina Chen
```

範例輸出

```
Department: Information Management
Student ID: 123456789
Name: Tina Chen
```

TQC+ 程式語言 Python 502 乘積

請撰寫一程式，將使用者輸入的兩個整數作為參數傳遞給一個名為 `compute(x, y)` 的函式，此函式將回傳 `x` 和 `y` 的乘積。

```
def compute(x,y):
```

```
    z = x*y
```

```
    return z
```

```
x = eval(input(""))
```

```
y = eval(input(""))
```

```
print(compute(x,y))
```

範例輸入

```
56
11
```

範例輸出

```
616
```

TQC+ 程式語言 Python 503 連加計算

請撰寫一程式，讓使用者輸入兩個整數，接著呼叫函式 `compute()`，此函式接收兩個參數 `a`、`b`，並回傳從 `a` 連加到 `b` 的和。

```
def compute(a,b):
```

```
    summation = 0
```

```
    for i in range(a,b+1):
```

```
        summation += i
```

```
    return summation
```

```
a = eval(input(""))
```

```
b = eval(input(""))
```

```
print(compute(a,b))
```

範例輸入

```
33
66
```

範例輸出

```
1683
```

TQC+ 程式語言 Python 504 次方計算

請撰寫一程式，讓使用者輸入兩個整數，接著呼叫函式compute()，此函式接收兩個參數a、b，並回傳 a^b 的值。

```
def compute(a,b):  
    z = a**b  
    return z  
  
a = eval(input(""))  
b = eval(input(""))  
print(compute(a,b))
```

範例輸入

```
14  
3
```

範例輸出

```
2744
```

TQC+ 程式語言 Python 505 依參數格式化輸出

請撰寫一程式，將使用者輸入的三個參數，變數名稱分別為a（代表字元 character）、x（代表個數）、y（代表列數），作為參數傳遞給一個名為compute()的函式，該函式功能為：一列印出x個a字元，總共印出y列。

```
def compute(a,x,y):  
    for i in range(y):  
        for i in range(x):  
            print(a,end=" ")  
        print()  
  
a = input()  
x = eval(input(""))  
y = eval(input(""))  
compute(a,x,y)
```

範例輸入

```
e  
5  
4
```

範例輸出

```
e e e e e  
e e e e e  
e e e e e  
e e e e e
```

TQC+ 程式語言 Python 506 一元二次方程式

請撰寫一程式，將使用者輸入的三個整數（代表一元二次方程式 $ax^2 + bx + c = 0$ 的三個係數a、b、c）作為參數傳遞給一個名為compute()的函式，該函式回傳方程式的解，如無解則輸出【Your equation has no root.】

```
from math import sqrt
```

提示：輸出有順序性

```
def compute(a,b,c):  
    if b ** 2 - 4 * a * c >= 0:  
        ans1 = (-b + sqrt(b ** 2 - 4 * a * c)) / (2 * a)  
        ans2 = (-b - sqrt(b ** 2 - 4 * a * c)) / (2 * a) #記得是減-sqrt  
        return ans1,ans2  
    else:  
        return 0  
  
a = eval(input(""))  
b = eval(input(""))  
c = eval(input(""))  
if compute(a,b,c) != 0 :  
    ans1,ans2 = compute(a,b,c)  
    print("{} , {}".format(ans1,ans2)) #記得"{}",空一格"  
else:  
    print("Your equation has no root.")
```

範例輸入1

```
2  
-3  
1
```

範例輸出1

```
1.0, 0.5
```

範例輸入2

```
9  
9  
8
```

範例輸出2

```
Your equation has no root.
```

TQC+ 程式語言 Python 507 質數

請撰寫一程式，讓使用者輸入一個整數 x ，並將 x 傳遞給名為 `compute()` 的函式，此函式將回傳 x 是否為質數 (Prime number) 的布林值，接著再將判斷結果輸出。如輸入值為質數顯示【Prime】，否則顯示【Not Prime】。

```
def compute(x):
    if x <= 1:
        return "Not Prime"
    for i in range(2,x):
        if x % i == 0:
            return "Not Prime"
    else:
        return "Prime"

x = eval(input(""))
print(compute(x))
```

範例輸入1	範例輸入2	範例輸入3	範例輸入4	範例輸入5
3	6	1	0	-5
範例輸出1	範例輸出2	範例輸出3	範例輸出4	範例輸出5
Prime	Not Prime	Not Prime	Not Prime	Not Prime

TQC+ 程式語言 Python 508 最大公因數

請撰寫一程式，讓使用者輸入兩個正整數 x 、 y ，並將 x 與 y 傳遞給名為 `compute()` 的函式，此函式回傳 x 和 y 的最大公因數。

```
def compute(x,y):
    if y == 0:
        return x
    else:
        return compute(y,x % y)

x,y = eval(input())
print(compute(x,y))
```

範例輸入1
12,8
範例輸出1
4
範例輸入2
4,6
範例輸出2
2

TQC+ 程式語言 Python 509 最簡分數

請撰寫一程式，讓使用者輸入二個分數，分別是 x/y 和 m/n (其中 x 、 y 、 m 、 n 皆為正整數)，計算這兩個分數的和為 p/q ，接著將 p 和 q 傳遞給名為 `compute()` 函式，此函式回傳 p 和 q 的最大公因數 (Greatest Common Divisor, GCD)。再將 p 和 q 各除以其最大公因數，最後輸出的結果必須以最簡分數表示。

```
from math import gcd

def compute(p,q):
    return gcd(p,q)

x,y = eval(input())
m,n = eval(input())
p = (x * n + m * y) / compute(x * n + m * y, y * n)
q = (y * n) / compute(x * n + m * y, y * n)
print("%d/%d + %d/%d = %d/%d"%(x,y,m,n,p,q))
```

範例輸入1
1,2 1,6
範例輸出1
1/2 + 1/6 = 2/3
範例輸入2
12,16 18,32
範例輸出2
12/16 + 18/32 = 21/16

TQC+ 程式語言 Python 510 費氏數列

請撰寫一程式，計算費氏數列（Fibonacci numbers），使用者輸入一正整數 num (num>=2)，並將它傳遞給名為 compute() 的函式，此函式將輸出費氏數列前 num 個的數值。

提示：費氏數列的某一項數字是其前兩項的和，而且第 0 項為 0，第一項為 1，表示方式如下：

$$F_0 = 0$$

$$F_1 = 1$$

$$F_n = F_{n-1} + F_{n-2}$$

```
def compute(n):  
    if n < 2:  
        return n  
    else:  
        return compute(n-1) + compute(n-2)  
  
num = int(input())  
for i in range(num):  
    print(compute(i),end=" ")
```

範例輸入1

10

範例輸出1

0 1 1 2 3 5 8 13 21 34

範例輸入2

20

範例輸出2

0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181

TQC+ 程式語言 Python 第 6 類：串列(List)的運作(一維、二維以及多維)

TQC+ 程式語言 Python 601 偶數索引值加總

請撰寫一程式，利用一維串列存放使用者輸入的 12 個正整數（範圍 1~99）。顯示這些數字，接著將串列索引為偶數的數字相加並輸出結果。

提示：輸出每一個數字欄寬設定為3，每3個一列，靠右對齊。

```
L = []
```

範例輸入

```
56
45
43
22
3
1
39
20
93
18
44
83
```

```
for i in range(12):
```

```
    L.append(int(input()))
```

```
s = 0
```

```
for j in range(12):
```

```
    if j % 2 == 0:
```

```
        s += L[j]
```

```
    print("{:3d}".format(L[j]),end="")
```

```
    if j % 3 == 2:
```

```
        print()
```

```
print(s)
```

範例輸出

```
56 45 43
22  3  1
39 20 93
18 44 83
278
```

TQC+ 程式語言 602 撲克牌總和

請撰寫一程式，讓使用者輸入 52 張牌中的 5 張，計算並輸出其總和。

```
L = []
```

提示：J、Q、K以及A分別代表11、12、13以及1。

```
for i in range(5):
```

```
    x = input()
```

```
    if x == "A":
```

```
        x = 1
```

```
    if x == "J":
```

```
        x = 11
```

```
    if x == "Q":
```

```
        x = 12
```

```
    if x == "K":
```

```
        x = 13
```

```
    L.append(int(x))
```

```
print(sum(L))
```

範例輸入

```
5
10
K
3
A
```

範例輸出

```
32
```

TQC+ 程式語言 Python 603 數字排序

請撰寫一程式，要求使用者輸入十個數字並存放在串列中。接著由大到小的順序顯示最大的 3 個數字。

範例輸入1

```
40
32
12
29
20
19
38
48
57
44
```

```
l = []
```

```
for i in range(10):
```

```
    l.append(eval(input()))
```

```
l.sort()
```

```
print(l[9],l[8],l[7])
```

範例輸出1

```
57 48 44
```

TQC+ 程式語言 Python 604 眾數

請撰寫一程式，讓使用者輸入十個整數作為樣本數，輸出眾數（樣本中出現最多次的數字）及其出現的次數。

```
l = []
for i in range(10):
    l.append(eval(input()))

s = 0
for i in range(10):
    if l.count(l[i]) > s:
        s = l.count(l[i])
        m = l[i]

print(m)
print(s)
```

提示：假設樣本中只有一個眾數。

範例輸入

```
34
18
22
32
18
29
30
38
42
18
```

範例輸出

```
18
3
```

TQC+ 程式語言 Python 605 成績計算

請撰寫一程式，讓使用者輸入十個成績，接下來將十個成績中最小和最大值（最小、最大值不重複）以外的成績作加總及平均，並輸出結果。提示：平均值輸出到小數點後第二位。

```
l = []
for grades in range(10):
    x = eval(input())
    l.append(x)

Sum = sum(l) - max(l) - min(l)
print(Sum)
print("%.2f"%(Sum/(10-2)))
```

範例輸入

```
89
78
67
80
75
98
77
89
76
60
```

範例輸出

```
631
78.88
```

TQC+ 程式語言 Python 606 二維串列行列數

請撰寫一程式，讓使用者輸入兩個正整數 rows、cols，分別表示二維串列 lst 的「第一個維度大小」與「第二個維度大小」。

串列元素[row][col]所儲存的數字，其規則為：row、col 的交點值 = 第二個維度的索引 col - 第一個維度的索引 row。

接著以該串列作為參數呼叫函式 compute()輸出串列。提示：欄寬為 4。

```
def compute(rows,cols):
    for i in range(rows):
        for j in range(cols):
            print("{:4d}".format(j-i),end="")
        print()

rows,cols = int(input()),int(input())
compute(rows,cols)
```

範例輸入

```
5
10
```

範例輸出

```
0  1  2  3  4  5  6  7  8  9
-1 0  1  2  3  4  5  6  7  8
-2 -1 0  1  2  3  4  5  6  7
-3 -2 -1 0  1  2  3  4  5  6
-4 -3 -2 -1 0  1  2  3  4  5
```

TQC+ 程式語言 Python 607 成績計算

請撰寫一程式，讓使用者輸入三位學生各五筆成績，接著再計算並輸出每位學生的總分及平均分數。

提示：平均分數輸出到小數點後第二位。

```
Rank = ["1st", "2nd", "3rd"]
grades = [[0 for i in range(5)] for j in range(3)]

for i in range(3):
    print("The {} student:".format(Rank[i]))
    for j in range(5):
        grades[i][j] = int(input())

for i in range(3):
    total = sum(grades[i])
    print("Student {} \n#Sum {} \n#Average {:.2f}"
          .format(i + 1, total, (total/5)))
```

The 1st student:

78
89
88
70
60

The 2nd student:

90
78
66
68
78

The 3rd student:

69
97
70
89
90

Student 1

#Sum 385

#Average 77.00

Student 2

#Sum 380

#Average 76.00

Student 3

#Sum 415

#Average 83.00

TQC+ 程式語言 Python 608 最大最小值索引

請撰寫一程式，讓使用者建立一個 3*3 的矩陣，其內容為從鍵盤輸入的整數（不重複），接著輸出矩陣最大值與最小值的索引。

```
l = []
for i in range(9):
    l.append(int(input()))

max_n = max(l)
max_n_index = l.index(max_n)
print("Index of the largest number {:d} is: ({:d}, {:d})"
      .format(max_n, (max_n_index // 3), (max_n_index % 3)))

min_n = min(l)
min_n_index = l.index(min_n)
print("Index of the smallest number {:d} is: ({:d}, {:d})"
      .format(min_n, (min_n_index // 3), (min_n_index % 3)))
```

範例輸入

6
4
8
39
12
3
-3
49
33

範例輸出

Index of the largest number 49 is: (2, 1)
Index of the smallest number -3 is: (2, 0)

TQC+ 程式語言 Python 609 矩陣相加

請撰寫一程式，讓使用者建立兩個 2*2 的矩陣，其內容為從鍵盤輸入的整數，接著輸出這兩個矩陣的內容以及它們相加的結果。

```
matrix1 = [[0 for i in range(2)]for j in range(2)]
matrix2 = [[0 for i in range(2)]for j in range(2)]
```

```
print("Enter matrix 1:")
for i in range(2):
    for j in range(2):
        matrix1[i][j] = int(input("[ {}, {}]: ".format(i+1,j+1)))
```

```
print("Enter matrix 2:")
for i in range(2):
    for j in range(2):
        matrix2[i][j] = int(input("[ {}, {}]: ".format(i+1,j+1)))
```

```
print("Matrix 1:")
for i in range(2):
    for j in range(2):
        print(matrix1[i][j],end=" ")
    print()
```

```
print("Matrix 2:")
for i in range(2):
    for j in range(2):
        print(matrix2[i][j],end=" ")
    print()
```

```
print("Sum of 2 matrices:")
for i in range(2):
    for j in range(2):
        print(matrix1[i][j]+matrix2[i][j],end=" ")
    print()
```

Enter matrix 1:

[1, 1]: 3

[1, 2]: 5

[2, 1]: 7

[2, 2]: 5

Enter matrix 2:

[1, 1]: 6

[1, 2]: 9

[2, 1]: 8

[2, 2]: 3

Matrix 1:

3 5

7 5

Matrix 2:

6 9

8 3

Sum of 2 matrices:

9 14

15 8

TQC+ 程式語言 Python 610 平均溫度

請撰寫一程式，讓使用者輸入四週各三天的溫度，接著計算並輸出這四週的平均溫度及最高、最低溫度。

提示 1：平均溫度輸出到小數點後第二位。

提示 2：最高溫度及最低溫度的輸出，如為 31 時，則輸出 31，如為 31.1 時，則輸出 31.1。

```
Temp = []
```

```
for i in range(4):
```

```
    print("Week {}".format(i+1))
```

```
    for j in range(3):
```

```
        Temp.append(eval(input("Day {}".format(j+1))))
```

```
print("Average: {:.2f}".format(sum(Temp) / len(Temp)))
```

```
print("Highest: {}".format(max(Temp)))
```

```
print("Lowest: {}".format(min(Temp)))
```

Week 1:

Day 1: 23.1

Day 2: 24

Day 3: 23.5

Week 2:

Day 1: 32

Day 2: 33

Day 3: 35.3

Week 3:

Day 1: 29

Day 2: 30

Day 3: 26

Week 4:

Day 1: 27.6

Day 2: 25

Day 3: 28.8

Average: 28.11

Highest: 35.3

Lowest: 23.1

TQC+ 程式語言 Python 第 7 類：數組 (Tuple)、集合 (Set) 以及詞典 (Dictionary)

TQC+ 程式語言 Python 701 串列數組轉換

請撰寫一程式，輸入數個整數並儲存至串列中，以輸入 **-9999** 為結束點（串列中不包含 -9999），再將此串列轉換成數組，最後顯示該數組以及其長度 (Length)、最大值 (Max)、最小值 (Min)、總和 (Sum)。

```
l = []
n = eval(input())
while n != -9999:
    l.append(n)
    n = eval(input())
```

```
numTup = tuple(l)
print(numTup)
print("Length:", len(numTup))
print("Max:", max(numTup))
print("Min:", min(numTup))
print("Sum:", sum(numTup))
```

範例輸入

```
-4
0
37
19
26
-43
9
-9999
```

範例輸出

```
(-4, 0, 37, 19, 26, -43, 9)
Length: 7
Max: 37
Min: -43
Sum: 44
```

TQC+ 程式語言 702 數組合併排序

請撰寫一程式，輸入並建立兩組數組，各以 **-9999** 為結束點（數組中不包含 -9999）。將此兩數組合併並從小到大排序之，顯示排序前的數組和排序後的串列。

```
numL1, numL2 = [], []
```

```
print("Create tuple1:")
n = eval(input())
while n != -9999:
    numL1.append(n)
    n = eval(input())
```

```
tup1 = tuple(numL1)
```

```
print("Create tuple2:")
n = eval(input())
while n != -9999:
    numL2.append(n)
    n = eval(input())
```

```
tup2 = tuple(numL2)
```

```
print("Combined tuple before sorting:", tup1+tup2)
print("Combined list after sorting:", sorted(tup1+tup2))
```

輸入與輸出會交雜如下，輸出的部份以粗體字表示

Create tuple1:

```
9
0
-1
3
8
-9999
```

Create tuple2:

```
28
16
39
56
78
88
-9999
```

Combined tuple before sorting: (9, 0, -1, 3, 8, 28, 16, 39, 56, 78, 88)

Combined list after sorting: [-1, 0, 3, 8, 9, 16, 28, 39, 56, 78, 88]

TQC+ 程式語言 Python 703 數組條件判斷

請撰寫一程式，輸入一些字串至數組（至少輸入五個字串），以字串"end"為結束點（數組中不包含字串"end"）。接著輸出該數組，再分別顯示該數組的第一個元素到第三個元素和倒數三個元素。

```
tmp=()
```

```
while True:
```

```
    x=input()
```

```
    if x == "end":
```

```
        break
```

```
    else:
```

```
        tmp+=(x,)
```

```
print(tmp)
```

```
print(tmp[0:3])
```

```
print(tmp[-3:])
```

範例輸入

```
president
dean
chair
staff
teacher
student
end
```

範例輸出

```
('president', 'dean', 'chair', 'staff', 'teacher', 'student')
('president', 'dean', 'chair')
('staff', 'teacher', 'student')
```

TQC+ 程式語言 Python 704 集合條件判斷

請撰寫一程式，輸入數個整數並儲存至集合，以輸入-9999 為結束點（集合中不包含-9999），最後顯示該集合的長度（Length）、最大值（Max）、最小值（Min）、總和（Sum）。

```
numSet = set()
```

```
while True:
```

```
    n = eval(input())
```

```
    if n == -9999:
```

```
        break
```

```
    numSet.add(n)
```

```
print("Length:",len(numSet))
```

```
print("Max:",max(numSet))
```

```
print("Min:",min(numSet))
```

```
print("Sum:",sum(numSet))
```

範例輸入

```
34
-23
29
7
0
-1
-9999
```

範例輸出

```
Length: 6
Max: 34
Min: -23
Sum: 46
```

TQC+ 程式語言 Python 705 子集合與超集合

請撰寫一程式，依序輸入五個、三個、九個整數，並各自儲存到集合 set1、set2、set3 中。接著回答：set2 是否為 set1 的子集合 (subset)？set3 是否為 set1 的超集合 (superset)？

```
s1,s2,s3=set(),set(),set()
```

```
print("Input to set1:")
```

```
for i in range(5):
```

```
    s1.add(eval(input()))
```

```
print("Input to set2:")
```

```
for i in range(3):
```

```
    s2.add(eval(input()))
```

```
print("Input to set3:")
```

```
for i in range(9):
```

```
    s3.add(eval(input()))
```

```
print("set2 is subset of set1:",s2 <= s1)
```

```
print("set3 is superset of set1:",s3 >= s1)
```

Input to set1:

3

28

-2

7

39

Input to set2:

2

77

0

Input to set3:

3

28

12

99

39

7

-1

-2

65

set2 is subset of set1: False

set3 is superset of set1: True

TQC+ 程式語言 Python 706 全字母句

全字母句 (Pangram) 是英文字母表所有的字母都出現至少一次 (最好只出現一次) 的句子。請撰寫一程式，要求使用者輸入一正整數 k (代表有 k 筆測試資料)，每一筆測試資料為一句子，程式判斷該句子是否為 Pangram，並印出對應結果 True (若是) 或 False (若不是)。

提示：不區分大小寫字母

```
a = int(input())
```

```
for i in range(a):
```

```
    str1 = set(input().upper())
```

```
    print(len(str1) == 27)
```

輸入與輸出會交雜如下，輸出的部份以粗體字表示 第1組

3

The quick brown fox jumps over the lazy dog

True

Learning Python is funny

False

Pack my box with five dozen liquor jugs

True

輸入與輸出會交雜如下，輸出的部份以粗體字表示 第2組

2

Quick fox jumps nightly above wizard

True

These can be weapons of terror

False

TQC+ 程式語言 Python 707 共同科目

請撰寫一程式，輸入 X 組和 Y 組各自的科目至集合中，以字串"end"作為結束點（集合中不包含字串"end"）。請依序分行顯示(1)X 組和 Y 組的所有科目、(2)X 組和 Y 組的共同科目、(3)Y 組有但 X 組沒有的科目，以及(4)X 組和 Y 組彼此沒有的科目（不包含相同科目）。

```
x,y = set(),set()
```

```
print("Enter group X's subjects:")
```

```
s1 = input()
```

```
while s1 != "end":
```

```
    x.add(s1)
```

```
    s1 = input()
```

```
print("Enter group Y's subjects:")
```

```
s2 = input()
```

```
while s2 != "end":
```

```
    y.add(s2)
```

```
    s2 = input()
```

```
print(sorted(x|y))
```

```
print(sorted(x&y))
```

```
print(sorted(y-x))
```

```
print(sorted(x^y))
```

輸入與輸出會交雜如下，輸出的部份以粗體字表示

Enter group X's subjects:

Math

Literature

English

History

Geography

end

Enter group Y's subjects:

Math

Literature

Chinese

Physical

Chemistry

end

['Chemistry', 'Chinese', 'English', 'Geography', 'History', 'Literature', 'Math', 'Physical']

['Literature', 'Math']

['Chemistry', 'Chinese', 'Physical']

['Chemistry', 'Chinese', 'English', 'Geography', 'History', 'Physical']

TQC+ 程式語言 Python 708 詞典合併

請撰寫一程式，自行輸入兩個詞典（以輸入鍵值"end"作為輸入結束點，詞典中將不包含鍵值"end"），將此兩詞典合併，並根據 key 值字母由小到大排序輸出，如有重複 key 值，後輸入的 key 值將覆蓋前一 key 值。

```
d1,d2 = {},{}
```

```
print("Create dict1:")
```

```
k1 = input("Key: ")
```

```
while k1 != "end":
```

```
    d1[k1] = input("Value: ")
```

```
    k1 = input("Key: ")
```

```
print("Create dict2:")
```

```
k2 = input("Key: ")
```

```
while k2 != "end":
```

```
    d2[k2] = input("Value: ")
```

```
    k2 = input("Key: ")
```

```
d1.update(d2)
```

```
for i in sorted(d1.keys()):
```

```
    print(i,": ",d1[i],sep="")
```

輸入與輸出會交雜如下，輸出的部份以粗體字表示

Create dict1:

Key: a

Value: apple

Key: b

Value: banana

Key: d

Value: durian

Key: end

Create dict2:

Key: c

Value: cat

Key: e

Value: elephant

Key: end

a: apple

b: banana

c: cat

d: durian

e: elephant

TQC+ 程式語言 Python 709 詞典排序

請撰寫一程式，輸入一顏色詞典 color_dict (以輸入鍵值"end"作為輸入結束點，詞典中將不包含鍵值"end")，再根據 key 值的字母由小到大排序並輸出。輸入與輸出會交雜如下，輸出的部份以粗體字表示

```
color_dict = {}
dKey = input("Key: ")
while dKey != "end":
    dValue = input("Value: ")
    color_dict[dKey] = dValue
    dKey = input("Key: ")

for i in sorted(color_dict.keys()):
    print(i,": ",color_dict[i],sep="")
```

```
Key: Green Yellow
Value: #ADFF2F
Key: Snow
Value: #FFFAFA
Key: Gold
Value: #FFD700
Key: Red
Value: #FF0000
Key: White
Value: #FFFFFF
Key: Green
Value: #008000
Key: Black
Value: #000000
Key: end
Black: #000000
Gold: #FFD700
Green: #008000
Green Yellow: #ADFF2F
Red: #FF0000
Snow: #FFFAFA
White: #FFFFFF
```

TQC+ 程式語言 Python 710 詞典搜尋

請撰寫一程式，為一詞典輸入資料 (以輸入鍵值"end"作為輸入結束點，詞典中將不包含鍵值"end")，再輸入一鍵值並檢視此鍵值是否存在於該詞典中。

```
d1 = {}
k1 = input("Key: ")
while k1 != "end":
    d1[k1] = input("Value: ")
    k1 = input("Key: ")

print(input("Search key: ") in d1.keys())
```

輸入與輸出會交雜如下，輸出的部份以粗體字表示

```
Key: 123-4567-89
Value: Jennifer
Key: 987-6543-21
Value: Tommy
Key: 246-8246-82
Value: Kay
Key: end
Search key: 246-8246-82
True
```

TQC+ 程式語言 Python 第 8 類：字串(String)的運作

TQC+ 程式語言 Python 801 字串索引

請撰寫一程式，要求使用者輸入一字串，顯示該字串每個字元的索引

```
word = input()
for i in range(len(word)):
    print("Index of '{}': {}".format(word[i],i))
```

範例輸入

Sandwich

範例輸出

```
Index of 'S': 0
Index of 'a': 1
Index of 'n': 2
Index of 'd': 3
Index of 'w': 4
Index of 'i': 5
Index of 'c': 6
Index of 'h': 7
```

TQC+ 程式語言 Python 802 字元對應

請撰寫一程式，要求使用者輸入一字串，顯示該字串每個字元的對應 ASCII 碼及其總和。

```
word = input()
total = 0
for i in range(len(word)):
    print("ASCII code for '{}' is {}".format(word[i],ord(word[i])))
    total += ord(word[i])
print(total)
```

範例輸入

Kingdom

範例輸出

```
ASCII code for 'K' is 75
ASCII code for 'i' is 105
ASCII code for 'n' is 110
ASCII code for 'g' is 103
ASCII code for 'd' is 100
ASCII code for 'o' is 111
ASCII code for 'm' is 109
713
```

TQC+ 程式語言 Python 803 倒數三個詞

請撰寫一程式，讓使用者輸入一個句子（至少有五個詞，以空白隔開），並輸出該句子倒數三個詞。

```
sentence = input().split(" ")
print(sentence[-3],sentence[-2],sentence[-1])
```

範例輸入

Many foreign students study in FJU

範例輸出

study in FJU

TQC+ 程式語言 Python 804 大寫轉換

請撰寫一程式，讓使用者輸入一字串，分別將該字串轉換成全部大寫以及每個字的第一個字母大寫。

```
n=input()
print(n.upper())
print(n.title())
```

範例輸入

learning python is funny

範例輸出

```
LEARNING PYTHON IS FUNNY
Learning Python Is Funny
```

TQC+ 程式語言 Python 805 字串輸出

請撰寫一程式，要求使用者輸入一個長度為 6 的字串，將此字串分別置於 10 個欄位的寬度的左邊、中間和右邊，並顯示這三個結果，左右皆以直線 | (Vertical bar) 作為邊界。

```
word = input()
print("|{:10}|".format(word))
print("|{:>10}|".format(word))
print("|{:<10}|".format(word))
```

```
python
|python·····|
|··python···|
|····python|
```

TQC+ 程式語言 Python 806 字元次數計算

請撰寫一程式，讓使用者輸入一字串和一字元，並將此字串及字元作為參數傳遞給名為 compute() 的函式，此函式將回傳該字串中指定字元出現的次數，接著再輸出結果。

```
compute = lambda x,y : x.count(y)
```

```
x = list(input())
y = input()
print("{:s} occurs {:d} time(s)".format(y,compute(x,y)))
```

範例輸入

```
Our country is beautiful
u
```

範例輸出

```
u occurs 4 time(s)
```

TQC+ 程式語言 Python 807 字串加總

請撰寫一程式，要求使用者輸入一字串，該字串為五個數字，以空白隔開。請將此五個數字加總 (Total) 並計算平均 (Average)。

```
s = list(input().split(" "))
```

```
all = 0
for i in range(len(s)):
    all += int(s[i])

print("Total =",all)
print("Average =",all/len(s))
```

範例輸入

```
-2 34 18 29 -56
```

範例輸出

```
Total = 23
Average = 4.6
```

TQC+ 程式語言 Python 808 社會安全碼

請撰寫一程式，提示使用者輸入一個社會安全碼 SSN，格式為 ddd-dd-dddd，d 表示數字。若格式完全符合 (正確的 SSN) 則顯示 【Valid SSN】，否則顯示 【Invalid SSN】。

```
num = input().replace("-", "")

print("Valid SSN") if num.isdigit() == True else print("Invalid SSN")
```

範例輸入1

```
329-48-4977
```

範例輸出1

```
Valid SSN
```

範例輸入2

```
837-a3-3000
```

範例輸出2

```
Invalid SSN
```

TQC+ 程式語言 Python 809 密碼規則

請撰寫一程式，要求使用者輸入一個密碼（字串），檢查此密碼是否符合規則。密碼規則如下：

- 必須至少八個字元。
- 只包含英文字母和數字。
- 至少要有一個大寫英文字母。
- 若符合上述三項規則，程式將顯示檢查結果為【Valid password】，否則顯示【Invalid password】。

```
psword,uppernum = input(),0
```

```
for i in range(len(psword)):
```

```
    if psword[i].isupper():
```

```
        uppernum = 1
```

```
if len(psword) >= 8 and psword.isalnum() and uppernum == 1:
```

```
    print("Valid password")
```

```
else :
```

```
    print("Invalid password")
```

範例輸入1

39Gfjkd98

範例輸出1

Valid password

範例輸入2

39dk8fh

範例輸出2

Invalid password

TQC+ 程式語言 Python 810 最大值與最小值之差

請撰寫一程式，首先要求使用者輸入正整數 k ($1 \leq k \leq 100$)，代表有 k 筆測試資料。每一筆測試資料是一串數字，每個數字之間以一空白區隔，請找出此串列數字中最大值和最小值之間的差。

提示：差值輸出到小數點後第二位。

```
for i in range(int(input())):
```

```
    num1 = input().split(" ")
```

```
    num1 = list(map(eval,num1))
```

```
    print("{:.2f}".format(max(num1) - min(num1)))
```

輸入與輸出會交雜如下，輸出的部份以粗體字表示

4

94 52.9 3.14 77 46

90.86

-2 0 1000.34 -14.4 89 50

1014.74

87.78 33333 29.3

33303.70

9998 9996 9999

3.00

TQC+ 程式語言 Python 第 9 類：檔案與異常處理

TQC+ 程式語言 Python 901 成績資料

請撰寫一程式，將使用者輸入的五筆資料寫入到 write.txt (若不存在，則讓程式建立它)，每一筆資料為一行，包含學生名字和期末總分，以空白隔開。檔案寫入完成後要關閉。

範例輸出

```
with open("write.txt","w") as file:
    for i in range(5):
        file.write("{} {}".format(input()+"\n"))
```

範例輸入

```
Leon 87
Ben 90
Sam 77
Karen 92
Kelena 92
```



TQC+ 程式語言 Python 902 資料加總

請撰寫一程式，讀取 read.txt 的內容 (內容為數字，以空白分隔) 並將這些數字加總後輸出。檔案讀取完成後要關閉。

```
with open("read.txt","r") as file:
    sum1 = list(map(int,file.readline().split()))
print(sum(sum1))
```

範例輸入

無

範例輸出

660

TQC+ 程式語言 Python 903 成績資料

請撰寫一程式，要求使用者輸入五個人的名字並加入到 data.txt 的尾端。之後再顯示此檔案的內容。

範例輸出

```
with open("data.txt","a") as file:
    for i in range(5):
        file.write("\n"+input())
print("Append completed!")
print('Content of "data.txt":')
```

範例輸入

```
with open("data.txt","r") as file:
    for i in file:
        print(i,end="")
```

範例輸入

```
Daisy
Kelvin
Tom
Joyce
Sarah
```

範例輸出

```
Append completed!
Content of "data.txt":
Ben
Cathy
Tony
Daisy
Kelvin
Tom
Joyce
Sarah
```

TQC+ 程式語言 Python 904 資料計算

請撰寫一程式，讀取 read.txt（每一列的格式為名字和身高、體重，以空白分隔）並顯示檔案內容、所有人的平均身高、平均體重以及最高者、最重者。提示：輸出浮點數到小數點後第二位。

```
name = []
height = []
weight = []
data = []
with open ("read.txt","r",encoding = "UTF-8") as file:
    for line in file:
        print(line)
        data = line.replace("\n","").split(" ")
        name.append(data[0])
        height.append(eval(data[1]))
        weight.append(eval(data[2]))
print("Average height :",sum(height)/len(height))
print("Average weight:",sum(weight)/len(weight))
TT = height.index(max(height))
TM = weight.index(max(weight))
print("The tallest is {} with {}cm".format(name[TT],height[TT]))
print("The heaviest is {} with {}kg".format(name[TM],weight[TM]))
```

範例輸入

無

範例輸出

```
Ben 175 65
Cathy 155 55

Tony 172 75
Average height: 167.33
Average weight: 65.00
The tallest is Ben with 175.00cm
The heaviest is Tony with 75.00kg
```

TQC+ 程式語言 Python 905 字串資料刪除

請撰寫一程式，要求使用者輸入檔案名稱 data.txt 和一字串 s，顯示該檔案的內容。接著刪除檔案中的字串 s，顯示刪除後的檔案內容並存檔。

```
ff,ss = input(),input()
with open (ff,"r",encoding = "UTF-8") as file:
    data = file.read()
print("=== Before the deletion")
print(data)

data = data.replace(ss,"")
print("=== After the deletion")
print(data)

with open (ff,"w",encoding = "UTF-8") as file:
    file.write(data)
```

範例輸入1

```
data.txt
Tomato
```

範例輸出1

```
=== Before the deletion
Apple Kiwi Banana
Tomato Pear Durian

=== After the deletion
Apple Kiwi Banana
Pear Durian
```

範例輸入2

```
data.txt
Kiwi
```

範例輸出2

```
=== Before the deletion
Apple Kiwi Banana
Tomato Pear Durian

=== After the deletion
Apple Banana
Tomato Pear Durian
```

TQC+ 程式語言 Python 906 字串資料取代

請撰寫一程式，要求使用者輸入檔名 data.txt、字串 s1 和字串 s2。程式將檔案中的字串 s1 以 s2 取代之。

```
ff,s1,s2=input(),input(),input()
```

```
with open(ff,"r",encoding ="UTF-8") as file:
```

範例輸入

```
data.txt
pen
sneakers
```

```
data = file.read()

print("=== Before the replacement")
print(data)

data = data.replace(s1,s2)
print("=== After the replacement")
print(data)
```

範例輸出

```
=== Before the replacement
watch shoes skirt
pen trunks pants
=== After the replacement
watch shoes skirt
sneakers trunks pants
```

TQC+ 程式語言 Python 907 詳細資料顯示

請撰寫一程式，要求使用者輸入檔名 `read.txt`，顯示該檔案的行數、單字數（簡單起見，單字以空白隔開即可，忽略其它標點符號）以及字元數（不含空白）。

```
ff = input()

fp = open(ff,"r",encoding = "UTF-8")
l,w,c =0,0,0
for i in fp:
    l += 1
    line_sp = i.split()
    w += len(line_sp)
    for j in range(len(line_sp)):
        c += len(line_sp[j])

print(l,"line(s)")
print(w,"word(s)")
print(c,"character(s)")
fp.close()
```

範例輸入

```
read.txt
```

範例輸出

```
6 line(s)
102 word(s)
614 character(s)
```

TQC+ 程式語言 Python 908 單字次數計算

請撰寫一程式，要求使用者輸入檔名 `read.txt`，以及檔案中某單字出現的次數。輸出符合次數的單字，並依單字的第一個字母大小排序。（單字的判斷以空白隔開即可）

```
ff = input()
n = int(input())
with open (ff,"r",encoding = "UTF-8") as file:
    data = sorted(file.read().split())
for i in sorted(set(data)):
    if data.count(i) == n:
        print(i)
```

範例輸入

```
read.txt
3
```

範例輸出

```
a
is
programming
```

TQC+ 程式語言 Python 909 聯絡人資料

請撰寫一程式，將使用者輸入的五個人的資料寫入 `data.dat` 檔，每一個人的資料為姓名和電話號碼，以空白分隔。再將檔案加以讀取並顯示檔案內容。

```
with open ("data.dat","w",encoding ="UTF-8") as file:
    for i in range(5):
        file.write(input()+"\n")

print("The content of \"data.dat\":")
with open ("data.dat","r",encoding ="UTF-8") as file:
    for i in file:
        print(i)
```

範例輸出

```
The content of "data.dat":
Karen 123456789
Bonnie 235689147
Simon 987612345
Louis 675489321
Andy 019238475
```

範例輸入

```
Karen 123456789
Bonnie 235689147
Simon 987612345
Louis 675489321
Andy 019238475
```



TQC+ 程式語言 Python 910 學生基本資料

請撰寫一程式，要求使用者讀入 read.dat (以 UTF-8 編碼格式讀取)，第一列為欄位名稱，第二列之後是個人記錄。請輸出檔案內容並顯示男生人數和女生人數(根據"性別"欄位，0 為女性、1 為男性)。

```
with open ("read.dat","r",encoding ="UTF-8") as file:
```

```
    content = file.readlines()
```

```
for i in content:
```

```
    print(i)
```

```
with open ("read.dat","r",encoding ="UTF-8") as file:
```

```
    data = file.read().split()
```

```
print("Number of males:",data.count("1"))
```

```
print("Number of females:",data.count("0"))
```

範例輸入

無

範例輸出

```
學號 姓名 性別 科系
101 陳小華 0 餐旅管理
202 李小安 1 廣告
303 張小威 1 英文
404 羅小美 0 法文
505 陳小凱 1 日文
Number of males: 3
Number of females: 2
```