

範例5_2a 請寫一個程式，可以由電腦產生小寫字元5次，每次由使用者鍵入此字元，電腦評判是否正確，統計正確與錯誤題數。

運算思維

前面已經說明如何產生亂數，如何輸入字元，如何比對字元是否相同，現在要重複5次，只要使用for 迴圈，程式如下：

程式列印

```
import random
r=0
w=0
for i in range(5):
    a=random.randint(ord('a'),ord('z'))#ord 傳回對應整數
    print(chr(a))
    b=input()
    if b==chr(a):
        r=r+1
        print("Right")
    else:
        w=w+1
        print("Wrong")
print("Right=%d" % r)
print("Wrong=%d" % w)
```

自我練習

1. 請使用 Debug 工具，觀察以上程式執行流程。
2. 請產生6個-3~3的亂數，統計正數、0、負數的個數。
3. 請產生10個1~6的亂數，統計偶數與奇數個數。
4. 請產生10組(x,y)座標亂數，(x,y)都介於-2與2的整數，並統計落在原

點、x 軸、y 軸、四個象限的個數。

- 5.請產生10個1~6的亂數，統計數字「1」出現的次數。
- 6.請產生10個1~6的亂數，統計所有數字出現的次數。
- 7.請產生10個1~6的亂數，統計哪一個數字出現的次數最多。
- 8.心算練習。請連續產生8題兩個1位數，由使用者輸入相加結果，電腦回應對或錯，並統計正確與錯誤的題數。
- 9.音感練習。請連續發出10個 Do 到高音 Do，由使用者輸入1~8，電腦回應對或錯，並統計正確與錯誤的題數。
- 10.請由電腦自動產生30個大於等於0且小於等於100的亂數 x，統計0~59,60~69,70~79,80~89,90~100的個數。
- 11.請寫一個程式，電腦可以連續出現50個小寫字元，電腦統計出現母音(a,e,i,o,u)的個數。

範例5_3b 編譯器的乘法運算

演算法

我們人類的乘法運算式是先背誦九九乘法表，然後用以下直式乘法計算兩數相乘。

$$\begin{array}{r} 8 2 \\ \times 3 6 \\ \hline 4 9 2 \\ 2 4 6 \\ \hline 2 9 5 2 \end{array}$$

但是電腦可不用如此麻煩，電腦的強項是使用循序法，循序逐一累加計算。

(電腦內部只有累加器與比較器)，所以就用for迴圈實現循序累加如下：

a=82

```
b=36

s=0

for i in range(b):

    s=s+82

print(s)
```

範例 5_3a 求解輸入數值是否質數。

運算思維

任一整數，除了 1 和本身外，若沒有任何數可以整除此數，則稱此數為質數。所以，本例也可使用循序法 for，使用 2 至該數減 1 的數一一試除，若無一數可整除，則稱此數為質數。其次，為了提高程式執行效率，若找到 1 個可以整除，那此數就不是質數，可以提早離開迴圈，這樣才不會浪費時間，可提高執行效率，因為許多考試的程式執行時間也要算分數。

程式列印

```
i=12
b=True
print(i)
for j in range(2,i):
    if i %j ==0 :
        b=False
        break
if b==True:
    print("prime")
else:
    print("not prime")
```

範例 5_6a 等差數列的求和。例如：寫一個程式計算 $1+2+3+\dots+100$ 的和，或計算 $3+6+9+\dots+93$ 的和。

運算思維

本例人類通常使用等差級數求和的公式（首項+末項）*項數/2，但電腦有快速累加能力，所以不用如此麻煩，不用任何學習。電腦就直接一個一個累加，程式如下，此稱為循序法。前面範例5_3b 的乘法計算，我們

也是使用循序法。

```
s=0
for i in range(100+1):
    s=s+i
print(s)
```

自我練習

- 1、請寫一個程式計算 $3+6+9+\dots+93$ 的和。
- 2、請寫一個程式計算 $1+1/2+1/3+\dots+1/20$ 的和。
- 3、 $3+6+12+24+\dots+3072$ 的和。
- 4、 $5+5/2+5/4+\dots+5/1024$ 的和。

循序猜值法

所謂循序猜值法，就是將所有可能的解一一循序代入，又稱為暴力猜值法。例如，您要求任一數的開平方，因為開平方的結果一定在0與此數之間，那我們就從0開始，每次遞增1、0.1、0.01或0.001...，至於是要遞增多少，那就依您要的精密度了。例如，要求整數解，就遞增1，要求到小數1位，就遞增0.1，要求到小數兩位，就遞增0.01等等等，請看以下範例說明。

範例 5_6b 請用循序猜值法求任意數的平方根。

運算思維

- 1、人類求開根號的方法，大致如下：(錯)

		3 7 2
1	法則 3	1 3 8 3 8 4 9
2	$3 \times 20 + 7 = 67$ $67 \times 7 = 469$	3 8 4 4 6 9
3	$37 \times 20 + 2 = 742$ $742 \times 2 = 1484$	1 4 8 4 1 4 8 4
4		0

這樣可以減少計算，但是電腦計算能力強，所以可使用循序猜值法或二分猜值法求解，此即為電腦的運算思維。(二分猜值法請看範例 5_6d)

【程式列印】

1、以下程式每次遞增 1。

```
a=9
for i in range(9):
    if i*i>=a:
        print(i)#3
        break
```

2、以下程式每次遞增 0.1。因為 range 的遞增值只能整數，不能是浮點實數 0.1，所以我把他先放大 10 倍，求結果時再除以 10。

```
a=9
s=10#放大倍數
for i in range(a*s):
    if (i/s)*(i/s)>=a:
        print(i/s)#3.0
        break
```

3、同理，以下程式每次遞增 0.01。

```
a=9
s=100#放大倍數
for i in range(a*s):
    if (i*i/(s**2))>=(a):
        print(i/s)
        break
```

自我練習

1、請用循序法求解某一正數的立方根。例如，輸入 27 可得到 3.0。本例小數點取 1 位。

2、請用循序法求解兩數相除的結果。例如，輸入 27 與 9 可得到 3.0。本例小

數點取 1 位。(本例假設除數是大於 1 的整數)

3、假設有一函式 $y=f(x)=x^2-4x-5$ 請分別印出 x 從 -10 到 10 的值。

4、同上題，請用循序猜值找出其整數解。提示： x 從 -10 到 10 一一代入，找出使函數為零的值，此即為暴力猜值法解題。

5、同上題，請找出極小值。

6、函數極值。請寫一程式，可以輸入一個一元二次函式，並求其極大或極小值。例如，輸入 $y=f(x)=x^2-2x+2$ 有極小值 1，輸入 $y=f(x)=-x^2-2x+2$ 有極大值 -1，

7、假設一個一元多次方程式含有實數解，請寫一程式，可求其解。例如， $y=f(x)=x^2+x-0.75=0$ 的解是 0.5 和 -1.5。提示：浮點運算時，若無法得到 0，此時要使用接近 0 的判斷。例如， $\text{abs}(y)<0.0001$ ，即可視為成立，0.0001 即是其精密度，請換成自己想要的精密度 0.1、0.01 或 0.001。但是 Python 竟然無此問題，請鍵入以下程式，並觀察結果。

```
s=10#放大倍數
for x in range(-10*s,10*s):
    y=((x/s)**2)+x/s-0.75
    if y==0:
        print (x/s)
```

8、請用循序法求解任意整數的所有因數。

範例 5_6c

請寫一個程式，找出 1 至 200 的整數中，找出含有 2 的數字，且統計共含有多少個 2。例如，122 有兩個 2。

演算法

這是高中排列組合的問題，解題要先分析 2 出現的位置，但電腦就神了，就通通列出來，再將這些數字分解、並計算 2 的個數，數字如何分解，請複習 3_3 節。

程式列印

```
a=300
s=0
for i in range(1,a+1):
    a1=i//100 #百位數
```

```

a2=(i-a1*100)//10 #十位數
a3=i %10 #個位數
if a1==2:
    s=s+1
if a2==2:
    s=s+1
if a3==2:
    s=s+1
print(s)

```

while

上一節的 for 是用於程式設計階段已知迴圈次數，但有些情況，我們於程式設計階段並不知迴圈的執行次數，一定要等到程式執行後，才能知道何時離開迴圈，此時即可使用條件迴圈 while 指令。while 的迴圈語法如下：

```

while(條件運算式):
    程式區塊
    [break]
    [continue]
    [pass]

```

以上語法說明如下：

- 1.同for迴圈，請留意冒號與縮排。
- 2.當條件運算式值為 (True) 時，繼續執行迴圈，運算式為 (False)時，離開迴圈。
- 3.while 程式區塊內亦適用 break 與 continue，前者為強迫提早離開迴圈，後者可略過部份指令，提早進入條件運算式。
- 4.若是空迴圈，也要放一個 pass。

範例 5-4a

假如沒有除法運算子，請自行使用加減法，完成除法運算。

演算法則

兩數相乘時，程式設計階段就知道迴圈執行次數，所以使用 for。但除法就是不知道，只能說，當被除數大於除數時，就連續減去除數，能減去的次數，就是商，剩下的就是餘數。以 8 除以 3 為例，8 可以連續減 3 兩次，所以商就是 2，剩下的就是餘數，本例餘數就是 2。

【程式列印】

```
a=8#被除數
b=3#除數
q=0#商
while(a>=b):#只要(被除數>除數) 就執行迴圈
    a=a-b    # (被除數)-(除數)
    q=q+1    #商每次遞增 1
print(q)# 商數
print(a)#餘數
```

【程式說明】

1. a=被除數。
2. b=除數。
3. 餘數 q=0。
4. 所謂的商就是被除數 a 共有幾個除數 b，也就是只要 a 大於等於 b，就要執行以下指令：

a=a-b

q=q+1

5. 本例以 8 除以 3，實際演練如下：

(1) a=8；b=3

a>=b 成立，所以執行迴圈指令

a=a-b=5

q=q+1=1

(2) a=5；b=3

a>=b 成立，所以執行迴圈指令

a=a-b=2

q=q+1=2

(3) a=2；b=3

a>=b 不成立，所以離開迴圈

q=2 (商)

a=2 (餘數)

自我練習

1. 請使用 Debug 工具，觀察程式執行流程。
2. 同範例，但可以求到小數點 1 位的實數除法。提示：將被除數先乘以 10，再運算，最後將商再除以 10。

範例 5_4b

請寫一個擲骰子程式，滿足以下條件：

- (1) 可以產生 1 至 6 的亂數。
- (2) 累加以上亂數。
- (3) 輸出此亂數與統計其和。
- (4) 若亂數不為 1，則重複 (1) ~ (3)，否則輸出其和並結束程式。

執行結果

```
2 3 3 2 3 5 2 4 6 1 30
```

程式列印

Python 並沒有後測試迴圈，所以有時候就要先執行一次，再進入迴圈，如以下程式。

```
import random
s=0
a=random.randint(1,6)
print(a,end=' ')
while a!=1:
    s=s+a;
    a=random.randint(1,6)
    print(a,end=' ')
print(s)
```

補充說明

這題也有人這樣出題『電腦一直產生 1 到 6 的亂數，直到產生 1 時停止』，所以 while 可以修改如下：

```
while not(a==1):
```

not 就想為『直到』這樣就很好理解。

自我練習

- 1、同上範例，但人和電腦玩，每人每次產生 1 個亂數，直到人贏為止，輸出共產生幾次亂數。
- 2、同上範例，但人和電腦玩，每人每次產生 1 個亂數，直到人贏 3 次為止，輸出共產生幾次亂數。
3. 請寫一個程式，滿足以下條件：(擲骰子遊戲)
 - (1) 可以產生兩個 1 至 6 的亂數。
 - (2) 累加以上亂數。
 - (3) 輸出此亂數與其和。
 - (4) 若亂數和大於 8，則重複 (1) ~ (3)，直到亂數和小於等於 8，則程式結束。
- 4、電腦心算測驗。請寫一程式，可以無限次數，顯示題號、出現兩個 1 位數，

並由使用者回答，直到答對 10 題，電腦顯示使用的時間。

5、同上題，但加上直到答錯才停止，並顯示連續答題數目。

6. 請寫一猜拳遊戲程式，可以讓人與電腦猜拳，並輸出結果，直到任一方贏 3 次為止。

二分猜值法

前面的循序法是循序一個一個猜，若沒猜中，每次僅減少一筆資料，這裡要介紹一個較有效率的猜值法，稱為二分猜值法。二分猜值法是每次猜其可能範圍的中間值，若猜的太大，則調整猜值上限為此中間值，表示後半部都刪除；若猜的太小，則調整猜值下限為此中間值，表示前半部都刪除，每次都縮小範圍為一半，所以可提升猜值效率，請看一下範例。

範例 5_6d 猜數字。請您先默想一個 1~100 的數字，讓電腦猜，但每次要回應太大『3』或猜中『2』或太小『1』。

執行結果

1、以下是我默想 40，逐次回答電腦猜值的過程。

```
電腦猜x=50
Please input 3(太大),2(猜中),1(太小):3
太大,電腦猜x=25
Please input 3(太大),2(猜中),1(太小):1
太小,電腦猜x=37
Please input 3(太大),2(猜中),1(太小):1
太小,電腦猜x=43
Please input 3(太大),2(猜中),1(太小):3
太大,電腦猜x=40
Please input 3(太大),2(猜中),1(太小):2
Bingo,The number is 40
```

演算法

- 1、讓電腦猜，電腦可從 1,2,3,4 一個一個開時猜，此稱為循序猜值法，若數字是 1，那運氣好，1 次就猜到；若數字是 100，那就要猜 100 次，所以平均是 50 次才可猜到，這就由讀者練習。
- 2、本範例要介紹二分猜值法，那就是每次猜其一半，例如，本例下限 $x_1=1$ ，上限 $x_2=100$ ，我就猜其中間值 $x=(x_1+x_2)/2=50$ ，然後您會告知猜中、太大或太小。
- 3、若是太大，則調整上限 $x_2=x-1=49$ ；若是太小，則調整下限 $x_1=x+1=51$ ，然後重複步驟 2 與 3，每次範圍縮小一半，很快就會猜到，程式如下：

程式列印

```
x1=1
x2=100
right=False
while not(right):
    x=(x1+x2)//2
    print('電腦猜 x=%d'%x,end='')
    a=input("Please input 3(太大),2(猜中),1(太小):")
    if a=='2':
        right=True
        break
    elif a=='3':
        print('太大,',end='')
        x2=x-1
    else:
        print('太小,',end='')
        x1=x+1
print("Bingo,The number is %d"% x)
```

範例 5_6e 請以二分猜值法求解一正數的平方根。本例要精密度到小數點以下第二位。

執行結果

```
1:x1=0.00,x2=9.00
2:x1=0.00,x2=4.50
3:x1=2.25,x2=4.50
4:x1=2.25,x2=3.38
5:x1=2.81,x2=3.38
6:x1=2.81,x2=3.09
7:x1=2.95,x2=3.09
8:x1=2.95,x2=3.02
9:x1=2.99,x2=3.02
10:x1=2.99,x2=3.01
2.9970703125
```

演算法

1. 本例以求9 的平方根為例。
2. 求解9 的平方根，其解必在0 到9 之間，所以設定下界為0，上界為9。首先，先猜4.5，如下圖步驟1，但4.5 的平方為20.25，大於9，表示猜的太大，那就縮小範圍，將上界調為4.5，持續在0 與4.5 之間猜值，如

下圖步驟2。第二次就猜2.25，但是2.25 的平方為5.025，小於9，表示猜的太小，那就調整下界為2.25，且持續在2.25 與4.5 之間猜，如下圖步驟3，那要猜到何時呢？答案是設定一個精密度，例如，您要小數一位，那就是下界與上界之間的距離大於0.1；若是要小數兩位，那就是下界與上界之間的距離大於0.01，通通要繼續猜；當離開迴圈時，此時下界或上界的值，就都是所要求的答案了。

猜值步驟	猜值內容		
1	x1 0	x 4.5 (太大)	x2 9
2	x1 0	x 2.25 (太小)	x2 4.5
3		x1 x 3.37 (太大)	x2

- (1) 設求解的正數為 y ，則其平方根必在 $x_1=0$ （下界）與 $x_2=y$ （上界）之間，猜值範圍為 $[x_1, x_2]$ 。
- (2) 首先猜 x_1+x_2 之和的一半 x 。
- (3) 若所猜 x 的平方小於 y ，表示猜的太小，並縮小猜值範圍為 $[x, x_2]$ 。
- (4) 若所猜的 x 的平方大於 y ，表示猜的太大，並縮小猜值範圍為 $[x_1, x]$ 。
- (5) 重覆步驟(2)、(3)、(4)，只要 $[x_1, x_2]$ 的範圍大於所要求的精密度（小數兩位0.01 或小數三位0.001），則要繼續猜；當離開迴圈時，此時的 x_1 或 x_2 即為平方根。

程式列印

```

y=9.0
x1=0.0
x2=y
n=1
while (abs(x1-x2)>0.01):#距離>0.01 繼續猜
    print("%d:x1=%2.2f,x2=%2.2f" %(n,x1,x2))
    x=(x1+x2)/2
    t=x*x

```

```
    if t<y :
        x1=x
    else:
        x2=x
    n=n+1
print(x)#2.99
```

自我練習

- 1、請以二分猜值法，求解兩數相除的結果。
- 2、請以二分猜值法，求解一正數的立方根。
- 3、請寫一程式，由電腦產生一個 1 到 100 的亂數，由使用者用二分猜值法猜，電腦應逐次回答太大、太小、或猜中，且回應幾次猜中。