

```
# 單位換算，打 dozen
```

```
n = eval(input())
if n % 12 == 0:
    print('%d dozen' %(n/12))
else:
    print('%d dozen and %d' %(n/12,n%12))
```

```
# 判斷成績是否及格 III
```

```
while True:
    roll=eval(input())
    if roll!=1 and roll!=2:
        print('roll error')
        break
    score=input()
    if roll==1:
        while True:
            if(int(score)<0)or(int(score)>100):
                print('score error')
                break
            elif(int(score)<60):
                print('fail')
                break
            else:
                print('pass')
                break
        break
    elif roll==2:
        while True:
            if(int(score)<0)or(int(score)>100):
                print('score error')
                break
            elif(int(score)<70):
                print('fail')
                break
            else:
                print('pass')
                break
        break
```

```
# 難免同籠
```

```
n = input()
nlist = n.split()
nfst = int(nlist[0])
nsnd = int(nlist[1])
anslist = []

for i in range(0 ,nfst+1):
    rhead = nfst - i
    if nsnd == rhead * 4 + i * 2:
        anslist.append(i)
        anslist.append(rhead)
if anslist == []:
    print('NO')
else:
    ans1 = int(anslist[0])
    ans2 = int(anslist[1])
    print('YES')
    print(ans1, ans2)
```

```
# 1 加到 n
```

```
"""
this is my code

n = eval(input())
total = 0
for i in range(1, n+1):
    total += i
    if (i <= (n - 1)):
        print(i,'+',end = '')
    else:
        print(i, end = '')
print(' = ', total)

"""

n = eval(input())
total = 0

for i in range (1, 1+n):
    total += i
    print(i, end = '')
    if(i < (n)):
        print('+', end = '')
print(' = ', total)
```

```
# 直到輸入q後程式結束 -while
```

```
while True:
    x=input()
    print(x)
    if x=='q':
        break
```

```
# 豪華聖誕樹
```

```
k = eval(input())
for p in range(1,k):
    print(' ', end = '')
print('*')
for box in range(1,k-2+1):
    for item in (2*box-1,2*box+1,2*box+3):
        for spaceAmount in range(1, int(k-(item-1)/2)):
            print(' ', end = '')
        for pItem in range(1,item+1):
            print('^', end = '')
        print()
for b in range(1, k-1):
    for a in range(1, int((k+3)/2)):
        print(' ', end = '')
    for c in range(k-2):
        print('#', end = '')
    print()
```

```
# n 階乘
```

```
n = eval(input())
total = 1
for i in range (1,n+1):
    total *= i
print(total)
```

```
# 數字轉文字輸出(中文大寫數字輸出)
```

```
n = eval(input())
while True:
    if n < 1 or n > 99999:
        print('out of range')
        break
    n1 = n//10000
    n2 = (n-n1*10000)//1000
    n3 = (n-n1*10000-n2*1000)//100
    n4 = (n-n1*10000-n2*1000-n3*100)//10
    n5 = (n-n1*10000-n2*1000-n3*100-n4*10)
    for i in (n1, n2, n3, n4, n5):
        if i==0:
            continue
        continue
    if i==1:
        print('壹', end = '')
    if n1>0:
        print('萬', end = '')
        n1 = 0
        continue
    if n2>0:
        print('仟', end = '')
        n2 = 0
        continue
    if n3>0:
        print('佰', end = '')
        n3 = 0
        continue
    if n4>0:
        print('拾', end = '')
        n4 = 0
        continue
    if i==2:
        print('貳', end = '')
    if n1>0:
        print('萬', end = '')
        n1 = 0
        continue
    if n2>0:
        print('仟', end = '')
        n2 = 0
        continue
    if n3>0:
        print('佰', end = '')
        n3 = 0
        continue
    if n4>0:
        print('拾', end = '')
        n4 = 0
        continue
    if i==9:
        print('玖', end = '')
    if n1>0:
        print('萬', end = '')
        n1 = 0
        continue
    if n2>0:
        print('仟', end = '')
        n2 = 0
        continue
    if n3>0:
        print('佰', end = '')
        n3 = 0
        continue
    if n4>0:
        print('拾', end = '')
        n4 = 0
        continue
    print('元整!')
break
```

```
# 2到m之間的質數
```

```
n = eval(input())
for nums in range(1, n+1):
    fct = 0
    for i in range(1,n+1):
        if nums%i ==0:
            fct += 1
    if fct ==2:
        print(nums, 'is prime')
```

```
# 猜數字進階版-終極密碼
```

```
m = eval(input())
ansMax=100
ansMin=1
guessAnswer = m
guessNum=0
while guessNum != guessAnswer:
    print(ansMin, '< ? <',ansMax)
    guessNum=eval(input())
    if guessNum > ansMin and guessNum < ansMax:
        if guessNum > guessAnswer:
            print('wrong answer, guess smaller')
            ansMax = guessNum
        elif guessNum < guessAnswer:
            print('wrong answer, guess larger')
            ansMin = guessNum
        else:
            print('bingo answer is', m)
```

```
# 三角形判斷
```

```
a = eval(input())
b = eval(input())
c = eval(input())
if a+b>c and b+c>a and a+c>b:
    print('True')
    if a>b and a>c:
        if a*a == b*b + c*c:
            print('Right Triangle')
elif b>c and b>a:
    if b*b == a*a + c*c:
        print('Right Triangle')
elif c>a and c>b:
    if c*c == a*a + b*b:
        print('Right Triangle')
else:
    print('Non-Right Triangle')
```

```
else:
```

```
# 例外處理-兩數相除
```

```
while True:
    print('False')
    break
    # while True:
    #     try:
    a = eval(input())
    b = int(input())
    c = a/b
    print('%d/%d = %.2f' %(a, b, c))
    break
except ZeroDivisionError:
    print('ZeroDivisionError')
except SyntaxError:
    print('SyntaxError')
except NameError:
    print('NameError')
except ValueError:
    print('ValueError')
```

```
# 三角數字列印
```

```
...
印出
1
2 3 4
5 6 7 8 9 10
三角形的函式
(但還是需要修改，這邊 kplus 數量不對)
```

```
n = eval(input())
k = 0
if n == 1:
    while True:
        print('1')
        break
else:
    for j in range (1,n+1):
        for i in range (1,j+1):
            for kplus in range(1,i+1):
                k += 1
                print (k, ' ',end = '')
        print()
...
n = eval(input())
k = 0
if n == 1:
    while True:
        print('1')
        break
else:
```

```
    for j in range (1,n+1): # 幾行
        for i in range (1,j+1): # 印幾個數字
            for p in range(1,2): # p 永遠都是1
                k += 1
                print (k, ' ', end = '')
        print()
```

```
# 但這個答案是錯的，正確答案在下面
```

```
n = eval(input())
k = 0
if n == 1:
    while True:
        print('1')
        break
else:
    for j in range (1,n+1):
        for i in range (1,j+1):
            for p in range(1,2):
                k += 1
                print (k, ' ', end = '',sep=' ')#因為你沒設定這邊的sep=' '，不然這樣會多出一個空
                白
        print()
```

```
#空白太多了！
```

```
# 小當家買食材
n1 = eval(input())
lst = []
dict01 = {}
for i in range(n1):
    n2 = input()
    n3 = n2.strip().split()
    lst.append(n3)
for i in lst:
    if i[0] not in dict01:
        dict01[i[0]] = i[1]
    else:
        y = int(dict01[i[0]]) + int(i[1])
        dict01[i[0]] = y
keyLst = []
valueLst = []
for i in dict01:
    keyLst.append(i)
    valueLst.append(int(dict01[i]))
valueLst2 = valueLst.copy()
valueLst2.sort()
ans = valueLst2[(len(valueLst2)-1)]
ansIndex = valueLst.index(ans)
print(keyLst[ansIndex],ans, end = '\n')
```

```
# AB猜數字遊戲
```

```
ans = input()
ansInt = eval(ans)
ans1 = ansInt//1000
ans2 = (ansInt - ans1*1000)//100
ans3 = (ansInt - ans1*1000 - ans2*100)//10
ans4 = ansInt - ans1*1000 - ans2*100 - ans3*10
ansLst = [ans1, ans2, ans3, ans4]
while True:
    guess = input()
    guessInt = eval(guess)
    if guessInt == ansInt:
        print('4A0B', end = '\n')
        print('You Win!')
        break
    else:
        g1 = guessInt//1000
        g2 = (guessInt - g1*1000)//100
        g3 = (guessInt - g1*1000 - g2*100)//10
        g4 = guessInt - g1*1000 - g2*100 - g3*10
        gLst = [g1, g2, g3, g4]
        act = 0
        bct = 0
        for g in range(4):
            for ans in ansLst:
                if gLst[g] == ans:
                    if g == ansLst.index(ans):
                        act += 1
                        continue
                    else:
                        bct += 1
                        continue
        bMinus = 0
        for n in gLst:
            if ansLst.count(n) > 0 and gLst.count(n) > 1:
                bMinus = gLst.count(n)-1
        bct -= bMinus
        print(act,end='')
        print('A',end='')
        print(bct,end='')
        print('B',end='\n')
        continue
```

```
# 文字外殼包裝
```

```
lst = []
lenlst = []
while True:
    n = input()
    if n == '-1':
        break
    n.strip().split(sep = ' ')
    lst.append(n)
sym = input()
for i in lst:
    lenlst.append(len(i))
maxLen = max(lenlst)
for pFst in range(maxLen+2):
    if pFst == maxLen+1:
        print(sym, end = '\n')
    else:
        print(sym, end = ' ')
for p in lst:
    print(sym, end = ' ')
    print(p,end = ' ')
    for space in range(maxLen-len(p)):
        print(' ',end = ' ')
    print(sym)
```

```

# 阿宅的書單
nlst = []
m1 = input()
m = m1.lower()
while True:
    n = input()
    if n == 'q':
        if nlst.count(m) != 0:
            print('Yes',end = ' ')
            print(nlst.index(m)+1)
            break
        else:
            sumN = 0
            for i in nlst:
                sumN += 1
            print('No',end = ' ')
            print(sumN)
            break
    else:
        nlst.append(n)

# 標準函式庫
import time
n = eval(input())
print(time.strftime('%Y-%m-%d %H:%M:%S', time.gmtime(n)))
print(time.strftime('%Y-%m-%d %H:%M:%S', time.localtime(n)))

# 系統內建os模組練習
import shutil
import os
n = eval(input())
if os.path.exists('files'):
    shutil.rmtree('files')
os.mkdir('files')
os.chdir('files')
for i in range(1,(n+1)):
    os.mkdir('f'+ str(i))
folder = os.listdir()
folder.sort()
print(folder)
os.rename('f1','folder1')
folder2 = os.listdir()
folder2.sort()
print(folder2)
os.rmdir('folder1')
folder3 = os.listdir()
folder3.sort()
print(folder3)

# 身分證檢驗
n = input()
n = n.upper()
if len(list(n)) != 10:
    print('fake')
else:
    first = n[0]
    dic = {'A':10,'J':18,'S':26,
           'B':11,'K':19,'T':27,
           'C':12,'L':20,'U':28,
           'D':13,'M':21,'V':29,
           'E':14,'N':22,'W':32,
           'F':15,'O':35,'X':30,
           'G':16,'P':23,'Y':31,
           'H':17,'Q':24,'Z':33,
           'I':34,'R':25}
    firstnum = (dic[first])
    firstnum01 = (firstnum//10)
    firstnum02 = (firstnum%10)
    lettersum = firstnum01*1+firstnum02*9
    k = lettersum + int(n[1])*8 + int(n[2])*7 + int(n[3])*6 + int(n[4])*5 + int(n[5])*4
    + int(n[6])*3 + int(n[7])*2 + int(n[8])*1 + int(n[9])*1
    if k%10 == 0:
        print('real')
    else:
        print('fake')

# Set 找出兩籃水果中沒有重複的水果
itemsA = {"蘋果","香蕉","鳳梨","芭樂"}
itemsB = {"鳳梨","蘋果","水梨","蓮霧"}

iA01 = input()
iA02 = input()
iB01 = input()
iB02 = input()

itemsA.add(iA01)
itemsA.add(iA02)
itemsA.remove('蘋果')
iAlist = list(itemsA)
iAlist.sort()
print('iA:',iAlist)

itemsB.add(iB01)
itemsB.add(iB02)
itemsB.remove('蓮霧')
iBlist = list(itemsB)
iBlist.sort()
print('iB:',iBlist)

u = itemsA.union(itemsB)
ulist = list(u)
ulist.sort()
print('union:', ulist)

intersec = itemsA.intersection(itemsB)
intersecLst = list(intersec)
intersecLst.sort()
print('intersection:', intersecLst)

dif01 = list(itemsA.difference(itemsB))
dif01.sort()
print('A diff B:', dif01)

dif02 = list(itemsB.difference(itemsA))
dif02.sort()
print('B diff A:', dif02)

ansXor = u - intersec
lst = list(ansXor)
lst.sort()
print('A xor B:', lst)

# dict字典練習II (C++ STL map)
dict01 = {'P':'Pikachu', 'M':'Mickey Mouse', 'H':'Hello kitty'}
while True:
    n = input()
    if n == '-1':
        break
    elif n in dict01:
        print(dict01[n])
        continue
    elif n not in dict01:
        print(n, 'does not exist. Enter a new one!')
        m = input()
        dict01[n] = m

```

成績計算與列印-二維陣列-輸入版

```

a = input()
b = input()
c = input()
d = input()
e = input()
a1 = a.strip().split()
b1 = b.strip().split()
c1 = c.strip().split()
d1 = d.strip().split()
e1 = e.strip().split()
sumValLst = []
avgLst = []
for m in (a1,b1,c1,d1,e1):
    val = int(m[0])+int(m[1])+int(m[2])
    sumValLst.append(val)
    avg = val/3
    avgLst.append(avg)
sumSum = 0
for i in sumValLst:
    sumSum += int(i)
gradeLst = [a1,b1,c1,d1,e1]
for i in range(5):
    print('student',i+1,end = '\n')
    for j in gradeLst:
        print(' 1:', j[0])#前面少一個空格
        print(' 2:', j[1])#前面少一個空格
        print(' 3:', j[2])#前面少一個空格
        gradeLst.remove(j)
        break
    print(' sum:',sumValLst[i], end = '\n')
    print(' avg:',%.2f%(avgLst[i]), end = '\n')#前面少一個
print('total:',sumSum, end = '')
print(' avg: %.2f%(sumSum/15), end = '\n')
avgLst2 = avgLst.copy()
avgLst2.sort()
stu = avgLst.index(avgLst2[4])+1
print
print('highest avg: student', stu, end = '')
print(' : %.2f%(avgLst2[4]), end = '\n')

```

公司大危機

```

n=int(input())
Money_List=[int(x) for x in input().strip().split()]

```

#這是小迴圈，把所有的list中元素轉成int，
#但是如果不會這樣寫的話，可以用一般for迴圈來處理

```

m=int(input())
Leave=[int(x) for x in input().strip().split()]

```

```

if 0 in Leave: #老師題目出錯，可能會有第0位的人，但事實上不存在，我們要把它去除
    Leave.remove(0)

```

for i in Leave:#把離開的人金額歸零

```
    Money_List[i-1] = 0
```

total = sum(Money_List)

Max = max(Money_List)

Max_Position = Money_List.index(Max) + 1

```

print("%d\n%d %d"%(total,Max_Position,Max))

```

讀寫*.csv檔 (檢查一組標準輸出與檔案輸出)

```

# ./stores_old.csv
f1 = open ('stores_old.csv', 'r', encoding = 'big5')
txtLst = f1.readlines()
f1.close()
for i in range(len(txtLst)):
    txtLst[i] = txtLst[i].strip()
for i in range(len(txtLst)):
    txtLst[i] = txtLst[i].split(',')
a=txtLst[0].index('sid')
b=txtLst[0].index('name')
c=txtLst[0].index('tel')
d=txtLst[0].index('wifi')

# ./ stores_new.csv
f2 = open ('stores_new.csv', 'w', encoding = 'utf-8')
for i in txtLst:
    f2.write(i[a])
    f2.write(',')
    f2.write(i[b])
    f2.write(',')
    f2.write(i[c])
    f2.write(',')
    f2.write(i[d])
    f2.write('\n')
f2.close()

f3 = open('stores_new.csv', 'r', encoding = 'utf-8')
txtLst2 = f3.readlines()
f3.close()
for i in range(len(txtLst2)):
    txtLst2[i] = txtLst2[i].strip()
for i in range(len(txtLst2)):
    txtLst2[i] = txtLst2[i].split(',')
    for i in txtLst2:
        print(i[0],end = '')
        print(',',end = '')
        print(i[1],end = '')
        print(',',end = '')
        print(i[2],end = '')
        print(',',end = '')
        print(i[3],end = '')
        print(',',end = '\n')

```

```

for elemLst in txtLst:
    print(elemLst[a],end = '')
    print(',',end = '')
    print(elemLst[b],end = '')
    print(',',end = '')
    print(elemLst[c],end = '')
    print(',',end = '')
    print(elemLst[d],end = '')
    print(',')

```

讀寫*.csv檔

```

f1 = open ('stores_old.csv', 'r', encoding = 'big5')
txtLst = f1.readlines()
f1.close()
for i in range(len(txtLst)):
    txtLst[i] = txtLst[i].strip()
for i in range(len(txtLst)):
    txtLst[i] = txtLst[i].split(',')
a=txtLst[0].index('sid')
b=txtLst[0].index('name')
c=txtLst[0].index('tel')
d=txtLst[0].index('wifi')

```

```
# 類別練習II 找出平均最高分
```

```
# 原本的答案系統說錯誤，因為空格問題
```

```
class student:  
    def __init__(self, name, gender,):  
        self.Name = name  
        self.Gender = gender  
        self.Grades = []
```

```
def avg(self):  
    return sum(self.Grades) / len(self.Grades)  
def add(self, score):  
    self.Grades.append(score)  
def fcount(self):  
    # 不及格的數量  
    fct = 0  
    for i in range(len(self.Grades)):  
        if self.Grades[i] < 60:  
            fct += 1  
    return fct  
def info(self):  
    infoAll = (self.Name, self.Gender, self.Grades, self.avg(), self.fcount())  
    return infoAll
```

```
# 矩陣相乘-函式練習
```

```
a1 = input().strip().split(sep = ' ')  
a2 = input().strip().split(sep = ' ')  
a3 = input().strip().split(sep = ' ')  
b1 = input().strip().split(sep = ' ')  
b2 = input().strip().split(sep = ' ')  
b3 = input().strip().split(sep = ' ')  
alst = [a1,a2,a3]  
blst = [b1,b2,b3]  
a = alst  
b = blst  
def matrixMultiPly(a, b) :  
    anslst = []  
    for m in range(3):  
        for n in range(3):  
            for k in range(3):  
                anslst.append(int(alst[m][k]) * int(blst[k][n]))  
    anslst1 = []  
    for a in range(0,len(anslst),3):  
        anslst1.append(anslst[a]+anslst[a+1]+anslst[a+2])  
    anslst2 = [anslst1[0],anslst1[1],anslst1[2]]  
    anslst3 = [anslst1[3],anslst1[4],anslst1[5]]  
    anslst4 = [anslst1[6],anslst1[7],anslst1[8]]  
    return [anslst2,anslst3,anslst4]
```

```
c = matrixMultiPly(a,b)
```

```
for i in c:  
    print(i,end = '\n')
```

```
# 這題不知道為何批改系統顯示錯誤，以下為助教給的答案
```

```
def matrixMultiPly(a,b):  
    n=len(a)  
    c=[[0]*n for row in range(n)]  
    for i in range(0, n):  
        for j in range(0, n):  
            for k in range(0, n):  
                c[i][j]=int(c[i][j])+int(a[i][k])*int(b[k][j])  
    return c
```

```
a1 = input().strip().split()  
a2 = input().strip().split()  
a3 = input().strip().split()  
a = [a1,a2,a3]
```

```
b1 = input().strip().split()  
b2 = input().strip().split()  
b3 = input().strip().split()  
b = [b1,b2,b3]
```

```
c=matrixMultiPly(a, b)  
for i in c:  
    print(i)
```

```
# 讀寫*.csv檔 並回傳 set
```

```
f1 = open('../app/math_list.csv','r',encoding = 'big5')  
lst1 = f1.readlines()  
f1.close()  
setNum1 = set()  
for i in range(len(lst1)):  
    lst1[i] = lst1[i].strip().split(sep = ',')  
    setNum1.add(lst1[i][0])  
setNum1.remove('學號')  
  
f2 = open('../app/english_list.csv','r',encoding = 'big5')  
lst2 = f2.readlines()  
f2.close()  
setNum2 = set()  
  
for i2 in range(len(lst2)):  
    lst2[i2] = lst2[i2].strip().split(sep = ',')  
    setNum2.add(lst2[i2][0])  
setNum2.remove('學號')
```

```
ans1 = list(setNum1.intersection(setNum2))  
ans1.sort()  
ans2 = list(setNum1.union(setNum2)-setNum1.intersection(setNum2))  
ans2.sort()  
ans3 = list(setNum1.difference(setNum2))  
ans3.sort()  
ans4 = list(setNum2.difference(setNum1))  
ans4.sort()  
ans5 = list(setNum1.union(setNum2))  
ans5.sort()
```

```
print(ans1)  
print(ans2)  
print(ans3)  
print(ans4)  
print(ans5)
```

```
# 讀取txt-尋找班上黑馬(Dark Horse)
```

```
file1 = input()
file2 = input()
f1txt = './3035/' + file1    #../app/MathScoreMid01.txt
f2txt = './3035/' + file2    # ../app/MathScoreFinal01.txt

f1 = open(f1txt,'r')
txtLst1 = f1.readlines()
f1.close()
stu = []
lst1 = []
lst2 = []
lst3 = []
for i in range(len(txtLst1)):
    txtLst1[i] = txtLst1[i].strip().split(sep = ',')

txtLst1.pop(0)

for i in txtLst1:
    lst1.append(int(i[1]))
    stu.append(i[0])

f1 = open(f2txt,'r')
txtLst2 = f1.readlines()
f1.close()

for i in range(len(txtLst2)):
    txtLst2[i] = txtLst2[i].strip().split(sep = ',')
txtLst2.pop(0)

for i in txtLst2:
    lst2.append(int(i[1]))
    stu.append(i[0])

for i in range(len(lst1)):
    lst3.append(lst2[i]-lst1[i])
maxScore = max(lst3)
stuLst = []
print('Dark Horse: ', end = '')

for i in txtLst2:
    if lst3.count(maxScore) > 1:
        for i in range(lst3.count(maxScore)):
            stuLst.append(stu[lst3.index(maxScore)])
            maxInd = lst3.index(maxScore)
            lst3.pop(maxInd)
            lst3.insert(maxInd,0)
            stuLst.sort()
        for i in stuLst:
            if stuLst.index(i) == len(stuLst)-1:
                print(i, end = '\n')
            else:
                print(i, end = ' ')
                print(' & ', end = '')
        else:
            print(stu[lst3.index(maxScore)], end = '\n')
    print('.1f '%(maxScore), end = '')
    print('Points Progress', end = '\n')

def top(a):
    maxAvg = 0
    maxStu = ()
    for i in a:
        if int(i[3]) > maxAvg:
            maxAvg = int(i[3])
            maxStu = i
    return maxStu

for i in (s1,s2,s3,s4,s5):
    print('Name: ', end='')
    print(i.Name)
    print('Gender: ',end='')
    print(i.Gender)
    print('Grades: ', end='')
    print(i.Grades)
    print('Avg: ',end='')
    print(i.avg())
    print('Fail Number: ',end = '')#少了空格，您寫成'Fail Number:'
    print(i.fcount(), end = '')
    print('\n')

k = (s1.info(),s2.info(),s3.info(),s4.info(),s5.info())
print('Top Student:',end='\n')#多了空格，您寫成'Top Student: '
print('Name: ',end='')
print(top(k)[0])
print('Gender: ',end='')
print(top(k)[1])
print('Grades: ',end='')
print(top(k)[2])
print('Avg: ',end = '')
print(top(k)[3])
print('Fail Number: ',end=' ')
print(top(k)[4],end=' ')
print('\n')
```

類型	描述	
%%	在字串中顯示%。	>>>
%d	以10進位整數方式輸出。	
%f	將浮點數以10進位方式輸出。	
%e,%E	將浮點數以10進位及科學記號輸出。	
%o	以8進位整數方式輸出。	
%x,%X	以16進位整數方式輸出。	
%s	使用str()將字串輸出 print(chr(65)) => 'A'	
%c	以chr方式輸出。	
%r	使用repr()輸出字串。	

e.g.

- print('%.2f' %(0.0232222))
 - 小數點後兩位輸出
- print('%.3s'%'iverson')
 - 取字串前3位元輸出
- print('%7.4s'%'iverson')
 - 取字串前4位元輸出，總長為7，補空白在左
- print('%4.7s'%'iverson')
 - 取字串的前7位元輸出，總長為4，補空白在左

```

03 print(s)      # 直接印出
04 print(s[2:])  # 設定為[2:]，代表從index = 2 的字元到最後一個字元
05 print(s[0:7] + 'The Beautiful' + s[6:12])
                  # 利用+ 運算子將多個字串連接起來
07 print(s[::-1]) # 設定為[::-1]，表示從第0 個字元到最後一個字元，以1 為間隔取值
08 print(s[:-2])  # 設定為[:-2]，表示從第0 個字元到最後一個字元，以2 為間隔取值
09 print(s[::-1]) # 設定為[::-1]，會將整個字串顛倒輸出
10 print(s[-1:0:-2]) # 設定為[-1:0:-2]，從-1 開始往回到0，以-2 間隔取值

```

- the result:

```

Hello, World
llo, World
Hello, The Beautiful World
Hello, World
Hlo ol
dlroW ,olleH
drW,le

```

slice

- e,g,

Ex:

```

>>> toast="PYTHONSlice"
>>> toast[-5]
'S'
... -3 -2 -1

```

```
>>> toast[-5:]
```

```
'SLICE'
```

```
>>> toast[:6]
```

```
'PYTHON'
```

```
>>> toast[:6]+toast[-5:]
```

```
'PYTHONSlice'
```

Ex:

```

>>> 'PYTHONSlice'[::6]==toast[:6]
True

```

型態轉換

- when your data type is not the one you would like to use
- 利用轉型函數將資訊轉換成特定的格式

函數	描述
int(x[base])	將x轉換成整數，base是轉換的基數，欲設為十進位。
float(x)	將x轉換成浮點數。
str(x)	將x轉換為字串。
tuple(s)	將s轉換為tuple。
list(s)	將s轉換為list。
chr(x)	將x轉換為chr。

- 內建 str() 可以將任何資料轉換成string

- e.g. print("Hello" + str(2))

str.index(sub[, start[, end]]) 回傳 sub 第一次出現的索引值，若sub不在其中則回傳

ValueError。start值與end值定義尋找的範圍。

str.replace(old, new[, count]) 將 str 中的old 子字串以 new子字串代換。

如果有給定count值，則只有被給定的count值所代表的old子字串會被替換掉。

- [] 可加可不加

join & split

- join 可將list 中許多字串轉換為單一字符串
- split依照分割符號切割為許多子字符串 並存入一個list中
- 可搭配合作join& split
 - join必須加空白或者是字符串

稍微複雜的字串處理方式

`str.split([sep=None, maxsplit=-1])` 以`sep`分割成子字串，回傳儲存子字串的`list`。

`maxsplit`為子字串**最多**的數量。

- **sep = 分割器**
 - 若無定義分割器則以空白字元當作分割器，在回傳時把所有空白字元刪除
 - 這三個相同：
 - `print(string.split(sep = None))`
 - `print(string.split(sep = ''))`
 - `print(string.split())`
 - **maxsplit**是最多切割幾次，預設為-1代表切無限次，直到結束
- **str.splitlines**
 - 以`\n & \r`作為切割依據
- **str.partition(sep)**
 - 只分割第一個符合`sep`引述的字元，行程`tuple`
- **partition**
 - 將字串做分割，回傳第一個切割部分，切割依據& 第二個依據

`str.count(sub[, start[, end]])`

計算 `sub` 出現的次數並以`int`型態回傳，

`start` 為起始計算索引值，`end` 為結束索引值

- `sub`是要搜尋的字，一定要給值
- `start`從給定的`index`開始計數

`str.find(sub[, start[,end]])`

回傳 `sub` 第一次出現的索引值，若`sub`不在其中

則回傳-1。`start`值與`end`值定義尋找的範圍

函數	描述
<code>list.clear()</code>	將整個 <code>list</code> 清空。
<code>list.index(value, [start, [stop]])</code>	查詢 <code>list</code> 當中 第一個 <code>value</code> 出現的 <code>index</code> 值並回傳。
<code>list.count(value)</code>	計算 <code>list</code> 中 <code>value</code> 出現的 次數 。
<code>list.sort()</code>	將 <code>list</code> 裡面的元素做 排序 並 儲存 。
<code>list.reverse()</code>	將 <code>list</code> 裡面的元素 順序顛倒 並 儲存 。
<code>list.copy()</code>	將 <code>list</code> 直接 複製 一份。

- **entend**
 - 只接受有序的物件
 - **remove**
 - 如果找到多個則移除第一個
 - **sort**
 - 不需要傳入參數
 - `sort()` 會將`list`所有元素排序
 - `list`中元素若為不同型態則無法比較
 - ◦ 如果`sort`的是字串，則比較字串第一個字母的`unicode`碼大小
 - ◦ 若是`list`或`tuple`,比較`index = 0`的元素 (作為比較的元素要是同型態)
 - ◦ **list.sort(key=int)** -> 可將`list`中的數字 按照大小排序
- **list**可以用 `max()`
 - e.g. `max(list01)`

方法	描述
str.rpartition(sep)	以 sep 從最右端分割 str 為三個部份，結果回傳具有三個子字串的序對
str.rsplit([sep[, maxsplit]])	將 str 從最右端以 sep 分割成子字串，回傳儲存子字串的串列，maxsplit 為子字串最多的數量
str.rstrip([chars])	從 str 的最右端中移除 chars 字元，預設為空白字元
str.split([sep[, maxsplit]])	將 str 以 sep 分割成子字串，回傳儲存子字串的串列，maxsplit 為子字串最多的數量

- %r
 - 是個比較特殊的格式化
 - 使用repr() 這個函數來輸出，回傳物件的字串表達形式，不管後面接是什麼，都是以字串來表示他
 - a = '{} + {} = {}'
 - print(a)
 - print(a.format(7, 9, 7+9)) 以7, 9, 7+9 來替換掉三組{}

```

■ b = '{city}, {country}'
print(b)
print(b.format(city = 'Taiwan', country = 'Taiwan'))
  □ 以taipei替換掉city, 以taiwan 替換掉country

■ c = '{1} + {2} = {0}'
print(c)
print(c.format(3+4, 3, 4)) 以 3+4, 3, 4 依序替換掉{0} {1} {2}

```

- 補0
 - e.g. print('會員編號 : %(#)08d'%('#': 123456))

- 索引
 - 輸出格式也可以是字典索引
 - name = {"game": "xbox", "apple": "iphone", "camera": "nikon"}
 - print("%(apple)s, %(camera)s, \$%(game)s" %name)

- 旗標 flag
 - e.g.
 - print("會員編號1: %d, 會員編號2:%d" %(10,20))
 - print('會員編號2: %(#)d, 會員編號1: %(#)d'%'#1:10, '#2:20))
 - print('會員編號1: %(num1)d, 會員編號2: %(num2)d'%'num1':10, 'num2': 20))
 - d1 = {a:1, b:2, 789:3} 等同於 {123:1, 456:2, 789:3}

元組型態的slice

- e.g.
 - toast = "PYTHONSlice"
 - tuples = toast [0:3], toast [3:6], toast [6:9], toast [9:11]
 - tuples =
 - ('PYT', 'HON', 'SLI', 'CE')
 - ◆ tuples[0] >>> "PYT"
 - ◆ tuples [2:4] >>> ('SLI', 'CE')
 - ◆ tuples[1] [0] >>> 'H'

字典型態的slice

```
days={1:"Sunday",2:"Monday",3:"Tuesday",4:"Wednesday",5:"Thursday",6:"Friday",7:"Saturday"}
```

- ▶ ○ >>> days[2],days[3],days[4]


```
('Monday', 'Tuesday', 'Wednesday')
```

數字型態無法slice, 但可用數學方法取出or先用str()轉換成字串

str.splitlines([keepends])

str.startswith(prefix[, start[, end]])

str.strip([chars])

str.swapcase()

str.title()

str.translate(map)

str.upper()

str.zfill(width)

將 str 以新行符號分割成子字串，回傳儲存子字串的串列

判斷 str 是否以 prefix 開頭

從 str 中移除 chars 字元，預設為空白字元

將 str 中的英文字母進行大小寫轉換

將 str 轉換成作為標題的字串

將 str 中的字元以 map 中配對的字元轉換

將 str 的英文字母都改成大寫

回傳以 o 填滿 width 的新字串

容器處理

序對 tuple 的方法 ()

- 只有兩個: index & count

串列 list 的方法 []

- 與tuple很相似，但list可以增加、修改，tuple不行
- tuple = 客製化的模板，建立好就不能再更動
- **回傳值的概念**
 - 如果一個方法有回傳值，可以在該方法後面繼續加'!'加上其他方法
 - 但若沒有回傳值e.g. sort, append則不行，且無法直接print出這些方法的結果
 - 有回傳值的通常不太會直接改變到容器本身
- **空間**
 - 若寫 lst2 = lst 3 這兩個共用同一個空間，對任何一個做改變都會蓋到另外一個
 - 此情況下可用copy

函數	描述
list.append(object)	在list的最後面加入object。
list.extend(l)	將另一個具有順序的物件l附加在此list的最後面。
list.insert(index, object)	將元素object插入在index值為i的位置。
list.remove(value)	將list裡面的第一個符合value之元素刪除。
list.pop(i)	將list裡面index值為i的元素刪除並回傳。 如果沒有給定i值，則刪除並回傳最後一個。

dictionary字典的方法 {}

- key-value對應的型態

計算	描述
d[x]	從d中取得x所對應的值
d[x]=y	將d中x所對應的值指定為y 若d中沒有x這個key則新增一組
del d[x]	刪除d中x所屬的組合
x in d	判斷x是否在d的key值中
x not in d	判斷x是否不在d的key值中
iter(d)	回傳由d的key值所建立的迭代器
len(d)	回傳d的資料組數

- set 是無序的 {}

- 操作同樣程式多次結果可能順序不同
- 但內容物應該都相同

- remove() & discard() 是一樣的

- 但當所傳入參數不在set裡面
 - remove 會回傳 KeyError
 - discard 不會做任何動作

- how to set a set ? **設定一個set**

- e.g. **set01 = set()**
- 因為他跟dictionary 共用小括號，但他比較少用

set 集合的方法 {}

函數
set.pop()
set.add(e)
set.remove(e)
set.clear()
set.copy()
set.discard(e)
set.difference(set2)
set.difference_update(set2)

函數
set.intersection(set2)
set.intersection_update(set2)
set.union(set2)
set.update(set2)
set.issubset(set2)
set.issuperset(set2)

功能
將set的隨意一個元素回傳給呼叫的程式碼，之後將此元素刪除。
將元素e加入set。
將元素e從set中移除。
將set的所有元素清空。
回傳一份複製的set。
將e從set中捨棄。
將兩個set作差集運算，並回傳一個新的set。
將兩個set作差集運算，並將結果更新到set。

功能
將兩個set作交集運算，並回傳一個新的set。
將兩個set作交集運算，並將結果更新到set。
將兩個set作聯集運算，並回傳一個新的set。
將兩個set作聯集運算，並將結果更新到set。
判斷set是否為set2的子集合，回傳bool值。
判斷set2是否為set的子集合，回傳bool值。

不定個數的參數

Multi_arg.py

```

1 def function_name(*arguments, **keywords):
2     #dosomething
3     return something

```

- 加上一個星號，會組成tuple
- 加上兩個星號，會組成dictionary
- e.g.
 - o tuple


```
def hello(*names):
    for n in names:
        print("Hello, %s."%n)
hello("Tom", "Peter", "Bob", "Rain")
```
 - o dictionary


```
def hello(**names):
    for n in names:
        print("Hello",n,end=', ')
    print("you are",names[n],","
hello(John=25, Tom=20, Bob=33)
```

就像宣告變數一般，不得為數字

函式庫舉例 - Time

time.time()

- 取得系統時間

time.sleep(num)

- 設定暫停時間

time.localtime()

- 回傳格式：

- time.struct_time(tm_year, tm_mon, tm_mday, tm_hour, tm_min, tm_sec, tm_wday, tm_yday_, tm_isdst)

time.strftime(TimeFormat)

- 格式化輸出時間

time.gmtime()

- 取得 UTC 世界標準時間

- time.time() 為 1970年到現在經過的秒數
 - 可以在程式中設計t1, t2，並把秒數相減，得到程式運作的時間長度

▪ sys.version

- 回傳目前安裝在系統上的Python版本
- 格式：'(#build_number, build_date, build_time)[compiler]'

▪ sys.api_version

- 回傳Python直譯器的C API版本

▪ sys.version_info

- 回傳一個tuple型態的值
- ('主要版本', '次要版本', '小版本')

▪ sys.winver

- 回傳的版本數字是註冊在Windows裡的Python版本

▪ sys.path

- 定義Python搜尋模組的路徑

random 模組

隨機整數：

```
>>> random.randint(0,100)
```

隨機選取0到100間的偶數：

```
>>> random.randrange(0, 100, 2)
```

隨機浮點數：

```
>>> random.random()
```

```
>>> random.uniform(1, 10)
```

隨機字元：

```
>>> random.choice('abcdefghijklmnopqrstuvwxyz')
```

多個字元中選取特定數量的字元：

```
>>> random.sample('abcdefghijklmnopqrstuvwxyz', 3)
```

隨機選取字串：

```
>>> random.choice(['apple', 'pear', 'peach', 'orange', 'lemon'])
```

洗牌：

```
>>> items = [1, 2, 3, 4, 5, 6]
```

```
>>> random.shuffle(items)
```

sys 模組

▪ sys.argv[0]

- 會回傳此程式檔案的位置與名稱
- 數字代入1以上則是檔案傳入參數

▪ len(sys.argv)>1

- 是判斷是否有帶入參數

▪ sys.builtin_module_names

- 回傳Python程式語言內所有內置模組名稱

▪ sys.modules.keys()

- 得知目前已經載入的模組

▪ sys.platform

- 取得目前作業系統的版本

▪ sys.exit()

- 宣告sys.exit(0)終止程式

os 模組

- remove只能移除檔案
- 刪除資料夾前的條件：目錄要是空的

os模組：顯示系統環境參數與指令功能函數

▪ os.rename(src, dst)

- 對檔案或目錄更換名稱
- src引數是原本的資料夾
- dst引數是修改後的資料夾名稱

▪ os.remove(path)

- 移除檔案
- path引數傳入檔案位置
- 不會移除資料夾，若要移除資料夾可用下一個指令

▪ os.removedirs(path) 或 os.rmdir(path)

- 移除空的資料夾

▪ os.listdir(path)

- 輸出path引數位置的目錄和檔案名稱

NTU CSIE

▪ os.path.getmtime(path)

- 取得檔案的修改日期

▪ os.path.getatime(pah)

- 取得檔案的存取日期

▪ os.path.isfile(path)

- 判斷傳入的path引數是否為檔案

▪ os.path.isdir(path)

- 判斷傳入的path引數是否為目錄

▪ os.path.exists(path)

- 判斷傳入的path引數目錄或檔案是否存在

▪ os.system("...") #os.system("paus

- 執行 Command line 命令

▪ os.chdir(path)與os.getcwd()

- os.chdir(path)函數是切換目錄到path引數位置
- os.getcwd()是顯示目前所在的目錄位置

▪ os.mkdir(path[, mode])與os.rmdir(path)

- os.mkdir()是建立資料夾
- path引數是建立/刪除目錄的位置
- mode引數是Unix平台使用的
- os.rmdir()函數是刪除目錄

▪ os.path.getsize(path)

- 取得檔案大小

▪ os.path.getctime(path)

- 取得檔案的建立日期

使用 shutil 模組 - High-level file operations

比較高階的應用層，提供數個針對檔案操作的功能

▪ shutil.copytree(src, dst)

- 複製整個目錄，包含目錄內的所有檔案

▪ shutil.copy(src, dst)

- 複製檔案

▪ shutil.rmtree(path)

- 移除整個目錄，包含目錄內的所有檔案

▪ shutil.move(src, dst)

- 移動檔案，移動時也可以進行更換檔案名稱

▪ shutil.copystat(src, dst)

- 複製檔案，會連同檔案屬性一同複製

SIE

```
# for loop

for i in ('Nini', 2, 4, 5):
    print(i)          # 看後面容器裡有幾個東西就幾次
else:
    print('執行完畢')  # 當執行完畢的動作
print('執行完畢++++')  # 目的：如果程式不正常終止++++就不會被印出來

list(range(0, 10, 2))

# (0,10,2) = 起始值，終止值，變動方向，
# 在python內，只要有範圍都包含起始值，但不包含終止值本身
```

```

name = '帳單之門'
age = 99
gen = 'female'

print("姓名：%s\n年齡：%d\n性別：%s" %(name, age, gen))

# 兩數相加並且依照格式輸出
n1= eval(input())
n2= eval(input())
print("%.2f+ %.2f = %.2f"%(n1, n2, n1+n2))

print('會員編號：%#08d'%'#': 123456)
print('%.2f' %(123.456))
money = 987.98
print('$%.2f' % (7, money))

name = {"game": "xbox", "apple": "iphone", "camera": "nikon"}
print("%(apple)s, %(camera)s, %(game)s" %name)

print("會員編號1: %d, 會員編號2:%d" %(10,20))
print('會員編號2: %(#2)d, 會員編號1: %(#1)d' %{'#1':10, '#2':20})
print('會員編號1: %(num1)d, 會員編號2: %(num2)d' %{'num1':10, 'num2':20})

# 數字型態必須先用str()換成字符串
a = 1
b = 2
print('會員編號1: %(1)d, 會員編號2: %(2)d' %{str(a):10, str(b):20})

# 九九乘法表5
m = eval(input())
for a in range (1, m+1):
    for k in range (1, m+1):
        print(k, '*', a, '=', k * a, end = '\t')
    print()

# 時間模組
import time

def GetTime(DateFormat):
    time_stamp = time.time()
    return time.strftime(DateFormat, time.localtime(time_stamp))

def GetTime2(fmt):
    return time.strftime(fmt)
st = GetTime2('%Y-%m-%d %H-%M-%S')
print(st)

# 樂透模擬器
from random import sample
from time import sleep

print(sample(range(1, 10+1), 10))

for i in range(3,(1-1), -1):
    print(i)
    sleep(1)
res = sample(range(1,(19+1)), 7)
print('樂透號碼:', res[:6])
print('特別號碼:', res[6])

# 印出每排 m 個星號
'''
m = eval(input())
for i in range(1, m+1):
    for j in range(m):
        print('#', end = '')
    print()
'''

# 印出星號第幾排有幾個星號
n = eval(input())
# n : 印幾列
for j in range (n):
    # j : 現在印第幾列
    # j = 0, 1, 2, 3, 4
    # j+1 = 1, 2, 3, 4, 5
    # 8: 現在這列印幾個*
    for i in range (j+1):
        # i : 現在印到第幾個*
        print ('*', end = '')
    print()

# 字串數數
s6 = 'A clear conscience laughs at false accusation'
for i in range(len(s6)):
    print("s6[%2d]: %s" %(i,s6[i]))
    # s6[%2d]: %s 例如 s6[0]: A
print(s6.count('a', 0, 36))

# 小練習1
s7 = 'Welcome to Python Word Game.'
print(s7.split())
s8 = "Few free fruit flies fly from flames"
print(s8.find('fly'))
# 找出 fly 第一次出現的位置

print(s8.replace('a', 'b').replace('c', 'd'))
# 可以合併不同的方法

# 另一個版本印出星號
'''
n = eval(input())
# n : 印幾列
for j in range (n-2):
    # j : 現在印第幾列
    # j = 0, 1, 2
    # 2j+3 = 3, 5, 7
    # 8: 現在這列印幾個*
    for i in range (2*j+3):
        # i : 現在印到第幾個*
        print ('*', end = '')
    print()
'''
```

```
# 小練習 - 1 乘到 n
def fac1(n):
    total = 1
    for i in range(1, n+1):
        total *= i
    return total
y = eval(input())
print(fac1(y))
```

進階版小練習 - 排列組合

```
def P(n,m):
    return int(fac1(n)/fac1(n-m))      # 轉成整數
a = eval(input())
b = eval(input())
print(P(a,b))

def C(n,m):
    return int(P(n,m)/fac1(m))
print(C(a,b))
```

預設參數 並且 合併上面兩題

```
def fac_sum(n):
    totalforfac = 1
    totalforsum = 0
    for i in range(1, n+1):
        totalforfac *= i
        totalforsum += i
    return totalforsum, totalforfac
m = eval(input())
print(fac_sum(m))
```

若要拆開可以這樣寫， 但這樣會重複呼叫一模一樣的程式

```
# print(fac_sum(m)[0])
# print(fac_sum(m)[1])
```

可以把計算好的存下來 再輸出 比較快

```
answer = fac_sum(m)
print(answer[0])
print(answer[1])
```

全域 區域 變數

```
def func1():
    # 若在函式內沒宣告過跟全域變數x相同的變數名稱，就可以將其讀取出來
    # print('In func1()', x = ', x')

    # 但是不能更改他的值，否則上一行會被認定是需告前使用區域變數
    # 除非將x強制設定成全域變數

    global x
    print('In func1(), x = ', x)
    x = 99
    print('In func1(), x = ', x)

def func2():
    # 在將全域變數x 讀取出來前使用 x = xxx的話便會認定x為區域變數
    x = 30
    print('In func2(), x = ', x)

x = 10
print('x = ', x)
func1()
func2()
print('x = ', x)
```

寫檔

```
lst = ['apple', 'banana', 'watermelon']
f1 = open('write01.txt', 'w')
f1.write('This is my file \n')
f1.write('這行不會換行')
f1.write('這是第三行\n')
f1.writelines(lst)
f1.write(','.join(lst) + '\n')
f1.close()
```

若 read 沒有 close 會無法刪除該檔案 因為python仍有控制權，或者 write 會無法寫入，因為 close 之後才會寫入檔案

```
# define a new function
def hi(n):          # n = 傳入參數
    for i in range(n):
        print('hi', i+1)
hi(3)              # 意思是 n = 3
print(hi)           # 印出 hi 的記憶體位置
print(hi(5))
print(hi(3))        # 為什麼會有None?
```

小練習 - 無回傳值

```
def f1(n,m):
    print('result = ', n*m)
f1(3,4)
f1(8,9)
```

小練習 - 有回傳值 要 print 出來

```
def f2(n,m):
    return n*m
print('ans = ', f2(3,4))
print('ans = ', f2(8,9))           # 一定要記得加上逗點
```

小練習 - 1 加到 n

```
def sum1(n):
    total = 0
    for i in range(1, n+1):
        total += i
    return total
k = eval(input())
print(sum1(k))
```

```

f1 = open ('file1.txt', 'r')
txtLst = f1.readlines()
f1.close()
for i in range(len(txtLst)):
    txtLst[i] = txtLst[i].strip()
    print(txtLst[i])

```

只要有 close 就可以重複使用 f1

stores_old.csv 練習

讀取檔案

```

f1 = open ('stores_old_copy.csv', 'r', encoding = 'big5')
txtLst2 = f1.readlines()
f1.close()

```

把每一筆資料的空格去掉變成一大塊 string, strip -> 把左右兩邊的字元刪除
for i in range(len(txtLst2)):

```

    txtLst2[i] = txtLst2[i].strip()
    print(txtLst2[i])

```

再把每一行變成一個單獨的 list

```

for i in range(len(txtLst2)):
    txtLst2[i] = txtLst2[i].split(',')
    print(txtLst2[i])

```

印出公館門市的資訊

```

for i in range(len(txtLst)):
    if '公館門市'in txtLst[i]:
        print(txtLst[i])
        break
    ...

```

把一模一樣的資料寫進一個新的 csv 檔

```

f2 = open ('stores_old_copy.csv', 'w')
for i in range(len(txtLst2)):
    f2.write(','.join(txtLst2[i]))
f2.close()

```

小練習 - 傳遞變數

```
lst = []
```

```

def find3rdbiggest(slst):
    slst.sort()
    return slst[-3]

```

while True:

```
    n = eval(input())
    if n == -1:
        break
    lst.append(n)

```

```

print(lst)
print(find3rdbiggest(lst.copy()))
print(lst)

# slst = lst.copy()

```

```

# sys
import sys
print(sys.argv)

if len(sys.argv) == 1:
    print('一個argument')

if len(sys.argv) == 1:
    print('一個argument, if 沒有括號也可以!')

# os 模組
import os
os.mkdir('files')
n = eval(input())

# 先檢查files 存不存在
if os.path.exists('files'):
    os.chdir('files')
    for i in os.listdir():
        os.rmdir(i)
    os.chdir('../')
    os.rmdir('files')

os.chdir('files')      # change directory
for i in range(1,(n+1)):
    os.mkdir('f'+ str(i))
os.listdir()
os.rename('f1','folder1')
pt = os.listdir()
pt.sort()
print(pt)
os.chdir('../')
print(os.getcwd())

# 刪除之前 目錄必須要是空的，所以再進去把東西刪掉

os.chdir('files')
for i in os.listdir():
    os.rmdir(i)
os.chdir('../')
os.rmdir('files')

```

```

class Pokemon:
    pct = 0 # 類別屬性
    # 每個pokemon心中知道世界上共有幾個pokemon

    def __init__(self, name, lv, hp, hpmax):
        self.Name = name
        self.Lv = lv    # 級級
        self.Hp = hp    # 血量
        self.HpMax = hpmax
        Pokemon.pct += 1

# 一開始在 define 中 如果不知道寫什麼可以 pass
# 例如 def Attack(self): pass

    def Attack(self, target):
        if target.Hp <= 0:
            print('對方', target.Name, '已失去攻擊力，不可攻擊')
            return
        if self.Hp <= 0:
            print('己方', self.Name, '已不具備攻擊力')
            return
        print(self.Name, '攻擊了', target.Name, self.Lv)
        target.Defense(self.Lv)
    def Defense(self, damage):
        self.Hp -= damage
        if self.Hp <= 0:
            self.Hp = 0
            print(self.Name, '已失去戰鬥力')
    def Cure(self):
        self.Hp = self.HpMax
        print(self.Name, '已恢復戰鬥力')
    def ShowInfo(self):
        print('Hello, this is a Pokemon data')
        print('Name:', self.Name)
        print('Lv:', self.Lv)
        print('Hp: %d/%d' %(self.Hp, self.HpMax) )

# 注意：格式化輸出 在 %(self.Hp, self.HpMax) 之前不用加上逗號

```

```

# 攻擊
p1.Attack(p2)
p1.Attack(p2)
p1.Attack(p2)
p2.Attack(p1)
p2.Cure()
p2.Attack(p1)

p1.ShowInfo()
p2.ShowInfo()

```

```

# help(Pokemon)
print(dir(Pokemon))

p1 = Pokemon('皮卡丘', 10, 15, 15)
# p1就是物件的實體

print(type(Pokemon))
print(type(p1))
print(p1.pct)
print(p1.Name)
p1.ShowInfo()

p2 = Pokemon('小火龍', 15, 20, 20)
p2.ShowInfo()

print(p1.pct)
print(p2.pct)
print(Pokemon.pct)
# 因為世界上有兩隻pokemon所以顯示的都是2，大家共有的類別屬性

p1.pct = 0
# 建立一個實體屬性 p1.pct 並不會影響到其他的，只是他自己所擁有的，如此一來 p1 叫不到其他的pct

print(p1.pct)
print(p2.pct)
print(Pokemon.pct)

del p1.pct
print(p1.pct)

# 若類別屬性 & 實體屬性識別字相同，實體物件只能存取實體屬性：若在 init 裡面定義 self.pct = 77，那麼 p1, p2 pct 都會是 77，只有 Pokemon.pct 會是 2

# 但若把個別的 pct 刪除，例如 del p1.pct，那麼 p1.pct 又會回到 2

```