

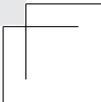
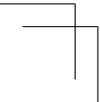
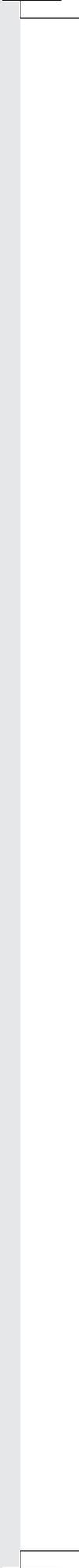
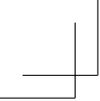
十二年國民基本教育
技術型高級中等學校商業與管理群

程式語言與設計

(下冊)

洪國勝、蔡懷文
蔡懷介、陳蘊慈 編著

泉勝出版有限公司
www.goodbooks.com.tw



編輯大意

- 一、本書係依據中華民國 107 年 8 月教育部發布之十二年國民基本教育技術型高級中等學校群科課程綱要 - 商業與管理群「程式語言與設計」編輯而成。
- 二、本書分上、下冊，供技術型高級中等學校商業與管理群第一學年第一學期及第二學期，每周上課二節二學分教學之用。
- 三、本書選擇 Python 作為程式語言與設計的工具。
- 四、因為學生的起點是國中畢業生，且技高學生 PR 值大多分佈在中後段，我們希望學生從作中學，所以很多地方的執行結果，我們不直接寫出，而是留給學生實作再填入結果，這樣印象才會深刻。例如：

```
a=5;b=4 #指派或稱賦值  
print(a/b) #實數除法，得到實數商_____  
print(a//b) #整數除法，得到整數商_____
```

- 五、本書教材範例豐富，所有指令都有範例解說，讓程式不只是程式，而是可以解決日常生活問題的工具。
- 六、充分的自我練習。我們有充分的自我練習，這樣學生才能充分練習，達到舉一反三的效果。
- 七、跨領域學習。本書的範例涵蓋數學、英文、資訊科技與生活科技，完全呼應與配合 108 課綱的跨領域教學與學習。
- 八、本書章節前加註 ※ 者，為進階主題，教師可依實際教學進度酌予刪減。
- 九、本書雖力求嚴謹，但疏漏之處在所難免，尚祈先進惠予指正。

⚙️ 本書學習表現如下：

1. 了解程式語言在生活上的應用與對科技創新的影響。
2. 具備撰寫程式語言基本能力，運用資訊科技方法解決問題。
3. 具備程式設計之邏輯思考能力，展現系統思考、分析與探索之素養。
4. 具備實作程式設計能力，展現程式語言跨域應用之軟實力。
5. 能思辨勞動法令規章與相關議題，省思自我的社會責任。

⚙️ 本書教學注意事項：

1. 本科目為技能領域實習科目。
2. 教學活動宜引用流程圖作為程式設計步驟的輔助說明。
3. 教學不宜偏重或強調單一軟體之功能，宜引導學生多認識與使用不同作業系統或雲端平台之相關應用軟體，增加學習之多元性與適應性。

目錄

下冊

▶ 第6章 串列程式設計

6-1	串列 (list).....	6-2
6-2	一維串列.....	6-2
6-3	二維串列.....	6-13
6-4	串列函式與串列生成式	6-29
6-5	串列程式設計實作演練	6-37
6-6	tuple.....	6-50
6-7	dict.....	6-51
6-8	set	6-55
6-9	本章內容摘要	6-62

▶ 第7章 函式應用

7-1	模組 (Module).....	7-2
7-2	內建函式庫的認識與應用	7-2
7-3	自定函式.....	7-18
7-4	全域變數與區域變數。	7-26
7-5	函式應用實例演練.....	7-27
7-6	本章內容摘要	7-33

▶ 第8章 綜合應用實例

8-1 字串的運算.....	8-2
8-2 時間的運算.....	8-20
8-3 本章內容摘要.....	8-32

▶ 附錄一：商科技藝競賽程式設計試題

▶ 附錄二：本書中英文詞彙索引對照表

串列程式設計

學習綱要

- ≡ 6-1 串列
- ≡ 6-2 一維串列
- ≡ 6-3 二維陣列
- ≡ 6-4 串列函式與串列生成式
- ≡ 6-5 串列程式設計實例演練
- ≡ ※6-6 tuple
- ≡ ※6-7 dict
- ≡ ※6-8 set
- ≡ 6-9 本章內容摘要

學習目標

- ≡ 1. 能瞭解串列程式設計。
- ≡ 2. 能瞭解與實作一爲串列。
- ≡ 3. 能瞭解與實作二維串列
- ≡ 4. 能瞭解與實作串列函式與串列生成式。
- ≡ 5. 能實作串列程式設計實例。
- ≡ 6. 能瞭解與實作 tuple 結構。
- ≡ 7. 能瞭解與實作 dict 結構。
- ≡ 8. 能瞭解與實作 set 結構。

6-1 串列 (List)

處理少量的資料，可以個別設定一些單一變數，但是如果資料龐大，例如，若有資料 3,8,4,7,2,9 要求極大、極小、平均等運算，若還是使用 6 個變數儲存，程式將會很冗長，本章將會介紹另一種資料結構『串列』，然後配合上一章的 for 或 while 迴圈，就可以以非常精簡的程式，處理以上大批資料。其次，串列還有維度之分，分別是一維、二維、三維等等，就要看您資料運算的方式，若只有同批資料的關係，例如，上例就是一維資料串列就可以，請看 6-2 節。但是，在一個班級學生裡，每個人都有國、英、數、自然、社會等 5 科成績，我們不僅要求縱向的各科平均，還要求橫向的每人平均，那就需要使用二維串列，請看 6-3 節。(補充說明：C/C++、Java 等連續資料的集合稱為 Array 翻譯為陣列，Python 則稱為 List，翻譯為串列。)

6-2 一維串列

⚙️ 一維串列宣告

建立一個連續數值的一維串列的方法很多種，最常見語法如下：

```
串列名稱=[初值 for i in range(串列長度)]
```

例如：

```
n=5  
a=[0 for i in range(n)]
```

則可使用 a[0],a[1],a[2],a[3],a[4] 等共 5 個記憶體位置，且其初值均為 0，以上 0~4 稱為串列索引，往後都是使用索引存取串列元素。又例如：

```
n=4  
b=[i for i in range(n)]  
print(b) #輸出串列 [0,1,2,3]
```

以上語法也可簡化如下：

```
串列名稱=[初值]*串列長度
```

例如：

```
a=[0]*5
```

也是同時指派 `a[0],a[1]..a[4]` 的初值為 0。

⚙️ 串列的宣告與預設初值

變數可以宣告的同時給予初值，串列也是。例如：

```
a=[3,2,7,6,8]
print(a)
```

即可宣告串列也給予初值。

⚙️ 串列值的存取

串列的每個元素都可使用索引存取其值。例如：

```
a=[0,1,2]
print(a[0]) #0 輸出串列索引0的值
a[0]=3 #修改串列索引0的值
b=a[0] #提取串列索引0的值
print(b)
```

串列索引可以是負值，索引負值代表從串列末端存取。例如：

```
a=[0,1,2,3]
print(a[-1]) #3
print(a[-2]) #2
```

⚙️ 串列的輸出

串列的輸出採用 `print()` 方法，以下三種方式均可輸出串列。方法一：使用 `print(串列名稱)`，請鍵入以下程式，寫出執行結果。

```
a=[1,2,3]
print(a)
```

方法二：使用索引值。請鍵入以下程式，寫出執行結果。

```
a=[1]*5
for i in range(len(a)): #len()是內儲函式，可得到串列a的長度，
                        #請看7-2節。
    print(a[i],end=" ")
```

方法三：針對 Iterable(可疊代的)物件(如串列、tuple、dict、set等，詳見 6-6 ~ 6-8 節)，Python 的 for-loop 提取每一元素的語法如下：

```
a=[1,2,3]
for a1 in a:
    print(a1)
```

以上 a1 可以是一個任意合法的變數名稱，例如 a1,aa,item 等。請鍵入以下程式，寫出執行結果。

```
a=[i for i in range(5)]
for a1 in a: #對於每一個a1 in a
    print(a1,end=' ') #輸出a1
```

⚙️ 串列的複製

串列的複製有兩種方式。第一種是取別名，兩者共用位址，也就是兩者互相連通，只要一個改變，另外一個也跟著改變。例如：

```
a=[3,2,7,6,8]
print(a)
b=a #串列的複製，共用位址
print(b)
a[4]=0
print(b)#[3,2,7,6,0]
b[0]=0
print(a)#[0,2,7,6,0]
```

第二種使用 `b=a[:]` 複製串列，此一複製彼此獨立，互不干擾。例如：

```
a=[3,2,7,6,8]
print(a)
b=a[:] #彼此獨立
print(b)
a[4]=0
print(b)#[3,2,7,6,8]
b[0]=0
print(b)#[0,2,7,6,8]
print(a)#[3,2,7,6,0]
```

⚙️ 子串列

從一個串列提取部分元素到另一串列，稱為提取子串列。提取子字串可使用範圍運算子 (`:`)，格式為「(開始索引:結束索引)」，但不含結束索引。若開始索引省略，則表示從頭提取；若結束索引省略，則表示提取到串列末端；索引同樣也是可以使用負值，請看以下程式。

```
a=[0,1,2,3,4]
b=a[1:3] #從索引1開始到索引3，不含索引3
print(b) #[1,2]
c=a[1:] #從索引1開始到結束
print(c) #[1,2,3,4]
d=a[:2] #從索引0開始到索引2，不含索引2
print(d) #[0,1]
e=a[1:-2] #從索引1開始到索引-2，不含索引-2
print(e) #[1,2]
f=a[:-2] #從索引0開始到索引-2，不含索引-2
print(f) #[0,1,2]
g=a[-3:] #從索引-3開始到結束
print(g) #[2,3,4]
```

範例6-2a

設有資料如下

77, 66, 99, 44, 55

請寫一個程式，可以計算平均、最高分及最低分。

👉 執行結果

```
Sum=341
Avg=68
Max=99
Min=44
```

👉 運算思維

1. 此題我們在第四章已經介紹，我們使用 5 個單一變數儲存以上資料，但是程式有點冗長。本例爲了能使用迴圈來簡化程式，資料結構採用一維串列儲存以上資料。

```
a = [ 7 7 , 6 6 , 9 9 , 4 4 , 5 5 ]
```

2. 有了串列，就有串列索引，有了串列索引，就可使用迴圈存取資料，參考程式如下：

```
a=[77, 66, 99, 44, 55] #以串列儲存資料
#先指派第0筆資料爲和、極大與極小值
sum=a[0] #和
max=a[0] #極大值
min=a[0] #極小值
#使用迴圈逐一從第1筆資料開始處理每一筆資料
for i in range(1,5):
    sum=sum+a[i] #累加每一筆資料
    if a[i]> max: #若該筆資料大於極大值
        max=a[i] #極大值以該筆資料取代
    if a[i]<min: #若該筆資料小於極小值
        min=a[i] #極小值以該筆資料取代
print("Sum=%d" % sum) #輸出和
print("Avg=%d" % (sum/5)) #輸出平均，(sum/5)括號不可省
print("Max=%d" % max) #輸出極大值
print("Min=%d" % min) #輸出極小值
```

👉 自我練習

1. 設有資料如下：
 - 1, 0, 3, 4, -5, 8
 - (1) 統計正數、0、負數的個數。
 - (2) 請問您如何儲存以上資料？

2. 假設有資料如下：

3,6,2,1,2,3,5,6,1

(1) 計算並輸出共有多少個 3。(2) 計算 1~6 各幾個。(3) 哪一個數字出現最多次

3. 假設有資料如下：

11,22,33,44,55,66,77,88,99,12,23,25,31,32,33,42,72,82,99

統計與輸出 0~9,10~19,20~29,30~39,40~49,50~59,60~69,70~79,80~89,90~100 等區間的人數。

範例6-2b

請寫一程式，可讀入 5 位學生的國文成績，並可由使用者輸入座號來查詢成績。

執行結果

```
1
input score:11
2
input score:22
3
input score:33
4
input score:44
5
input score:55
輸入完畢，可開始輸入座號查詢，輸入0結束
input number:2
座號：2 成績:22
```

運算思維

1. 本例資料使用陣列儲存資料，這樣才能使用迴圈，簡化程式。
2. 使用 `a=[0]*6` 宣告 a 陣列，則可使用 `a[0]..a[5]`，但索引 0 可以不用。
3. 輸入 1~5 號成績。
4. 進入查詢，且要能重複查詢，所以使用不定數量 while 迴圈。

👉 程式列印

```
n=5+1 #宣告學生人數，本例索引0不用，所以加1
a=[0]*n #宣告a[0]..a[5]，但本例索引0不用
#使用迴圈，逐一輸入1號到5號成績
for i in range(1,n):
    print(i,end=' ')
    b=int(input("input score:"))
    a[i]=b
print("輸入完畢，可開始輸入座號查詢，輸入0結束",end='')
#進入查詢，且要能重複查詢，且為不定數量，所以使用不定數量while迴圈
c=int(input("input number:")) #記得轉為int
while c!=0: #只要c!=0 重複迴圈
    print("座號:%d 成績:%d" %(c,a[c]),end='')
    #繼續輸入座號
    c=int(input("input number:")) #記得轉為int
```

👉 自我練習

1. 假設班上共有 12 人，請寫一程式，可以由使用者自行輸入座號，電腦並隨時輸出共有哪些人未到。(提示：可用串列的值表示到與未到，例如串列值 1 表到，0 表未到)

範例6-2c

英文打字練習。假設有單字如下：

look、at、one、two、tree、and、day、book、from、go

請寫一個程式，可以出現 10 題英文單字，讓使用者可鍵入此單字，並評判正確與錯誤題數和使用時間。

👉 執行結果 (以下僅摘取前面兩題)

```
look :
look
at :
at
```

👉 程式列印

1. 將單字用串列儲存如下：

```
import time #使用time.time()
a=['look','at','one','two','tree','and','day','book','from','go']
n=10
right=0
wrong=0
t1=int(time.time())#取得系統時間的秒數
#使用迴圈逐一輸出每一題單字
for i in range(n):
    print(a[i],':')#每次取一個輸出
    b=input()#由使用者鍵入此單字
    if b==a[i]:#比較是否相同
        right=right+1
    else:
        wrong=wrong+1
t2=int(time.time())
t=t2-t1
print('right=%d'% right)
print('wrong=%d'% wrong)
print('total time=%d second'% t)
```

👉 自我練習

1. 同範例 6-2c，但使用者答題後，馬上可回應正確或錯誤，若是錯誤也要回應正確的單字。
2. 同範例 6-2c，但增加為 100 個單字，每次測驗依亂數出現 10 個單字，且不能重複。亂數的取法請複習上冊 3-4 節。
3. 同範例 6-2c，但增加為 100 個單字，每次測驗 3 分鐘，題目依照亂數出現，統計使用者可答對題數。亂數的取法請複習上冊 3-4 節。
4. 同範例 6-2c，但增加為 100 個單字，題目依照亂數出現，使用者答錯了才停止，統計使用者可連續答對題數。亂數的取法請複習上冊 3-4 節。

⚙️ 查表

寫程式有時候會用到查表問題，例如，十進位轉為十六進位時，當除以 16 時，其餘數的對應值如表 6-1：

表 6-1 十六進位符號對照表

餘數	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
對應值	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

此時即可使用陣列。

範例6-2d

請寫一程式，可將任意十進位數轉為 N 進位數。(2≤N≤16)

👉 運算思維

1. 將 16 進位的 16 個符號以串列儲存如下：

```
c=["0","1","2","3","4","5","6","7","8","9","A","B","C","D","E","F"]
```

2. 只要待轉換數字大於 0，將待轉換數字除以欲轉換的基底，每得到 1 個餘數，就查表。例如，餘數是 10，那就得到『A』。
3. 向左串接餘數，即為所求。例如：待轉換數字若為 18，則運算過程如下：

```
d= '' (先指派一個變數作為轉換後的十六進位數)
18/16=1..2 (商是1，餘數p='2')
d=p+d (d= '2'，商大於0，程式繼續)
1/16=0..1 (商是0，餘數p='1')
d=p+d (d='12'，商沒有大於0，程式結束)
```

程式完成，18 的十六進位是 '12'。

👉 程式列印

```
#將十六進位的16個符號以串列儲存
c=["0","1","2","3","4","5","6","7","8","9","A","B","C","D","E","F"]
a=18 #待轉換數字
b=16 #轉換基底
d="" #轉換結果
#因為數字長度不定，所以使用不定while迴圈，逐一求餘數
while (a>0): #只要待轉換數字a還大於0，則重複迴圈
    r=a%b; #以基底為除數，求餘數
    p=c[r] #將餘數查表轉為字元
    d=p+d #向前串接所求得字元
    a=a//b #除以基底
print(d) #輸出結果 FF
```

👉 補充說明

1. 先產生的餘數是放在右邊，所以本例是 $d=p+d$ 向左串接，請自行修改為 $d=d+p$ ，並觀察執行結果。

👉 自我練習

1. 假設有一個大富翁遊戲，共有 10 個位置，每個位置的獎罰如表 6-2，遊戲一開始假設有 10000 元，請寫程式讓使用者玩，每次擲一顆骰子，決定其行走距離、落腳位置、增減金額，且記錄每次剩餘金額。

表 6-2 大富翁遊戲賞罰表

0	1	2	3	4	5	6	7	8	9
中發 票得 200元	闖紅 燈罰 200元	中樂 透得 10000 元	任意 停車 , 罰 1000 元	繳汽 車稅 2000 元	繳房 屋稅 3000 元	停玩 一次	得到 股息 1000 元	抽獎 得到 3000 元	繳停 車費 30元

2. 以串列重做上冊範例 4-5d 的電子琴。

運算思維提示：

- (1) 前面上冊 3-4 節已經介紹電腦如何發聲，例如：以下程式可發出電子琴「Do」的音。

```
import ctypes
p = ctypes.windll.kernel32
p.Beep(523,200)
```

- (2) 前面上冊範例 4-5d 也用選擇結構製作一台電子琴。例如：按鍵盤「1」鍵，電腦發出「Do」音；按鍵盤「2」發出「Re」音，以上功能的程式如下：

```
from tkinter import * #載入視窗模組，不是本書介紹範圍，請在此先體驗就好
import ctypes #載入發音模組
p = ctypes.windll.kernel32
def aa(e):
    k=e.char
    tk.title(k)
    if k=='1':
        p.Beep(523,200) #發Do音
    elif k=='2':
        p.Beep(587,200)
    elif k=='3':
        p.Beep(659,200)
    elif k=='4':
        p.Beep(698,200)
    elif k=='5':
        p.Beep(784,200)
    elif k=='6':
        p.Beep(880,200)
    elif k=='7':
        p.Beep(988,200)
    elif k=='8':
        p.Beep(1046,200)
tk=Tk() #產生視窗
tk.geometry("300x100+50+70") #視窗大小
enyl=Entry(tk) #產生輸入盒
enyl.bind('<Key>',aa) #綁定aa函式
enyl.pack() #版面配置
```

現在請用串列查表的方式，重做本範例。

3. 假設有兩筆資料如下：

```
3, 4, 7, 9, 12
1, 5, 6, 8, 15,
```

請先以兩個串列儲存，再合併成一個串列。

4. 假設有兩筆資料如下：（每筆資料都已經是由小而大排列）

```
3, 4, 7, 9, 12
1, 5, 6, 8, 15, 16, 17, 18
```

請以兩個串列儲存以上資料，再幫忙合併，且合併後還是由小而大。

- ※ 5. 假如有兩筆資料如下：

```
3, 1, 4, 5, 6, 7
5, 1, 7, 4, 3
```

- (1) 請問如何儲存以上資料？
- (2) 請比較與分析有哪些資料重複。
- (3) 將重複的資料放到第三個串列。（交集）
- (4) 將兩筆資料合併，放到第四個串列，但資料不可重複。（聯集）

- ※ 6. 假如有資料如下

```
3, 1, 4, 5, 6, 7, 1, 1, 3
```

請將重複的資料去掉。

6-3 二維串列

前面範例 6-2c 的單字

```
look \ at \ one \ two \ tree \ and \ day \ book \ from \ go
```

此稱為一維資料，我們使用一維串列儲存如下：

```
a=['look', 'at', 'one', 'two', 'tree', 'and', 'day', 'book', 'from', 'go']
```

這樣可以用迴圈存取每個單字，現在若有英文單字的詞性與中文資料如下：希望能出現中文，讓使用者輸入英文，並檢查正確與否？

編號	英文	詞性	中文
1	delighted	adj	高興的
2	gear	n	服裝
3	behave	v	表現

以上每一個單字都還有詞性與中文註解，此稱為二維資料，此時就需要使用二維串列儲存以上二維資料。本例可宣告二維串列如下：

```
a=[["delighted", "adj", "高興的"],
    ["gear", "n", "服裝"],
    ["behave", "v", "表現"]]
```

則以上單字、詞性、中文的索引編號（以下簡稱索引）如下：

英文	詞性	中文
delighted (0,0)	adj (0,1)	高興的 (0,2)
gear (1,0)	n (1,1)	服裝 (1,2)
behave (2,0)	v (2,1)	表現 (2,2)

因為是二維資料，此時就需要二維索引（列索引，行索引），例如，`a[0][2]` 的 0 稱為列索引，2 稱為行索引。有了索引，往後就可以使用變數名稱與索引存取以上資料。例如：

```
print(a[0][2]) # _____
print(a[2][1]) # _____
```

以上即為本節所要介紹的二維串列。

⚙️ 二維串列的宣告

二維串列的宣告常用語法如下：

```
串列名稱=[[初值 for i in range (行數)]for j in range (列數)]
```

例如：以下程式可快速初始化一個三行四列的二維串列，其初值為 0。

```
a=[[0 for i in range (3)]for j in range (4)]
```

以上程式亦可簡化如下：

```
b=[[0]*3 for i in range(4)]
print(a) # _____
print(b) # _____
```

⚙️ 二維串列初始化

單一變數、一維串列都可宣告變數同時給予初值，二維串列也可以，以下程式可初始化一個二維串列，且預設初值。

```
c=[[1,2,3],[4,5,6]]
print(c)
```

以上二維串列的索引如表 6-3。

表 6-3 二維串列資料與索引對照表

c[0][0]=1	c[0][1]=2	c[0][2]=3
c[1][0]=4	c[1][1]=5	c[1][2]=6

請鍵入以下程式，寫出執行結果。

```
c=[[1,2,3],[4,5,6]]
print(c)
c[0][1]=12
d=c[0][1]
print(d)
print(c)
```

範例6-3a

假設有單字的詞性與中文資料如下：請寫一個程式，此程式會連續 3 次出現中文與詞性，讓使用者輸入其英文，並統計答對與答錯題數及答題花費時間。

編號	英文	詞性	中文
1	delighted	adj	高興的
2	gear	n	服裝
3	behave	v	表現

輸出結果

```
高興的 adj :
delighted
服裝 n :
gear
表現 v :
beha
right=2
wrong=1
total time=23 second
```

運算思維

1. 將以上二維資料以二維串列儲存，本例使用 a 串列。
2. 使用串列儲存資料，就有索引，有索引就可使用迴圈操作以上資料。

程式列印

```
import time #使用time()函式
a=[["delighted", "adj", "高興的"],
    ["gear", "n", "服裝"],
    ["behave", "v", "表現"]]
n=3 #題數
right=0 #答對題數
wrong=0 #答錯題數
t1=int(time.time())#取得系統時間的秒數
#使用for迴圈逐一輸出每一題的中文與詞性
for i in range(n):
    print(a[i][2],a[i][1],':',end='') #輸出每一題的中文、詞性
```

```
b=input()#等待使用者輸入英文
if b==a[i][0]:#將使用者輸入與原題目比對
    right=right+1
else:
    wrong=wrong+1
t2=int(time.time())#取得系統時間的秒數
t=t2-t1
print('right=%d'% right)
print('wrong=%d'% wrong)
print('total time=%d second'% t)
```

👉 補充說明

1. `t2=int(time.time())` # 可取得系統時間，單位是秒數，使用前需要先載入 `time` 模組，請看 7-2 節。待程式結束時，再取系統時間，兩者相減，即可計算使用者使用的秒數。

👉 自我練習

1. 同範例 6-3a，但增加當使用者答題後，馬上告知正確與否；若是錯誤，也告知正確答案。
2. 同範例 6-3a，但增加為 100 個單字，每次測驗依亂數出現 10 個單字，同次測驗中所出現單字不能重複，統計使用者答對題數。亂數的取法請複習上冊 3-4 節。
3. 同範例 6-3a，但增加為 100 個單字，每次測驗 3 分鐘，題目依照亂數出現，同次測驗中所出現單字不能重複，統計使用者答對題數。亂數的取法請複習上冊 3-4 節。
4. 同範例 6-3a，但增加為 100 個單字，題目依照亂數出現，同次測驗中所出現單字不能重複，使用者答錯了才停止，統計使用者可連續答對題數。亂數的取法請複習上冊 3-4 節。
5. 請任選一個學科，如英文、國文等，製作 10 題填充題，寫一個程式讓使用者回答此 10 題填充題。

範例6-3b

選擇題測驗。假設英文單字的選擇題選項與答案資料如下：

編號	單字	類別	答案	選項 1	選項 2	選項 3	選項 4
1	delighted	adj	1	高興的	悲傷的	生氣的	憤怒的
2	gear	n	3	背包	褲子	服裝	鞋子
3	behave	v	2	生氣	表現	難過	看到

請寫一程式，可以讓使用者使用選擇題測驗以上 3 個單字，使用者每回答 1 題，電腦馬上回應正確或錯誤，結束測驗並統計答對與答錯題數及答題所花費時間。

執行結果

```

1 . delighted,adj :(1) 高興的 :(2) 悲傷的 :(3) 生氣的 :(4) 憤怒的 :
1
Right
2 . gear,n :(1) 背包 :(2) 褲子 :(3) 服裝 :(4) 鞋子 :
3
Right
3 . behave,v :(1) 生氣 :(2) 表現 :(3) 難過 :(4) 看到 :
1
Wrong
The Right number is :2
The wrong number is :1
total time=12 second

```

運算思維

1. 以上英文測驗題目是二維資料，所以使用二維串列儲存，如以下 a 串列，每筆資料第三個 (索引 2) 是答案，如以下第 1 題答案是 2。
2. 有了串列，就有索引編號，即可使用迴圈操作資料，程式如下：

程式列印

```

import time
a=[["delighted", "adj", "1","高興的", " 悲傷的", "生氣的", "憤怒的"],
   ["gear", "n","3" , "背包", "褲子","服裝", "鞋子"]
   ["behave", "v", "2", "生氣", "表現", "難過", "看到"]]
n=3 #題數

```

```

c=0 #答對題數
d=0 #答錯題數
t1=int(time.time())
#使用for迴圈逐一輸出每一題的題目與4個選項
for i in range(1,n+1):
    print("%d . %s,%s :" % (i,a[i][0],a[i][1]),end='')
                                                #輸出題號,英文單字,詞性

    for j in range(1,4+1):
        print("(%d) %s :" % (j,a[i][j+2]),end='')#輸出四個選項
    b=input() #等待使用者輸入答案
    if (b==a[i][2]): #比對
        c=c+1 #正確題數加1
        print("Right")
    else:
        print("Wrong")
        d=d+1 #錯誤題數加1
t2=int(time.time())
t=t2-t1
print("The Right number is :%d"%c)
print("The wrong number is :%d"%d)
print('total time=%d second'% t)

```

範例6-3c

前面 3-4 節的「小毛驢」程式有點冗長。請問，資料應該如何數位化，程式才會精簡呢？

👉 執行結果 (以下僅摘一小段)



我
有
一
隻

👉 運算思維

1. 本例一分鐘 80 拍，所以 1 拍的時間是 $60000\text{ms}/80$ ，以一個四分音符為 1 拍，所以一個八分音符所占的時間是 $60000\text{ms}/(2*80)$ ，每小節是 2 拍。

2. 小毛驢所有音符的最小速度是八分音符，所以我就取八分音符的時間為 1 個單位。
3. 簡譜的 1，是 Do，對應頻率是 523，所以，以 a 串列為對照表。(前面索引 0，我不用，所以放 0)

```
a = [ 0 , 5 2 3 , 5 8 7 , 6 5 9 , 6 9 8 , 7 8 4 , 8 8 0 , 9 9 8 , 1 0 4 6 ]
```

4. 將音階、歌詞、延遲時間數位化。以下僅作 4 小節，「我有一隻小毛驢我從來也不騎」。

```
song = [[1, '我', 1], [1, '有', 1], [1, '一', 1], [3, '隻', 1 ],
         [5, '小', 1 ], [5, '毛', 1 ], [5, '驢', 1 ], [5, '我', 1 ],
         [6, '從', 1 ], [6, '來', 1 ], [6, '也', 1 ], [8, '不', 1 ], [5, '騎', 4 ]]
```

以上每一列是由 [音階, 歌詞, 延遲時間] 所構成。

5. 使用迴圈依序輸出每一個列，參考程式如下：

👉 程式列印

```
import ctypes
p = ctypes.windll.kernel32
a = [ 0 , 523 , 587 , 659 , 698 , 784 , 880 , 998 , 1046 ]
t = 60000//160
song = [[1, '我', 1], [1, '有', 1], [1, '一', 1], [3, '隻', 1 ],
         [5, '小', 1 ], [5, '毛', 1 ], [5, '驢', 1 ], [5, '我', 1 ],
         [6, '從', 1 ], [6, '來', 1 ], [6, '也', 1 ], [8, '不', 1 ], [5, '騎', 4 ]]
#使用迴圈，每次輸出二維串列的一列
for i in song:
    print(i[1])#歌詞
    p.Beep(a[i[0]],i[2]*t)# 音階, 時間
```

範例6-3d

假設有兩個矩陣 A、B 如下：

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \quad B = \begin{pmatrix} 1 & 2 & 0 \\ 2 & 1 & 3 \end{pmatrix}$$

請求其相加的結果如下：

$$\begin{pmatrix} 2 & 4 & 3 \\ 6 & 6 & 9 \end{pmatrix}$$

請問要使用何種資料結構較省事？

👉 執行結果

```
[[1, 2, 3], [4, 5, 6]]
[[1, 2, 0], [2, 1, 3]]
[[2, 4, 3], [6, 6, 9]]
2 4 3
6 6 9
```

👉 程式列印

1. 本題因為是二維資料，若用二維串列儲存，就可以使用雙迴圈處理縱向與橫向二維資料。

```
#逐一初始化每一串列
a=[[1,2,3],[4,5,6]]
b=[[1,2,0],[2,1,3]]
c=[[0,0,0],[0,0,0]]
#使用雙迴圈，將兩個二維串列a與b，逐一將每一行、列資料相加，放到c串列
for i in range(0,2): #逐一處理每一列
    for j in range(0,3): #逐一處理每一行
        c[i][j]=a[i][j]+b[i][j]
print(a) #使用函式輸出a串列
print(b) #使用函式輸出b串列
print(c) #使用函式輸出c串列
#自行使用雙迴圈輸出c串列
for i in range(0,2):
    for j in range(0,3):
        print(c[i][j],end=" ")
    print() #跳列
```

👉 自我練習

1. 假設有矩陣如下：

$$\begin{pmatrix} 1 & 2 & 0 \\ 0 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 0 \end{pmatrix}$$

- (1) 請問以何種資料結構儲存。
- (2) 請檢查幾個 0。
- (3) 請統計所有元素和。
- (4) 再將每一元素乘以 2 後輸出。

範例6-3e

假設有學生成績如下：

座號	國文	英文	數學	平均	不及格科數
1	50	60	70		
2	30	40	50		
3	70	80	90		
4	66	77	88		
5	22	33	44		
平均					

1. 請選用適當資料結構儲存以上資料。
2. 計算每人平均。
3. 統計每人不及格科數。
4. 統計各科平均。

執行結果

```
No chin eng math avg fail
1 50 60 70 60 1
2 30 40 50 40 3
3 70 80 90 80 0
4 66 77 88 77 0
5 22 33 44 33 3
0 47 58 68 58 2
```

運算思維

1. 本例雖可使用 5 個一維串列儲存以上資料，這樣只簡化縱向的各科成績計算，但橫向每一位學生的平均計算還是無法使用迴圈，若使用二維串列，那就可以直向、縱向通通配合迴圈，使得程式非常精簡。
2. 使用二維 a 串列儲存以上二維資料

```
a=[[1,50,60,70,0,0],
    [2,30,40,50,0,0],
    [3,70,80,90,0,0],
    [4,66,77,88,0,0],
    [5,22,33,44,0,0],
    [0,0,0,0,0,0]]
```

3. 以上資料的列、行索引如下表：

座號	國文	英文	數學	平均	不及格科數
1 (0,0)	50 (0,1)	60 (0,2)	70 (0,3)	(0,4)	(0,5)
2 (1,0)	30 (1,1)	40 (1,2)	50 (1,3)	(1,4)	(1,5)
3 (2,0)	70 (2,1)	80 (2,2)	90 (2,3)	(2,4)	(2,5)
4 (3,0)	66 (3,1)	77 (3,2)	88 (3,3)	(3,4)	(3,5)
5 (4,0)	22 (4,1)	33 (4,2)	44 (4,3)	(4,4)	(4,5)
科目平均	(5,1)	(5,2)	(5,3)	(5,4)	(5,5)

4. 每個人國英數的橫向資料累加。此一累加過程為「行」索引先變化，例如 $(0,1)+(0,2)+(0,3)$ 先累加，然後要放到 $(0,4)$ ，所以內迴圈 j 要從 1 到 3，且運算後放在「行」索引 4。然後，外迴圈 i 要從 0 到 4，分別計算 1 號到 5 號的成績和，變數 i 為則為「列」索引。所以程式如下：

```
for i in range(0,5):#每個學生
    for j in range(1,4):#累加1,2,3等3個科目
        a[i][4]=a[i][4]+a[i][j]
```

5. 每個科目國、英、數的縱向資料累加。此一累加過程為「列」索引先變化，例如， $(0,1)+(1,1)+(2,1)+(3,1)+(4,1)$ 先累加，然後放在 $(5,1)$ ，所以，內迴圈 j 要從 0 到 4，且放在「列」索引 5。然後，外迴圈 i 要從 1 到 4，分別計算『國、英、數、平均』等 4 個科目的和，此變數 i 則為「行」索引，所以程式如下：

```
for i in range(1,5):#每個科目
    for j in range(0,5):#累加每個學生的成績
        a[5][i]=a[5][i]+a[j][i]
```

👉 程式列印

```
#使用二維串列儲存二維資料
a=[[1,50,60,70,0,0],
   [2,30,40,50,0,0],
   [3,70,80,90,0,0],
   [4,66,77,88,0,0],
   [5,22,33,44,0,0],
   [0,0,0,0,0,0]]
#逐一計算每個人總分、不及格科數
for i in range(0,5):
    for j in range(1,4):
        a[i][4]=a[i][4]+a[i][j] #計算每個人總分
        #計算不及格科數
        if a[i][j]<60:
            a[i][5]=a[i][5]+1 #計算不及格科數
    a[i][4]=a[i][4]/3 #計算每個人平均
#逐一計算每一科目總分
for i in range(1,5):
    for j in range(0,5):
        a[5][i]=a[5][i]+a[j][i] #累加每一個人的分數
    a[5][i]=a[5][i]/5 #計算每一科目的平均
#輸出每一個人成績
for i in range(0,6):
    print(a[i])
```

👉 自我練習

1. 假設某一公司有三個業務員，其每季的業績如下：

編號	第一季	第二季	第三季	第四季	和	零的個數
1	0	2	4	6		
2	8	0	2	1		
3	7	0	0	9		
和						
零的個數						

- (1) 計算每一個人的業績總和及零的個數。
- (2) 計算每一季的業績總和及零的個數。
- (3) 請問資料結構為何？

範例6-3f

邏輯運算子（APCS106 年 10 月試題）

1. 小蘇最近在學三種邏輯運算子 AND、OR 和 XOR。這三種運算子都是二元運算子，也就是說在運算時需要兩個運算元，例如 $a \text{ AND } b$ 。對於整數 a 與 b ，以下三個二元運算子的運算結果定義如下列三個表格：

a AND b			a OR b			a XOR b		
	b 為 0	b 不為 0		b 為 0	b 不為 0		b 為 0	b 不為 0
a 為 0	0	0	a 為 0	0	1	a 為 0	0	1
a 不為 0	0	1	a 不為 0	1	1	a 不為 0	1	0

舉例來說：

- (1) $0 \text{ AND } 0$ 的結果為 0， $0 \text{ OR } 0$ 以及 $0 \text{ XOR } 0$ 的結果也為 0。
 - (2) $0 \text{ AND } 3$ 的結果為 0， $0 \text{ OR } 3$ 以及 $0 \text{ XOR } 3$ 的結果則為 1。
 - (3) $4 \text{ AND } 9$ 的結果為 1， $4 \text{ OR } 9$ 的結果也為 1，但 $4 \text{ XOR } 9$ 的結果為 0。
2. 請撰寫一個程式，讀入 a 、 b 以及邏輯運算的結果，輸出可能的邏輯運算為何。

測試資料

範例一：輸入 0 0 0 範例一：正確輸出 AND OR XOR	範例二：輸入 1 1 1 範例二：正確輸出 AND OR
範例三：輸入 3 0 1 範例三：正確輸出 OR XOR	範例四：輸入 0 0 1 範例四：正確輸出 IMPOSSIBLE

執行結果

```
input a b c:0 0 0
AND
OR
XOR
```

👉 運算思維

1. a,b 若不為 0，就修正為 1。
2. 採用二維串列查表的方式，將 AND、OR、XOR 的結果，使用二維串列定義好。

```
d=[[0,0],[0,1]]#AND, [0,0]is 0,[0,1]=0,[1,0]=0,[1,1]=1
e=[[0,1],[1,1]]#OR, [0,0]is 0,[0,1]=1,[1,0]=1,[1,1]=1
f=[[0,1],[1,0]]#XOR [0,0]is 0,[0,1]=1,[1,0]=1,[1,1]=0
```

3. 檢查輸入與對應到的輸出是否滿足『AND』，若是，則輸出『AND』且標記 found=True。
4. 檢查輸入與對應到的輸出是否滿足『OR』，若是，則輸出『OR』且標記 found=True。
5. 檢查輸入與對應到的輸出是否滿足『XOR』，若是，則輸出『XOR』且標記 found=True。
6. 假如，not found，代表沒有一項成立，輸出『IMPOSSIBLE』。

👉 程式列印

```
a,b,c=input("input a b c:").split(" ")
a=int(a)
b=int(b)
c=int(c)
#修正
if a!=0 :
    a=1
if b!=0:
    b=1
#使用二維串列定義表格
d=[[0,0],[0,1]]#AND
e=[[0,1],[1,1]]#OR
f=[[0,1],[1,0]]#XOR
#一一檢查是否滿足
found=False
if d[a][b]==c:
    print("AND")
    found=True
if e[a][b]==c:
```

```

    print("OR")
    found=True
if f[a][b]==c:
    print("XOR")
    found=True
if not found:
    print("IMPOSSIBILE")

```

範例6-3g

最大和（APCS105 年 10 月試題）

給定 N 群數字，每群都恰有 M 個正整數。若從每群數字中各選擇一個數字（假設第 i 群所選出數字為 t_i ），將所選出的 N 個數字加總即可得總和 $S = t_1 + t_2 + \dots + t_N$ 。請寫程式計算 S 的最大值（最大總和），並判斷各群所選出的數字是否可以整除 S 。

第一行輸出最大總和 S 。

第二行按照被選擇數字所屬群的順序，輸出可以整除 S 的被選擇數字，數字與數字間以一個空格隔開，最後一個數字後無空白；若 N 個被選擇數字都不能整除 S ，就輸出 -1。

範例一：輸入

3 2
1 5
6 4
1 1

範例一：正確輸出

12
6 1

（說明）挑選的數字依序是 5,6,1，總和 $S=12$ 。而此三數中可整除 S 的是 6 與 1，6 在第二群，1 在第 3 群所以先輸出 6 再輸出 1。注意，1 雖然也出現在第一群，但她不是第一群中挑出的數字，所以順序是先 6 後 1。

範例二：輸入

4 3
6 3 2
2 7 9
4 7 1

9 5 3

範例二：正確輸出

31
-1

（說明）挑選的數字依序是 6,9,7,9，總和 $S=31$ 。而此四數中沒有可整除 S 的，所以第二行輸出 -1。

👉 執行結果

```
input N M:3 2

input data:1 5

input data:6 4

input data:1 1
[[1, 5], [6, 4], [1, 1]]
12
6 1
```

👉 程式列印

```
#由使用者輸入列數、行數
N,M=input("input N M:").split(" ")
N=int(N) #列數
M=int(M) #行數
a = [[0]*M for i in range(N)] #初始化N列M行
b=[0]*N #用來儲存每列的最大值
#由使用者輸入N列、M行資料
#C/C++資料預設空白，Python預設逗點(,)，
#所以當資料與資料之間以空白隔開，都要以空白分割輸入
for i in range(N):
    d1=input("input data:").split(" ")
    d3=[int(d2) for d2 in d1]
    for j in range(M):
        a[i][j]=d3[j]
#輸出所輸入資料，確認是否正確
print(a)
#求每列最大值，放入b串列
for i in range(N):
    for j in range(M):
        if a[i][j]>b[i]:
            b[i]=a[i][j]
#統計每列最大值的和，放入變數s
s=0
for i in range(N):
    s=s+b[i]
print(s)
#逐一於每列最大值，找出能整除s者
```

```

found=False
for i in range(N):
    if s % b[i]==0:
        if found :
            print(" %d" % b[i],end="")
        else:
            print("%d" % b[i],end="")
            found=True
#若每列最大值，均無法整除s，則輸出「-1」
if not found :
    print("-1")

```

6-4 串列函式與串列生成式

⚙️ 串列函式

前面 6-2、6-3 節是由程式開發者自行使用迴圈操作串列，Python 的特色是增加一些處理串列的函式，這樣就可以減輕程式設計師的工作。以下是一些常用的串列函式：

▶ sum()

前面範例 6-2a 已經介紹如何使用迴圈計算串列的和。Python 則提供 sum() 函式計算串列和。例如，

```

a=[3,2,1]
print(sum(a))#6

```

也可指定範圍，也就是可跳過非數值的資料。例如：

```

a=["aa",3,2,1]
print(sum(a[1:4]))#6 [1:4]表示取索引1~4，但不含索引4

```

二維陣列也可以，例如：

```

a=[[1,50,60,70,0,0],
   [2,30,40,50,0,0],
   [3,70,80,90,0,0],

```

```
[4,66,77,88,0,0],  
[5,22,33,44,0,0],  
[0,0,0,0,0,0]]  
a[0][4]=sum(a[0][1:4])  
print(a[0]) #1,50,60,70,180,0
```

▶ max()、min()

求串列極大與極小，與 sum() 用法相同，請自行練習。

▶ append()

在串列末端加入元素。例如：

```
a=[1,2,3]  
a.append(4)  
print(a) #1,2,3,4
```

▶ insert()

在指定索引加入資料，原索引資料則後退。例如：

```
a=[0,1,2,3]  
a.insert(2,4)  
print(a) #0,1,4,2,3
```

又例如，以下程式可將資料加在最前面，這樣就如同佇列的操作。

```
a=[1,2,3]  
a.insert(0,0)  
print(a) #0,1,2,3
```

▶ pop()

取出指定索引元素，且刪除此元素。例如：

```
a=[0,1,2,3]  
b=a.pop(1)  
print(b) #1  
print(a) #[0,2,3]
```

若沒有指派索引值，那就是刪除串列末端元素，所以可以拿來當作佇列用。

```
a=[0,1,2,3]
b=a.pop()
print(b)#3
print(a)#[0,1,2]
```

► remove()

取出指定元素，且刪除此元素。例如：

```
a=[1,2,3]
a.remove(2)
print(a)#[1,3]
```

► sort()

排序串列，例如：

```
a=[1,8,3]
a.sort()#由小而大
print(a)#1,3,8
a.sort(reverse=True)#由大到小
print(a)#8,3,1
```

二維串列也可以使用 sort() 排序，但要使用 lambda 指派欄位。例如：

```
a=[[1,50,60,70,0,0],
   [2,30,40,50,0,0],
   [3,70,80,90,0,0],
   [4,66,77,88,0,0],
   [5,22,33,44,0,0],]
a.sort(key=lambda x:x[1])#欄位1
for i in range(5):
    print(a[i])
```

```
[5, 22, 33, 44, 0, 0]
[2, 30, 40, 50, 0, 0]
[1, 50, 60, 70, 0, 0]
[4, 66, 77, 88, 0, 0]
[3, 70, 80, 90, 0, 0]
```

▶ sorted()

前面 sort 是將串列排序，sorted() 則將排序結果複製到另一串列，且原串列不變。例如：

```
a=[1,8,3]
b=sorted(a)#由小而大
print(b)#1,3,8
b=sorted(a,reverse=True)#由大到小
print(b)#8,3,1
print(a)#1,8,3
```

▶ extend()

合併兩個串列。例如：

```
a=[1,2,3]
b=[4,5]
a.extend(b)
print(a)#1,2,3,4,5
```

▶ in

判斷元素是否在物件裡面。物件可以是 List、Tuple、Dict 或 Set。例如：

```
a=[1,2,3]
print(1 in a)#True
```

▶ not in

判斷元素是否『不在』物件裡面，用法同 in。

▶ len()

傳回串列長度。例如：

```
a=[1,2,3]
print(len(a))#3
```

▶ `count()`

傳回指定元素出現次數。例如：

```
a=[1,2,3,2]
print(a.count(2))#2
```

▶ `index()`

傳回指定元素第一次出現的索引。例如：

```
a=[2,3,1,3]
print(a.index(3))#1
```

⚙️ 串列生成式

前面 6-2、6-3 節是由程式開發者自行使用迴圈操作串列，串列生成式 (List Generator) 也是 Python 很神的方法，其方式是可以在串列的括號內使用運算式與迭代物件產生新的串列。其語法如下：

```
new串列=[運算式 for 元素 in 迭代物件]
```

例如，以下程式可以將整個串列元素都乘以 2，再放入另一串列。

```
a=[1,2,3]
b=[a1*2 for a1 in a]#以上兩個a1，變數名稱要相同，且遵守變數命名規則
print(b)#2 4 6
```

以下程式可快速初始化串列：

```
c=[0 for i in range(5)]
print (c)#0,0,0,0,0
d=[i for i in range(5)]
print (d)#0,1,2,3,4
```

又例如，以下串列 a 的元素都是字串，那要如何通通轉為數值呢？程式如下：

```
a=['1','2','3']
print(a[0]+a[1])#12
```

```
b=[int(a1) for a1 in a]
print(b[0]+b[1])#3
```

⚙️ 條件生成

Python 真的非常富有想像力，串列生成式還可加條件。例如，以下可將及格的分數放到 b 串列。

```
a=[22,66,33,88]
b=[a1 for a1 in a if a1>=60]
print(b)#66 88
```

又例如，以下程式可直接找出指定數字 12 的因數。

```
a=[i for i in range(1,12+1) if 12 % i==0]
print(a)
```

又例如，以下程式就可直接完成三位數阿姆斯壯數。

```
a=[[100*i+10*j+k]for i in range(1,9+1) for j in range (0,9+1)
for k in range(0,9+1) if (100*i+10*j+k)== (i**3+j**3+k**3)]
print (a)
```

又例如，以下程式可將二維陣列每一個元素加 1。

```
a=[[1,1,1,0,1],[0,1,0,1,1],[0,0,0,0,0],[0,0,0,1,0]]
b=[[ a11+1 for a11 in a1] for a1 in a]
print(b)
```

👉 自我練習

1. 請鍵入以下程式，並觀察執行結果。

```
a = [[i for i in range(3)] for j in range(4)]
print(a)
print(a[0][0])
b=[sum(a1) for a1 in a ]
print (b)
```

2. 假設有資料如下，單位是公斤

2,5,6,8,10

請使用串列儲存，且將它轉為台斤數且放到另一串列。

3. 請寫一程式，可以初始化一個 100 個元素的一維串列，每一個元素是 1 到 6 的亂數，其次，將偶數放到另一串列。
4. 請寫一程式，將二維陣列初始化如下：

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31

範例6-4a

編碼（106/10 APCS 試題）

1. 輸入。

第一列為取碼長度 m 。例如：

```
1
```

第二列輸入若干數字，以 0 結束。例如：

```
252 373 28 0
```

2. 處理

將第二列的每一項數字從左邊取長度 m 的數字字元，然後比較這些取出的數字大小，取極大值輸出。

將第二列的每一項數字從右邊取長度 m 的數字字元，然後比較這些取出的數字大小，取極小值輸出。

3. 範例

資料編號	輸入	輸出
1	1 (取碼長度) 252 373 28 0	2 3 2 max=3 (取左邊 1 個) 2 3 8 min=2 (取右邊 1 個) 3 2
2	2 (取碼長度) 258 0	25 max=25 (取左邊 1 個) 58 min=58 (取右邊 1 個) 25 58
3	2 (取碼長度) 123 58 612 0	12 58 61 max=61 取左邊 2 個) 23 58 12 min=12 (取右邊 2 個) 61 12

👉 執行結果

```
['252', '373', '28']
['2', '3', '2']
3
['2', '3', '8']
[2, 3, 8]
2
```

👉 程式列印

```
m=1 #取碼長度，測試資料1
a=[252,373,28] #先將資料以串列儲存
#m=2 #取碼長度，測試資料2
#a=[258]
#m=2 #取碼長度，測試資料3
#a=[123,58,612]
b=[str(a1) for a1 in a] #逐一將串列每一元素轉字串
print(b) #輸出資料，確認是否正確
c=[b1[0:m] for b1 in b] #串列每一元素取左邊m個位數
print(c) #['2', '3', '2']
d=[int(c1) for c1 in c] #將取到的資料轉回數值
print(max(d)) # 3，依題目要求，取資料的極大值，再輸出
#取右邊指定m位數，然後取極小值
#print(min([int(b1[-m:]) for b1 in b])) #用這一行就代表下面全部
e=[b1[-m:] for b1 in b] #取右邊指定位數
print(e)
f=[int(e1) for e1 in e] #轉回數值
print(f)
print(min(f)) #取極小值
```

自我練習

1. 以上資料的輸入，我們先使用變數 `a` 指派測試資料，如程式 `a=[252,373,28]`，請同學依題目說明，自行改為使用「`input`」指令輸入測試資料。

6-5 串列程式設計實作演練

排序 (Sort)

資料由大而小或由小而大稱為排序。前面 4-5 節，我們已經介紹 3 ~ 5 筆資料的氣泡排序，資料量少時，我們還可逐一寫程式完成，倘若有 1000 筆，甚至 10000 筆呢，這時候就要使用串列配合迴圈，本書僅提供 2 種簡單演算法，分別是氣泡排序法與計數排序法，分別說明如下：

氣泡排序法 (Bubble Sort)

在一數列當中，從第一筆資料逐一往後兩兩給予比較，若前者大於後者，則兩者交換（本例為由小而大排序，若由大而小排序，則當前者小於後者時，兩者交換），每次的比較與交換均可得該數列的最大值於數列最右邊（末端），所以若有 N 筆資料進行排序，則應作 $N-1$ 階次的逐一比較，且每階次的逐一比較範圍均逐漸減一個，此即為氣泡排序法 (Bubble Sort)。若有 8、9、7、1、2 五筆資料，則其排序過程如下：

1. 將資料由左而右排列如下：
8 9 7 1 2
2. 本例共有五筆資料 $N=5$ ，必須進行四個階次的逐一比較，每一階次都要將該階次的最大值移至最右邊。此一程式在 4-5 節已經介紹，但是當資料量變多時，程式將會很冗長，現在將找出規律性，由規律性導入迴圈，才能簡化程式的撰寫。所以我們以 i 迴圈代表階次， j 迴圈代表每階次所要執行的次數，以上四個階次的逐一比較， i, j 的變化說明如下：
(1) 從第一筆到第五筆兩兩比較，若前者大於後者則兩者交換如表 6-4a，得最大值 9 於最右邊。9 已是最大值，待會兒不必參與比較，所以下一次只要從第一筆比較至第四筆即可。

表 6-4a 汽泡排序法階次一

i=1 (階次)	比較前	8 9 7 1 2	
j=1, 2, 3, 4 (比較次數)	比較次數	↓ ↓ ↓ ↓	比較四次
	比較後	8 7 1 2 9	

- (2) 從第一筆到第四筆兩兩比較，若前者大於後者，則兩者交換如表 6-4b，得次大值 8 於右邊。8 與 9 已達定位，待會兒不用參與比較。

表 6-4b 汽泡排序法階次二

i=2 (階次)	比較前	8 7 1 2 9	
j=1, 2, 3 (比較次數)	比較次數	↓ ↓ ↓	比較三次
	比較後	7 1 2 8 9	

- (3) 從第一筆資料到第三筆資料兩兩比較，若前者大於後者則兩者交換如表 6-4c，得第三大值 7 於右邊。

表 6-4c 汽泡排序法階次三

i=3 (階次)	比較前	7 1 2 8 9	
j=1, 2 (比較次數)	比較次數	↓ ↓	比較二次
	比較後	1 2 7 8 9	

- (4) 從第一筆資料到第二筆資料兩兩比較，若前者大於後者則兩者交換如表 6-4d，排序完成。

表 6-4d 汽泡排序法階次四

i=4 (階次)	比較前	1 2 7 8 9	
j=1 (比較次數)	比較次數	↓	比較一次
	比較後	1 2 7 8 9	

- 若有 N 筆資料，則應作 $N-1$ 階次的比較與交換，每階次的比較範圍均逐漸縮小 $[N-1, N-2, \dots, 1]$ ，其比較次數總計大約為 $[N*(N-1)]/2$ ，資料交換次數大約為 $[N*(N-1)]/4$ 。
- 以上每次均找到一個最大值於右邊，若將該次搜尋所得最大值與右邊位置交換，則可大大減少資料交換次數，且其比較次數不變仍為 $[N*(N-1)]/2$ ，但其資料交換次數總計約為 N ，請讀者自行練習。

以上氣泡排序法，在第四章已經寫程式完成，當時資料限於 5 筆內，逐一比較與交換尚可克服，但若資料超過 5 筆，那程式可就落落長，現在使用迴圈的演算法如下：

1. 設有 N 筆資料進行排序。
2. 應進行 N-1 階次尋找最大值，並將最大值移至最右邊，我們以 i 迴圈表示如下：

```
for(i=1; i<=N-1; i++)
```

3. 每一階次均應逐次尋找最大值，並移至右邊，每階次的比較次數以 j 表示，且 j 與外迴圈 i 的關係為 $i+j=N$ ，所以內迴圈的比較次數 j 為 N-i。

```
for(j=1; j<=N-i; j++)
```

4. 兩兩逐一比較，若前者大於後者則兩者交換。

```
if(a[j]>a[j+1]) { swap(a[j],a[j+1]) }
```

範例6-5a

示範氣泡排序法。

👉 執行結果

```
[0, 1, 2, 3, 4, 5]
```

👉 程式列印

```
a=[0,5,4,3,2,1]#要用最糟的初值，索引0不用
n=5
for i in range(1,n-1+1):
    for j in range(1,n-i+1):
        if a[j]>a[j+1]:
            a[j],a[j+1]=a[j+1],a[j]
print(a)
```

👉 補充說明

1. 本例串列索引從 1 開始用。
2. range 因為不含終值，本書習慣外加一個 1。
3. 若索引從 0 使用，那外迴圈要減 1，內迴圈要減 2，程式如下：

```
a=[5,4,3,2,1]#要用最糟的初值
n=5
for i in range(0,n-1+1-1):
    for j in range(0,n-i+1-2):
        if a[j]>a[j+1]:
            a[j],a[j+1]=a[j+1],a[j]
print(a)
```

4. 以上程式簡化如下：

```
a=[5,4,3,2,1]#要用最糟的初值
n=5
for i in range(0,n-1):
    for j in range(0,n-i-1):
        if a[j]>a[j+1]:
            a[j],a[j+1]=a[j+1],a[j]
print(a)
```

👉 自我練習

1. 同上範例，但每階次均尋找一個最大值，且逐次與第 $n, n-1, n-2$ 筆資料交換即可。

範例6-5b

假設有資料如下：

座號	姓名	國文	數學	缺課時數	總分
1	aa	8	10	1	18
2	bb	9	9	2	18
3	cc	7	10	0	17
4	dd	10	9	1	19
5	ee	9	8	2	17

- (1) 請以適當資料結構儲存。
- (2) 以總分為依據，從小而大輸出。

👉 執行結果

```
cc    7   10   0   17
ee    9    8   2   17
aa    8   10   1   18
bb    9    9   2   18
dd   10    9   1   19
```

👉 程式列印

```
n=5
a=[['aa',8,10,1,0],
    ['bb',9,9,2,0],
    ['cc',7,10,0,0],
    ['dd',10,9,1,0],
    ['ee',9,8,2,0]]
print (a)
for i in range(n):
    a[i][4]=a[i][1]+a[i][2]
print (a)
for i in range(n-1):
    for j in range(n-i-1):
        if a[j][4]>a[j+1][4]:
            #不只成績要交換，所有欄位都要換
            for k in range(5):
                a[j][k],a[j+1][k]=a[j+1][k],a[j][k]
print(a)
for i in range(n):
    print(a[i][0],end=" ")
    for j in range(1,4+1):
        print("%4d" %(a[i][j]),end=' ')
    print()
```

👉 自我練習

1. 同範例，但總分相同時，請以數學為排序依據。
提示：將（總分 *10+ 數學）當作排序依據就可以。

⚙️ 計數排序法 (Counting Sort)

前面的氣泡排序法是採用兩者比較並交換，達到由小而大或由大而小排列的效果，不過有時候我們還是希望按照原順序輸出，此時就可以使用計數排序法，因為計數排序法是不移動資料，但直接給予名次。為了充份理解計數排序法的原理，你先假想一個 N 位學生的教室裡，每人均舉著自己的分數，每一學生如何知道自己是第幾名呢？有一個辦法就是每個人均數一數分數比自己高的人數再加一（一就是自己），就是自己的名次，其比較次數是 $N*N$ 。若令他們排成一排，由第一個逐一往右比較，且分數低的，令其名次加 1，則第二個人就不必與第一個比較，同理，第三個也是往右比較即可，不必與第一、二個人比較，所以其總比較次數約為 $[N*(N-1)]/2$ ，此即為計數排序法，若有數列資料 8、2、9、7、1 則其演算過程如下：

1. 先假設每人均為第一名，如表 6-5a。

表 6-5a 計數排序準備表

分數	8	2	9	7	1
名次	*	*	*	*	*

2. 共有五筆資料，分別以前四筆資料為基準往右比較：
 - (1) 以第一筆資料 8 為基準，往右（第二、三、四、五筆）比較，分數低的，將其名次加一，則其名次如表 6-5b，第一筆資料 8 確定為第二名，下次不用再參與比較。

表 6-5b 計數排序階次一

分數	8	2	9	7	1
名次	**	**	*	**	**

- (2) 以第二筆資料 2 為基準，往右（第三、四、五筆）比較，分數低的，將其名次加一，則其名次如表 6-5c，第二筆資料 2 確定為第四名，下次不用再參與比較。

表 6-5c 計數排序階次二

分數	8	2	9	7	1
名次	**	****	*	**	***

- (3) 以第三筆資料 9 為基準，往右（第四、五筆）比較，分數低的，將其名次加一，則其名次如表 6-5d，第三筆資料 9 確定為第一名，下次不用參與比較。

表 6-5d 計數排序階次三

分數	8	2	9	7	1
名次	**	****	*	***	*****

- (4) 以第四筆資料 7 為基準，往右（第五筆）比較，分數低的，將其名次加一，則其名次如表 6-5e，全部名次確定，排序完成。

表 6-5e 計數排序階次四

分數	8	2	9	7	1
名次	**	****	*	***	*****

3. 若有 N 筆資料，則須作 N-1 階次的比較，每階次的比較次數均逐漸縮小，分別為 N-1，N-2，.....，1，故其總比較次數約為 $[N*(N-1)]/2$ 。

範例6-5c

示範計數排序法。

執行結果

```
[1, 1, 1, 1, 1, 1]
[0, 8, 2, 9, 7, 1]
[1, 2, 4, 1, 3, 5]
```

程式列印

1. 以下程式，索引 0 先不用，索引從 1 開始，較符合人類思考方式。

```
a=[0,8,2,9,7,1] #以串列儲存每一個人的成績
b=[1 for a1 in a]#先指派每個人都是第1名
n=5 #資料個數
print (b)
for i in range(1,n-1+1): #排序所需階次
    for j in range(i+1,n+1): #每階次所需比較次數
        if a[i]>a[j]:
```

```

        b[j]=b[j]+1#分數低的名次加1
    else:
        b[i]=b[i]+1
print(a) #成績
print(b) #名次

```

2. 以下程式索引改從 0 開始，則內外迴圈通通減 1 就可以，程式如下：

```

n=5
a=[8,2,9,7,1] #每一個人的成績
b=[1 for a1 in a] #先指派每個人都是第1名
print (b)
for i in range(n-1):
    for j in range(i+1,n):
        if a[i]>a[j]:
            b[j]=b[j]+1
        else:
            b[i]=b[i]+1
print(a) #成績
print(b) #名次

```

以上氣泡排序與計數排序的差異，如表 6-6 所示。

表 6-6 氣泡排序法與計數排序法差異比較表

比較項目	氣泡排序法	計數排序法
資料是否移動	是	否
是否需要增加記憶體	否	是
迴圈階次	雙迴圈	雙迴圈

範例6-5d

假設有資料如下：

座號	姓名	國文	數學	缺課時數	總分
1	aa	8	10	1	18
2	bb	9	9	2	18
3	cc	7	10	0	17
4	dd	10	9	1	19
5	ee	9	8	2	17

- (1) 請以適當資料結構儲存。
- (2) 以總分為依據，計算其名次，輸出以座號小者先輸出。

👉 執行結果

aa	8	10	1	18	3
bb	9	9	2	18	2
cc	7	10	0	17	5
dd	10	9	1	19	1
ee	9	8	2	17	4

👉 程式列印

1. 本例預留一個位置 a[][5] 放名次。

```
n=5
a=[['aa',8,10,1,0,0],
    ['bb',9,9,2,0,0],
    ['cc',7,10,0,0,0],
    ['dd',10,9,1,0,0],
    ['ee',9,8,2,0,0]]
print (a)
for i in range(n):
    a[i][4]=a[i][1]+a[i][2]#計算總分
    a[i][5]=1#指派名次都是第1名
print (a)
for i in range(n-1):
    for j in range(i+1,n):
        if a[i][4]>a[j][4]:
            a[j][5]=a[j][5]+1
        else:
            a[i][5]=a[i][5]+1
for i in range(n):
    print(a[i][0],end=" ")
    for j in range(1,5+1):
        print("%4d" %(a[i][j]),end=' ')
    print()
```

👉 自我練習

1. 以上程式沒有考慮同分的處理，請自行思考如何處理同分同名次，也就是 18 分應該並列第 2，然後要跳掉第 3 名，17 分要第 4 名。

2. 同範例 6-5d，總分相同時，請以數學為排序依據。(提示：因為每科成績上限為 10 分，所以可以增加 1 個欄位，此欄位為總分 *10+ 數學成績，且以此欄位作為排序依據。)
3. 延伸思考，同自我練習 2，總分相同時，請以數學較低分的為優先。

🔧 搜尋 (Search)

我們日常生活總是不斷的搜尋，在家裡找東西、到圖書館找書、到超商買東西，都是在搜尋。本單元將介紹若干資料搜尋的方法，分別是循序搜尋 (Sequential Search) 與二分搜尋法 (Binary Search)，分別說明如下：

🔧 循序搜尋法

於序列中，從頭到尾逐一比對搜尋的方式，稱為循序搜尋法，其次，若是資料沒有按照某一規則存放，那就只能靠循序搜尋法，請看以下範例。

範例 6-5e

假設有員工電話如下，請以適當資料結構儲存，並能輸入姓名而查得電話號碼。

編號	姓名	電話號碼
1	aa	1111168
2	hh	2222168
3	cc	3333168
4	gg	4444168
5	ff	5555168
6	ii	6666168
7	ee	7777168
8	bb	8888168
9	jj	9999168
10	dd	1688168
11	kk	2688268
12	ll	3688368

👉 執行結果

1. 如輸入『bb』傳回 bb 的電話號碼。

```
input name:bb
bb
8888168
```

👉 程式列印

將資料以二維串列儲存，使用循序法逐一搜尋資料。

```
a=[['aa','1111168'],['hh','2222168'],['cc','3333168'],
    ['gg','4444168'],['ff','5555168'],['ii','6666168'],
    ['ee','7777168'],['bb','8888168'],['jj','9999168'],
    ['dd','1688168'],['kk','2688168'],['ll','3688168']]
b=len(a)
found=False
c=input("input name:")
print(c)
#使用for迴圈，逐一於串列搜尋
for i in range(b):
    if a[i][0]==c:
        found=True
        break
if found:
    print(a[i][1])
else:
    print(" not found")
```

👉 自我練習

1. 同範例 6-5e，但可重複查詢。
2. 假如有帳號密碼如下表，只有輸入這些帳號與密碼才能使用範例 6-5c 的計數排序，請寫程式完成此功能。

編號	帳號	密碼
1	aa	1111168
2	hh	2222168
3	cc	3333168
4	gg	4444168
5	ff	5555168
6	ii	6666168

編號	帳號	密碼
7	ee	7777168
8	bb	8888168
9	jj	9999168
10	dd	1688168
11	kk	2688268
12	ll	3688368

⚙️ 二分搜尋法

前面的線性搜尋法，是採用逐一訪視的方式搜尋所要的資料，但當資料量龐大時，就顯得毫無效率。舉例而言，資料數量若有 1000 筆，則採用線性搜尋法平均搜尋次數為 500 次，但若採用本節所討論的二分搜尋法，則至多不會超過 10 次 ($2^{10}=1024$)。使用二分搜尋法前應先將資料排序。在一個已排序的序列中，首先與序列的中間值比較，若待搜尋資料比中間值大，則待搜尋資料必在此中間值的後面，反之則在前面，所以二分搜尋法每次可將搜尋範圍減半，如此對於資料量龐大時，可提高搜尋效率，請看以下範例：

範例6-5f

資料同範例 6-5e，但使用二分搜尋法。

👉 執行結果

1. 將串列以氣泡排序法排序如下。
2. 以下為輸入“dd”的執行結果。

```
[['aa', '1111168'], ['bb', '8888168'], ['cc', '3333168'],
['dd', '1688168'], ['ee', '7777168'], ['ff', '5555168'],
['gg', '4444168'], ['hh', '2222168'], ['ii', '6666168'],
['jj', '9999168'], ['kk', '2688168'], ['ll', '3688168']]

input name:dd
dd
1688168
```

👉 程式列印

1. 本例使用二維串列儲存資料。
2. 資料要先排序，才能使用二分搜尋法。

```

a=[['aa','1111168'],['hh','2222168'],['cc','3333168'],
    ['gg','4444168'],['ff','5555168'],['ii','6666168'],
    ['ee','7777168'],['bb','8888168'],['jj','9999168'],
    ['dd','1688168'],['kk','2688168'],['ll','3688168']]
#使用氣泡排序法排序
n=12 #資料筆數
for i in range(0,n-1):
    for j in range(0,n-i-1):
        if a[j][0]>a[j+1][0]:#使用姓名欄位由小而大排序
            #兩個欄位都要交換
            a[j][0],a[j+1][0]=a[j+1][0],a[j][0]
            a[j][1],a[j+1][1]=a[j+1][1],a[j][1]
print(a)
found=False
c=input("input name:")
print(c)
l=0 #資料搜尋下限
u=len(a)-1 #資料搜尋上限
m=0 #每次搜尋的位置
#使用while迴圈，執行二分搜尋
while (l<=u) and not found: #只要l<=u 且還沒找到，持續搜尋
    m=(l+u)//2 #每次都是找中間值
    if c==a[m][0]:
        found=True
    else:
        if c>a[m][0]: #資料大於中間值
            l=m+1#調整下界
        else:
            u=m-1#調整上界
if found:
    print(a[m][1])
else:
    print(" not found")

```

以上循序搜尋法與二分搜尋法的差異，如表 6-7 所示。

表 6-7 循序搜尋法與二分搜尋法差異比較表

比較項目	循序搜尋法	二分搜尋法
功能比較	都是資料搜尋	都是資料搜尋
搜尋範圍	每次減少 1 筆資料	每次減少一半資料
效能比較 (以 1000 筆資料為例)	大約搜尋 500 次	不用超過 10 次即可完成搜尋

6-6 tuple

tuple 的用法和串列很接近，爲了提升電腦處理速度，如果串列的內容不會改變，即可使用 tuple 代替串列，這樣不但可以提升電腦處理速度，也可保障資料內容不被更動。tuple 的宣告方式如下：

```
a=(1,2,3)
print(a)
```

請留意是小括號，且前面串列是中括號。但是 tuple 型態取元素時，還是使用中括號。例如：

```
print(a[0])
print(a(0))#錯
```

因爲 tuple 定位在不改變元素內容，串列中只要不改變元素內容的方法，tuple 也都能用。例如：

```
a=(1,2,3)
print(len(a))
```

但是

```
a.append(4)
```

就不行了，因爲 tuple 定位在不改變元素內容。

🔧 tuple 與串列型態互轉

分別使用 `tuple()` 或 `list()`，即可互轉資料型態。例如：

```
a=[1,2,3]
b=tuple(a)#轉為tuple
print(b)#(1,2,3)
c=list(b)#轉為串列
print(c)#[1,2,3]
```

6-7 dict

若資料的處理為查表的功能，那 `dict` 可說是最完美的資料型態了。例如，會員資料的查詢，中英文翻譯等工作，這就最適合使用 `dict`。例如，若有會員資料如下：

姓名	電話號碼
aa	0900888888
bb	0900999999
cc	0939168168

以上會員資料建立如下：

```
a={"aa":"0900888888","bb":"0900999999","cc":"0939168168"}
```

那要查詢會員的電話號碼，程式如下：

```
b=a["aa"]#查詢aa的電話號碼
print(b)#0900888888
```

真是太神奇了，竟然程式只要一行。冒號前面稱為『鍵』，冒號後面稱為『值』，照規定『鍵』不能重複，但我發現若重複了，則搜尋時，是找出 `dict` 後面出現的『值』，例如，以下程式，`cc` 定義 5，又定義 6，那查詢 `cc` 時，就回應 6。

```
a={"bb":"1","bb":"2","cc":"5","bb":"3","cc":"6"}
b=a["cc"]#查詢cc的電話號碼
print(b)#6
```

▶ 宣告空白dict

宣告空白 dict 程式如下：

```
a={}
```

▶ 新增

dict 新增資料的程式如下：

```
a={"aa":10,"bb":3,"cc":5}
print(a)
a["dd"]=6
print(a)
```

```
{'aa': 10, 'bb': 3, 'cc': 5}
{'aa': 10, 'bb': 3, 'cc': 5, 'dd': 6}
```

請自行輸入以下程式，再次觀察執行結果。

```
a={}
print(a)
a["洪國勝"]='3455922'
print(a)
b='洪國勝'
print(a[b])#查詢『洪國勝』電話號碼
```

▶ 更動

dict 更動資料的程式如下：

```
a={"aa":10,"bb":3,"cc":5}
a["bb"]=9
print(a)
```

```
{'aa': 10, 'bb': 9, 'cc': 5}
```

dict 的值也可以取出來運算，例如：

```
a={"aa":10,"bb":3,"cc":5}
a['cc']=a['cc']+1
print(a['cc']) #6
```

▶ 輸出

dict 輸出資料的程式如下，要利用 items()：

```
a={"aa":10,"bb":3,"cc":5}
for a,b in a.items():
    print("%s,%d" %(a,b))
```

```
aa,10
bb,3
cc,5
```

► 值的變化

Python 充滿彈性與想像力。例如，dict 的『值』可以是『數值』，也可以是串列、tuple、dict 等。例如，以下是放串列。

<pre>a={"aa":[2,3,10,"cc"],"bb":"bb", ,"cc":5} print(a["aa"]) print(a["bb"]) print(a["cc"])</pre>	<pre>[2, 3, 10, 'cc'] bb 5</pre>
---	----------------------------------

👉 自我練習

1. 請設計一個程式，可以執行中英文名詞的翻譯。例如，輸入『book』得到『書』，輸入『書』，得到『book』。
2. 請設計一個程式，可輸入一個車牌號碼，得到客戶資料。

範例6-7a

密碼解碼（APCS107 試題）

1. 定義密碼如下：

0101	0111	0010	1101	1000	1100
A	B	C	D	E	F

2. 輸入、輸出如下：

輸入	1 0101	2 0010 1101	4 1000 1100 0101 0111	備註 輸入的值只有 以上 6 種
輸出	A	CD	EFAB	

👉 執行結果

```
input num:2
input four char:0010
input four char:1101
CD
```

👉 程式列印

```
#本例即為dict結構的應用，將資料對照表以dict結構建立，輸入資料為「鍵」，
對應輸出為「值」如a所示：
a={'0101':'A','0111':'B','0010':'C','1101':'D','1000':'E','1100':'F'}
b=int(input("input num:"))
c=''
#使用for迴圈逐一輸入字串，以輸入字串為「鍵」然後串接dict的「值」即為所求
for i in range(b):
    b=input("input four char:")
    c=c+a[b]
print(c)
```

範例6-7b

停車場計費。請寫一個程式，可以輸入車牌號碼管理與計算其停留時間，當此車牌第一次輸入時，記住其入場時間，第二次輸入時，輸出其入場時間、出場時間與停留時間。

👉 執行結果

```
input car number:1234

input car number:2345

input car number:2345
enter time is: Thu Jul 14 16:29:32 2022
leave time is: Thu Jul 14 16:29:49 2022
stay time is:16

input car number:|
```

👉 程式列印

```
#本例為查表應用範例，可建立a{"車牌":"時間"}的dict結構
import time #使用時間函式
a={} #建立空白dict a{"車牌":"時間"} 此為人類時間格式，此資料給人看
b={} #建立空白dict b{"車牌":"時間"} 此為電腦時間格式，此資料給電腦用
while True: #使用while無窮迴圈，可重複查詢，直到輸入Q 或 q
    c=input('input car number:')
    if c=='Q' or c=='q':
        break
```

```

if c in a: #如果車牌已經存在，表示此車將要離開
    ta=time.asctime() #傳回現在時間Wed Nov 6 15:57:20 2019
    ts=time.time() #傳回1970/1/1起的秒數
    print('enter time is: %s'% a[c])
    print('leave time is: %s'% ta)
    print('stay time is:%d'% (ts-b[c]))
else: #車牌不存在，表示此車正要進入
    ta=time.asctime() #傳回現在時間Wed Nov 6 15:57:20 2019
    ts=time.time() #傳回1970/1/1起的秒數
    a[c]=ta #將資料加入dict a
    b[c]=ts #將資料加入dict b

```

👉 補充說明

1. Python 已經有免費車牌辨識套件，只要安裝攝影機，就可自動解讀車牌號碼，請自行線上查詢。

👉 自我練習

1. 請寫一個門禁管理程式，只有輸入指定的帳號，才可進入，且記錄每位進入與離開時間。本例假設帳號只有 aa,bb,cc,dd 四個人。

6-8 set

set 就是一種集合運算，若您資料處理主要用途要進行集合的運算，例如，要進行交集、聯集等運算，那就採用 set 型態。例如，您一直收集客戶資料，那一定有一些客戶資料會重複，我就曾經同時接到一家科技公司的三張週年慶邀請函，這樣就窘了，有點浪費郵資，竟然自己內部資料沒整理。但若採用 set 儲存客戶資料，就可完全不用吹灰之力，電腦會自動剔除重複的資料。set 型態的建立也是大括號，例如，如以下敘述。

```
a={"aa", "bb", "cc", 1, 2, "aa"}
```

資料建立時，資料重複通通沒關係，但是電腦會自動剔除重覆資料。例如以上資料建立後，重複的資料『aa』就剔除，且將資料由小而大排序，先數值，然後字串。

```
print(a)#{1, 2, 'aa', 'bb', 'cc'}
```

► add()

增加元素。例如：

```
a={"aa","bb",1,2}
a.add(3)
print(a)#{1, 2, 3, 'aa', 'bb'}
```

但若是元素已經存在，電腦也會自動剔除。例如：

```
a={"aa","bb",1,2}
a.add(1)
print(a)#{1, 2, 'aa', 'bb'}
```

► discard()

捨棄指定元素。例如：

```
a={"aa","bb",1,2}
a.discard(1)
print(a)#{ 2, 'aa', 'bb'}
```

► pop()

取出集合的任意一個元素且刪除此元素（摘自 Python 的線上手冊），但經由實測是取出集合第 1 個元素（索引 0）且刪除該元素。例如，

```
a={"aa","bb",2,3}
print(a) #{2, 3, 'bb', 'aa'}
b=a.pop()
print(b) #2
print(a) #{3, 'bb', 'aa'}
```

⚙️ 物件轉為集合

所有的串列、tuple、dict 都可轉為 set，這樣就可以幫忙刪除重複的元素。例如：

```
a=[1,2,3,1,3]
print(a) #1,2,3,1,3
b=set(a)
```

```
a=list(b)
print(a)#1,2,3
```

⚙️ 集合的運算

集合的運算有交集、聯集、差集、屬於、包含於，這些定義與符號通通與數學課本的內容相同。例如：

```
a={1,2,3,4}
b={3,4,5}
c=a & b #交集
print (c) #3,4
c=a |b #聯集
print (c)#1,2,3,4,5
c=a-b #差集
print(c)#1,2
c=a ^ b #對稱差集
print (c)#1,2,5
print (1 in a)#屬於 True
c={1,2}
print(c < a)#包含於 True
```

▶ frozenset()

前面當要保障串列的內容不被更動，那可以改用 tuple，這樣還可提升執行速度。set 也是同樣的道理，若 set 的內容不用更動，也可使用 frozenset，這樣也可提升資料處理速度，且保障資料不被更動。例如：

```
a=frozenset({1,2,3})
b=frozenset({3,4})
```

以上就稱為不可變動的 set，也可套用所有 set 的運算。例如：

```
print(a-b)
```

其次 frozenset 既然不可更動，那當然無法再新增元素，或使用會變動集合的方法。

範例6-8a

孤獨字串 (APCS107/10 月考題)

當一個字串，使用字母 (A ~ Z) 的個數最少，則他就是一個孤獨字串。例如：

```
"HELLO"
```

字串長度 5，但是使用四個字母組成，但是

```
"ABBBAA"
```

字串長度 6，僅使用 2 個字母組成，那就比上一個字串『HELLO』孤獨，本例請於所輸入字串中，找出最孤獨字串，也就是含字母種類最少者。其次，若孤獨字串有兩個以上，請繼續將這些孤獨字串按照字母順序排列，找出排在最前面的字串。例如，『ABBBAA』與『AAAABB』均謹含兩個字母，那還要繼續比較，找出字母順序最前面的字串，所以是『AAAABB』，以下是參考的測資：

```
5
HELLO
ABBBAA
AAAABB
GOODBYE
ABCDEFGHIJKL
```

表示 5 個測資，則輸出

```
AAAABB
```

👉 執行結果

```
[{'L', 'E', 'O', 'H'}, {'A', 'B'}, {'A', 'B'}, {'D', 'B',
'O', 'Y', 'G', 'E'}, {'D', 'I', 'J', 'B', 'F', 'H', 'L',
'G', 'E', 'A', 'K', 'C'}]
[4, 2, 2, 6, 12]
2
[1, 2]
2
AAAABB
```

👉 程式列印

```
#本例要去掉重複的資料，此即為集合運算，可運用set結構，就能省事
a=['HELLO','ABBBAA','AAAABB','GOODBYE','ABCDEFGHijkl']
#a=['HELLO','ABBBAA','AAAABC','GOODBYE','ABCDEFGHijkl']
b=[set(a1) for a1 in a] #逐一轉為集合
print(b)
#[{'E', 'O', 'L', 'H'}, {'A', 'B'}, {'A', 'B'}, {'B', 'Y', 'D',
'E', 'G', 'O'}, {'B', 'K', 'C', 'D', 'I', 'E', 'G', 'J', 'F',
'H', 'A', 'L'}]
#轉為集合，相同元素就自動刪除
c=[len(b1) for b1 in b] #逐一求出每個集合長度
print(c)#[4,2,2,6,12] #逐一輸出每個集合長度，確認是否正確
d=min(c) #集合長度最小值，此即為使用最少字元
print(d)#2
e=[i for i in range(len(c)) if c[i]==d] #找出極小值的索引
print(e)#[1,2] #索引1,2都是僅含2個字元
f=len(e) #此即為稀疏字串個數
print(f)#2
if f==1: #檢查極小值是否只有1個
    #極小值只有一個，即為所求
    print(a[e[0]])
else:
    #極小值有兩個以上，繼續再求ASCII最小者
    min=a[e[0]] #預設極小值為第0個
    #使用迴圈找出極小值
    for i in range(1,f):
        if a[e[i]]< min:
            min=a[e[i]]
print(min)
```

範例6-8b

卡通部隊 (APCS 108/06 月試題)

給定卡通部隊成員個數與隊伍組數。例如，以下代表有 10 個成員，6 個隊伍，10 個成員分別是 ABCDEFGHIJ

```
10 6
AJBA
HCEFGGC
```

```

BIJDAIJ
EFCDHGI
HCEFGA
JAJB

```

若任兩組沒有相同成員，且含全部成員，那稱此兩對互補，請找出以上有幾組互補成員。但是 AKB,KAKB,BAK 視為相同隊伍。例如，本例 1,4 互補，2,3 也互補，4,6 也互補所以答案是 3 組。

👉 執行結果

```

0 3
1 2
3 5
3

```

👉 程式列印

```

m=10 #成員數量
n=6 #隊伍數量
#使用串列儲存6個隊伍的成員
f=['AJBA', 'HCEFGGC', 'BIJDAIJ', 'EFCDHGI', 'HCEFGA', 'JAJB']
e=len(f)
num=0
#所有隊伍兩兩比較，找出沒有交集，且包含全部成員
#先列出比較方式，再依照其規律性，使用迴圈表示
#(1,2) (1,3) (1,4) (1,5) (1,6)
#(2,3) (2,4) (2,5) (2,6)
#(3,4) (3,5) (3,6)
#(4,5) (4,6)
#(5,6)
#由以上規律性，以雙迴圈表示如下
for i in range(e-1):
    for j in range(i+1,e):
        #print(i,j)
        #以下將串列轉為集合，即可自動刪除重複成員
        #再使用集合的交集與聯集運算子
        if (len(set(f[i]) & set(f[j]))==0 and len(set(f[i]) |
                                                    set(f[j]))==m):
            print(i,j) #沒有交集，且包含全部成員
            num=num+1
print(num)

```

範例6-8c

完美彩帶。(APCS 108/06 試題)

給定顏色個數 m 與彩帶長度 n ，例如：

```
4 10 #4代表有4種不同顏色，10代表有10種測資
1 7 6 4 4 6 1 7 4 7
```

又例如：

```
3 7
1 2 2 0 1 2 1
```

若有連續 m 個元素同時包含 m 個顏色，則稱此段為完美彩帶。例如第 1 筆測資的 1 7 6 4、4 6 1 7 與 6 1 7 4 都是連續 4 個元素，且這 4 個元素就包含四種顏色，所以通通是完美彩帶，所以答案是 3。第 2 筆測資的 2 0 1 與 0 1 2 也是完美彩帶，所以答案是 2。

執行結果

```
[1, 7, 6, 4]
[7, 6, 4, 4]
[6, 4, 4, 6]
[4, 4, 6, 1]
[4, 6, 1, 7]
[6, 1, 7, 4]
[1, 7, 4, 7]
3
```

程式列印

```
#將資料以串列儲存
#第1筆測資
a=[1,7,6,4,4,6,1,7,4,7]
m=4 #4代表有4種不同顏色
n=10 #10代表有10筆測資
#第2筆測資
#a=[1,2,2,0,1,2,1]
#m=3 #3代表有3種不同顏色
#n=7 #7代表有7筆測資
num=0
```

```
#使用for迴圈，逐一於a串列，每次取m個元素
for i in range(n-m+1):
    b=a[i:i+m] #每次取m個元素
    print(b) #印出來用肉眼檢查
    b=set(b) #轉為集合
    if len(b)==m: #若有m個顏色，表示此為完美彩帶
        num=num+1
print(num)
```

由以上參考解答可知，Python 語法較寬鬆，功能也較強，同樣問題所需程式比 C/C++ 簡化許多。

6-7 本章內容摘要

1. 一維串列的常用宣告語法如下：

```
串列名稱=[初值]*串列長度
```

例如：以下可宣告一個一維串列，串列長度是 5，可用索引是 a[0],a[1],a[2],a[3],a[4]。

```
a=[0]*5
print(a)
```

2. 二維串列的宣告常用語法如下：

```
串列名稱=[[初值 for i in range (行數)]for j in range (列數)]
```

例如：以下程式可快速初始化一個三行四列的二維串列，其初值為 0。

```
a=[[0 for i in range (3)]for j in range (4)]
print(a)
```

3. 以上程式亦可簡化如下：

```
b=[[0]*3 for i in range(4)]
```

4. 善用串列函式可減少程式設計師負擔，降低程式撰寫門檻，串列常用函式有 `sum()`、`max()`、`min()`、`sort()`、`sorted()`。
5. 善用串列 `append()`、`inserted()`、`pop()`、`remove()` 等函式，可快速增加、插入、提取、刪除指定元素，簡化程式的撰寫。
6. 善用串列生程式，可減少程式所需要的長度。例如：以下程式可逐一將 `a` 串列值所有元素乘以 2，且複製到另一串列。

```
a=[2,3]
b=[a1*2 for a1 in a]#以上兩個a1，變數名稱要相同，且遵守變數命名規則
print(b)
```

7. 以下是氣泡排序的樣版程式。

```
a=[5,4,3,2,1]#要用最糟的初值
n=len(a)
for i in range(0,n-1):
    for j in range(0,n-i-1):
        if a[j]>a[j+1]:
            a[j],a[j+1]=a[j+1],a[j]
print(a)
```

8. 以下是計數排序的樣版程式。

```
a=[8,2,9,7,1] #每一個人的成績
n=len(a)
b=[1 for a1 in a] #先指派每個人都是第1名
for i in range(n-1):
    for j in range(i+1,n):
        if a[i]>a[j]:
            b[j]=b[j]+1
        else:
            b[i]=b[i]+1
print(a) #成績
print(b) #名次
```

9. dict 結構最適合查表的場合。例如會員姓名與電話可建立 dict 結構如下：

```
a={"aa":"0900888888","bb":"0900999999","cc":"0939168168"}
```

以下程式，即可查詢會員的電話號碼。

```
b=a["aa"]#查詢aa的電話號碼
print(b)#0900888888
```

10. set 最適合集合運算。集合的宣告如下：

```
a={"aa","bb","cc",1,2,"aa"}
print(a)
```

電腦會自動剔除重複資料，且集合的所有運算，如交集 (&)、聯集 (|)、屬於 (in)、包含於 (<) 等都有運算子。

課後評量

(填充題，請寫出程式執行結果)

題號	題目	執行結果
1	<pre>a=[i for i in range(3)] print(a)</pre>	
2	<pre>a=[2,3,4] print(a) print(a[1])</pre>	
3	<pre>a=[2,3,4] b=a c=a[:] b[1]=5 print(a[1]) c[1]=6 print(a[1])</pre>	
4	<pre>a=[[0,1,2],[3,4,5]] print(a[0][1])</pre>	

5	<pre>a=[[0,1,2],[3,4,5]] print(a[0][1]) print(a[1]) print(max(a[0])) print(a) print(max(a)) print(sum(a[0]))</pre>	
6	<pre>a=[0,1,2] b=[3,4,5] c=a+b print(c) d=b+a print(d) a=a*2 print(a)</pre>	
7	<pre>a=[0,1,2] a.append(3) print(a)</pre>	
8	<pre>a=[0,1,2] a.append(3) print(a) a.insert(1,4) print(a) a.remove(2) print(a) a.pop(1) print(a) a.pop() print(a)</pre>	
9	<pre>a=[3,7,2,8] a.sort() print(a) print(a.sort()) b=sorted(a,reverse=True) print(b) b=a.sort() print(b)</pre>	

10	<pre>a=[3,7,2,8] print(len(a)) print(3 in a) print(1 not in a) print(a.count(0)) print(a.index(7))</pre>	
11	<p>以下程式是否正確?</p> <pre>a=(1,2,3) a[1]=5 print(a[1])</pre>	
12	<pre>a=(1,2,3) b=list(a) b.append(4) c=tuple(b) print(c)</pre>	
13	<pre>a={"aa": "1", "bb": "2", "cc": "5"} b=a["cc"] print(b) a["dd"]=6 b=a["dd"] print(b)</pre>	
14	<p>以下程式是否正確?</p> <pre>a={"aa": "1", "bb": "2", "cc": "5"} b=a["5"] print(b)</pre>	
15	<pre>a={"aa":10, "bb":3, "cc":5} a['cc']=a['cc']+3 print(a['cc']) a["cc"]=9 print(a['cc'])</pre>	
16	<pre>a={"aa":10, "bb":3, "cc":5} for a,b in a.items(): print("%s,%d" % (a,b))</pre>	

17	<pre> a={0,1,2,3,0} print(a) a.add(1) print(a) a.discard(2) print(a) b=a.pop() print(b) print(a) </pre>	
18	<pre> a=[0,1,2,0,2] b=set(a) print(b) c=list(b) print(c) </pre>	
19	<pre> a="0123012" b=set(a) print(b) c="".join(b)#轉字串 print(c) </pre>	
20	<pre> a="0123012" b=set(a) c=list(b) print(c) d="".join(c)#轉字串 print(d) </pre>	
21	<pre> a={1,2,3,4} b={3,4,5} c=a & b #交集 print(c) #3,4 c=a b #聯集 print(c) #1,2,3,4,5 c=a-b #差集 print(c) #1,2 c=a ^ b #對稱差集 print(c) #1,2,5 print(1 in a)#屬於 True c={1,2} print(c < a)#包含於 True </pre>	

22	<pre>a={0,1,2,3,0} b=frozenset(a) print(b)</pre>	
23	<pre>a=frozenset({1,2,3}) b=frozenset({3,4}) print(a-b)</pre>	

課後習題

- 假設資料如下：
3,7,2,1,5
請寫一個程式，可以輸入一個數字，並判斷此數字是否在以上資料內。例如，輸入 1，得到『是』，輸入『8』，得到『否』。
- 設有資料如下：
3,2,1,9,6,7,3,1,4
請寫一個程式，可以輸入一個整數，並統計其出現次數。例如，輸入 3，得到 2，輸入 7 得到 1
- 請寫一個程式，判斷串列資料是否全為正數。例如，3,6,2,1,7 傳為『是』，2,-3,5,1,7,8 傳回『否』。
- 請寫一個程式，判斷串列資料是否全為遞增數。例如，3,6,8,11,17 傳為『是』，3,9,5,11,17,18 傳回『否』。
- 樂透對獎。假設樂透號碼是 1 到 42 號球，假設本期樂透開獎號碼是 2,3,8,14,22,39，特別號是 29，使用者可輸入 6 個號碼，電腦回應中獎情形。請選擇適當資料結構儲存以上資料，且寫一程式，可以協助使用者對獎。
- 樂透對獎。假設以下是張三已經購買的樂透彩券號碼。

編號	號碼 1	號碼 2	號碼 3	號碼 4	號碼 5	號碼 6
1	12	13	14	15	16	22
2	12	13	14	15	16	32
3	12	13	14	15	22	16
4	2	3	12	13	14	15
5	2	3	4	12	13	14
6	1	4	8	12	15	39

- (1) 請以適當資料結構儲存。
- (2) 請寫一程式，可以輸入樂透中獎號碼(本例假設中獎號碼是 22,12,13,14,15,16,17(特別號)，並核對中獎情形。
7. 同上題，但是自己開獎(自動產生 7 個不重複亂數)，並比對中獎情形。
8. 請寫一個程式，可以指派參加抽獎的人數與獎品數量，然後就可以得到得獎的編號。例如，本次收到 42 個來信，要抽出 5 位得獎編號，程式一執行，可以得知中獎的編號。
9. 假設有 10 個人要摸彩，獎品 4 個，有「電視、電腦、筆電、手機」，請寫一個程式，可以讓每個人都享受摸一次獎的樂趣。
10. 請寫一程式，滿足以下條件。
- (1) 某次考試共 10 題，假設正確答案如下：

1, 4, 2, 4, 2, 3, 3, 3, 1, 1

- (2) 某班共 5 人，某次考試共 10 題，每人答案如下，請寫程式評定其成績。

	1	2	3	4	5	6	7	8	9	10
1	1	2	4	2	3	3	3	3	1	1
2	1	3	2	2	2	2	2	2	2	1
3	1	4	3	4	3	3	3	3	1	1
4	1	3	2	2	2	2	2	1	1	1
5	1	2	1	1	1	1	1	1	1	1

11. 音階校正。(APCS 試題) 給定若干音階，例如

8 (8 個數字)

1 7 8 8 9 16 17 26

從第 4 個起，若

$$|y_i - y_{i-1}| > 5$$

則將此音修正為前 3 音的中位數。例如，以上音符修正如後如下：

1 7 8 8 9 8 8 8

(提示：任意數量的數字，將數字由小而大排序，若數字數量是奇數，則定義中間的數字是中位數，例如 3,5,4 先排序為 3,4,5 則中位數是 4；若數字個數是偶數，則中間兩個數字平均定義為中位數，例如四個數字是 1,9,3,7，先排序為 1,3,7,9，取中間兩個平均是 5，所以中位數是 5。)

12. 請寫一個程式，可以將二進位轉為十六進位。測試資料如下：

編號	輸入	輸出
1	11001000	0xc8
2	1100011110	0x31e
3	11010	0x1a

提示：

方法一：先轉為十進位，再轉十六進位。

方法二：

1. 以 dict 建立二進位與十六進位對照表（二進位取 4 碼剛好可對應 1 個十六進位）。
2. 將每筆資料長度調整為 4 的倍數，不足 4 位者前面補 0。
3. 逐一取 4 碼二進位轉為十六進位。
4. 串接轉換而得的十六進位。

函式應用

學習綱要

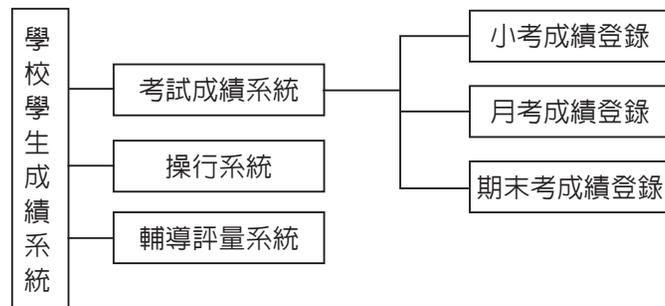
- ≡ 7-1 模組
- ≡ 7-2 內建函式庫的認識與應用
- ≡ 7-3 自訂函式
- ≡ 7-4 全域變數與區域變數
- ≡ 7-5 函式應用實作演練
- ≡ 7-6 本章內容摘要

學習目標

- ≡ 1. 能瞭解模組化程式設計的意義。
- ≡ 2. 能瞭解與認識內建函式庫。
- ≡ 3. 能熟練函式的宣告與定義。
- ≡ 4. 能熟練函數引數的傳遞。
- ≡ 5. 能熟練函式傳回值的程式語法。
- ≡ 6. 能熟練全域變數與區域變數。
- ≡ 7. 能實作函數應用實例。

7-1 模組 (Module)

工業化的產品有所謂模組化設計，例如，一部電腦分成主機、螢幕、硬碟、鍵盤、滑鼠等模組，產品的設計與維修都是模組化，彼此獨立，也就是鍵盤壞了，就換鍵盤，滑鼠壞了就換滑鼠。軟體工業是發展較晚的新興產業，也是遵循此道理，希望將問題模組化，每個模組也和硬體系統一樣，是可組合、可分解和可更換的單元。模組化程式設計是指解決一個複雜問題時由上而下逐層把軟體系統劃分成若干模組。每個模組完成一個特定的子功能，所有的模組的功能與介面要先定義，並可按某種方法組裝起來，完成整個系統所要求的功能。例如，要撰寫學校學生成績管理系統，就可先分成三個模組，分別是教務處的考試成績模組、學務處的操行系統模組、輔導處的輔導評量系統模組，這些子系統都可完成特定功能，且資料交換介面也已經定義且公開。教務處的考試成績系統模組則再分為小考成績登錄模組、月考成績登錄模組、期末考成績登錄模組等系統，如圖 7-1 所示。



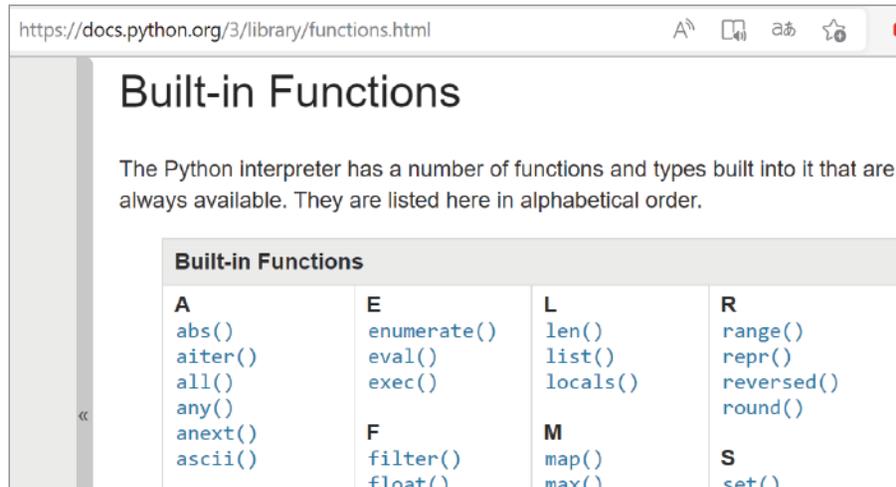
★ 圖 7-1 學校學生成績系統示意圖

7-2 內建函式庫的認識與應用

Python 內建函式庫分為內建函式與模組，分別說明如下：

⚙️ 內建函式 (Built-in Function)

Python 雖然是以物件導向語言為基礎所發展的語言，所有函式庫通通以模組分類存放，但仍保留使用者舊有習慣，將最常用的功能以函式存在，Python 內建公用函式如圖 7-2。



★ 圖 7-2 Python 內建函式圖

茲將以上內建函式抽取高職常用函式，介紹如下：

► str()

將數值參數轉為字串。例如：

```
a=123
b=456
print(a+b)
```

此為數值相加，結果是 579，若要將這些數字連結，則要先轉為字串，再相加，程式如下：

```
print(str(a)+str(b)) #字串相加123456
```

► int(x)

將字串參數轉為數值。例如：

```
a="123"
b="456"
print(a+b) #字串相加123456
print(int(a)+int(b)) #579 數值相加
```

▶ ord(x)

傳回字元參數的二進位編碼。例如：

```
print(ord("A"))#65
```

每一個數字、符號、大小寫英文字元，都要數位化，才能用電腦儲存，這些常用字元沒有超過 127 個，所以使用 7 位元編碼，例如，字元 A 編碼為「01000001」=65，叫到 65 號就是 A；字元 B 的編碼是「01000010」=66，叫到 66 號，就是” B”；字元 1 的編碼是「00110001」=49，叫到 49 號就是 1，此稱為 ASCII 編碼。

▶ chr(x)

傳回數值參數所對應的字元。例如：

```
a=65  
print(chr(a))#A
```

▶ abs(x)

傳回數值參數所對應的絕對值。例如：

```
a=-3  
print(abs(a)) #3
```

▶ pow(a,b)

傳回 a^b 的值。例如：

```
print(pow(2,3))#8
```

▶ len(a)

傳回物件 a 的長度。例如：

```
a=[3,2,1]  
print(len(a))#3  
b="abcd"  
print(len(b))#4
```

▶ round(a,b)

將 a 依照 b 所指定位數四捨五入，b 可為正數，也可為負數，正數代表取小數幾位，負數代表取對應的個、十、百、千位。例如：

```
print(round(255.555,2))#255.56 小數取2位，小數第3位四捨五入
print(round(255.555,1))#255.6 小數取1位，小數第2位四捨五入
print(round(255.555,0))#256 取到整數，小數第1位四捨五入
print(round(255.555,-1))#260 取到十位數，個位數四捨五入
print(round(255.555,-2))#300 取到百位數，十位數四捨五入
```

▶ divmod(a,b)

傳回 a//b 與 a%b 之值，且其資料型態是 tuple。例如：

```
a=divmod(10,3)
print(a)#(3,1)
print(type(a))#<class 'tuple'
print(a[0])#3
print(a[1])#1
```

▶ sum(x)

計算串列 x 的總和。例如：

```
a=[3,2,1]
print(sum(a))
```

▶ max(x)、min(x)

傳回參數 x 的極大或極小值，參式 x 可為串列或不定數量的數值。例如：

```
a=[3,2,1];b=6;c=3;d=9
print(max(a))#3
print(max(b,c))#6
print(min(b,c,d))#3
```

▶ dir()

傳回物件所對應的可用方法。例如：

```
a=[3,2]
print(dir(a))
```

結果如圖 7-3。

```
[ '_add_', '_class_', '_contains_', '_delattr_', '_delitem_',
  '_dir_', '_doc_', '_eq_', '_format_', '_ge_',
  '_getattr_', '_getitem_', '_gt_', '_hash_', '_iadd_',
  '_imul_', '_init_', '_init_subclass_', '_iter_', '_le_',
  '_len_', '_lt_', '_mul_', '_ne_', '_new_', '_reduce_',
  '_reduce_ex_', '_repr_', '_reversed_', '_rmul_', '_setattr_',
  '_setitem_', '_sizeof_', '_str_', '_subclasshook_', '_append_',
  '_clear_', '_copy_', '_count_', '_extend_', '_index_', '_insert_', '_pop_', '_remove_',
  '_reverse_', '_sort_' ]
```

★ 圖 7-3 串列可用方法

► eval()

傳回算術運算式的運算結果。例如：

```
print(eval("3+2*4"))#11
```

這也很神，例如，用來製作計算器就很方便，只要讓使用者按完一個運算式，Python 就幫您自動計算結果。

⚙️ 模組

物件導向則是依照函式功能分類，寫在一個資料夾，就稱為一個模組。下圖則是 Python 的內建函式庫 (點選『開始 /Python/Python Module Docs』)，所有方法已經依照模組分類存放，如圖 7-4 所示：

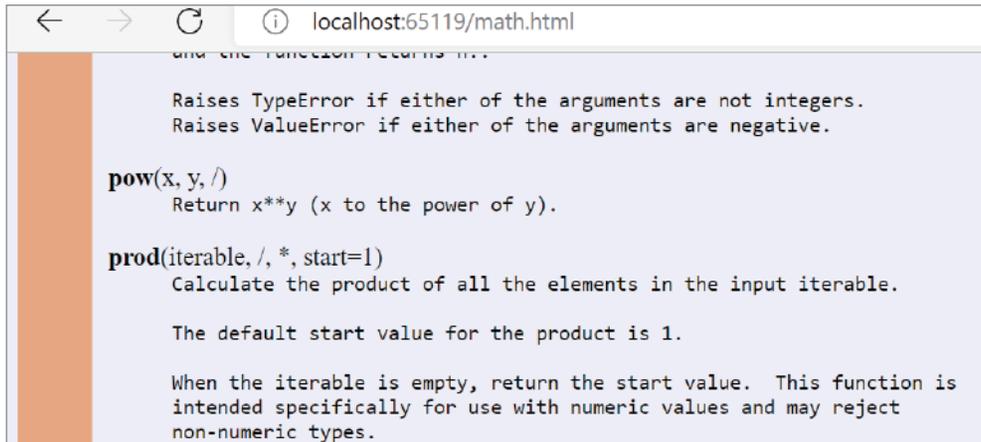


★ 圖 7-4 Python 內建模組

以上模組相當豐富與完整，以下我們摘錄一些高職生常用模組。熟悉這些模組以後，自然可自行探索其餘模組。

► math

開啓 math 模組，如圖 7-5 所示：



★ 圖 7-5 math 模組可用方法

裡面有很多數值運算的方法。使用前應先載入此模組，例如：

```
import math
print(math.fabs(-3.14))#3.14
print(math.pow(2.1,3.1))#9.974
```

有時候模組名稱很長，此時也可將模組取一個別名，然後使用「別名.方法」撰寫程式。例如：

```
import math as m
print(m.fabs(-3.14))#3.14
print(m.pow(2.1,3.1))#9.974
```

► random

random 模組有一些隨機亂數的方法，例如，之前我們已經介紹產生亂數的方法 randint()，程式如下：

```
import random
a=random.randint(1,6)
print(a)
```

以下再介紹一些 random 模組常用方法。

▶ choice(串列)

於串列中，任選一個資料傳回。例如，先將獎品放在串列中，以下程式就可幫我們隨機抽選一個獎品。

```
import random
a=['電視','冰箱','手機','銘謝惠顧']
b=random.choice(a)
print(b)
```

▶ sample()

剛剛 choice() 是選一個，但是樂透開獎卻是選指定個數的球，sample() 就可以傳回指定個數的球數。例如，以下程式可從 1 ~ 42 號球任意取出 7 球，我們假設最後一個球就是特別號。

```
import random
a=random.sample(range(1,43),7)
print(a)
b=a.pop()
print('special number is %d'%b)
```

或是於 42 個人抽出 7 人，程式也是同上。

▶ time

time 模組內有一些關於時間的方法，以下篩選常用方法如下：

▶ localtime()

以 tuple 的方式傳回時間的個別資料。例如：

```
import time #載入時間模組
a=time.localtime()
print(a[0])#年
print(a[1]) #月
print(a[2]) #日
print(a[3]) #時
print(a[4]) #分
print(a[5]) #秒
```

```

print(a[6]) #星期幾
print(a[7]) #該年第幾天
print(a[8]) #是否夏令時間
print(time.asctime())#傳回現在時間Wed Nov 6 15:57:20 2022
print(time.time())#傳回1970/1/1起的秒數

```

► sleep(n)

強制讓電腦暫停 n 秒。例如，前面我們已經使用 sleep(1) 讓電腦暫停 1 秒。這樣就可簡單完成時鐘程式。

```

import time
t=0
while (1):
    t=(t+1)%(24*60*60)
    h=t//3600
    m=(t-h*3600)//60
    s=t %60
    print('%2d:%2d:%2d'% (h,m,s))
    time.sleep(1)#暫停1秒

```

⚙️ twstock 模組

因為 Python 是免費開源軟體（英語：open source software，縮寫：OSS）又稱開放原始碼軟體，也就是原始碼可以任意取用的電腦軟體，這種軟體的著作權持有人允許使用者學習、修改且散播自己創作的軟體。所以熱心的人士與協力廠商就源源不絕開發能相容 Python 的實用模組。以股市分析程式為例（可能是股票買賣公司，為了鼓勵大家投資股市，就幫忙寫一些股票看盤套件），例如，以下程式，就可輸出近三十日『大立光』收盤價。

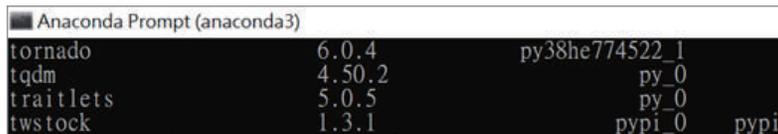
```

import twstock as a
b=a.Stock("3008")#3008大立光股票代碼
print(b.price)#近三十日收盤價

```

👉 補充說明

1. 本書推薦與介紹使用 Anaconda 作為 Python 整合開發環境，主要是 Anaconda 已經蒐集使用者常用模組且安裝。例如：以上 twstock 模組，雖然是第三方模組，但因為熱門且常用，所以安裝完 Anaconda 就可以看到 twstock 模組，如圖 7-6：



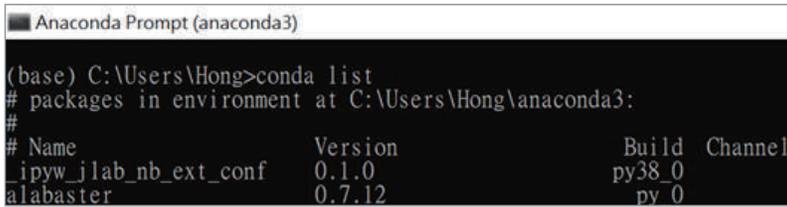
```

Anaconda Prompt (anaconda3)
tornado 6.0.4 py38he774522_1
tqdm 4.50.2 py_0
traitlets 5.0.5 py_0
twstock 1.3.1 pypi_0 pypi

```

★ 圖 7-6 Anaconda 所安裝模組

2. 開啓「Anaconda Prompt」(點選功能 Window/開始/Anaconda/Anaconda Prompt)，並鍵入「conda list」，如圖 7-7，再往下捲動，即可見圖 7-6。



```

Anaconda Prompt (anaconda3)
(base) C:\Users\Hong>conda list
# packages in environment at C:\Users\Hong\anaconda3:
#
# Name          Version      Build      Channel
ipyw_jlab_nb_ext_conf 0.1.0        py38_0
alabaster       0.7.12       py_0

```

★ 圖 7-7 Anaconda Prompt 視窗

3. 若所需模組未出現，請於圖 7-7 Anaconda Prompt 視窗，鍵入「conda install 模組名稱」即可安裝該模組。

⚙️ numpy 模組

numpy 模組存在很多解國高中數學的方法，這些方法可以幫助理解或解決國中、高中的數學問題。

▶ roots() 方法

解一元多次方程式是使用 roots() 方法，可解一元一次、一元二次、一元三次等方程式。例如，若有一元一次方程式如下：

$$x+2=0$$

則求其解的程式如下：

```
import numpy as np
b=np.roots([1,2])#x+2=0
print(b)#-2
```

又例如，一元二次方程式如下：

$$x^2-4x-12=0$$

求其解的程式如下：

```
import numpy as np
b=np.roots([1,-4,-12])#x**2-4x-12=0
print(b)#6,-2
```

又例如，一元三次方程式如下：(一元三次方程式是一元二次方程式的推廣，高職商管數學無介紹，僅供參考)

$$x^3-3x^2-16x-12=0$$

求其解的程式如下：

```
import numpy as np
b=np.roots([1,-3,-16,-12])#x**3-3x**2-16x-12=0
print(b)#6,-2,-1
```

► linalg.solve()方法

二元一次或三元一次要使用 linalg 模組的 solve 方法。例如，若有二元一次方程式如下：

$$\begin{aligned} 3x+y &= 5 \\ x-2y &= -3 \end{aligned}$$

則求其解的 Python 程式如下：

```
import numpy as np
c=np.array([[3,1],[1,-2]])
d=np.array([5,-3])
```

```
ans=np.linalg.solve(c,d)
print(ans)#1,2
```

又例如，三元一次方程式如下：(三元一次方程式是二元一次方程式的推廣，高職商管數學無介紹，僅供參考)

```
x+y-z=-2
x+z=2
x-y+2z=5
```

則求其解的 Python 程式如下：

```
import numpy as np
c=np.array([[1,1,-1],[1,0,1],[1,-1,2]])
d=np.array([-2,2,5])
ans=np.linalg.solve(c,d)
print(ans)#1,-2,1
```

⚙️ matplotlib.pyplot 模組

國高中數學有很多數學函數，本節就使用 matplotlib.pyplot 繪圖模組來繪製這些函數圖形，尤其是高一數學的指數、對數、三角函數，您只要多畫幾次，就如同圖像式學習，那就會更瞭解這些數學函數的內涵。其次，matplotlib.pyplot 繪圖模組還要使用 numpy 模組，來存放所需要的連續數據。(補充說明：若使用 Anaconda3，那以上模組都是預設模組，通通不用再額外安裝)

▶ 直線

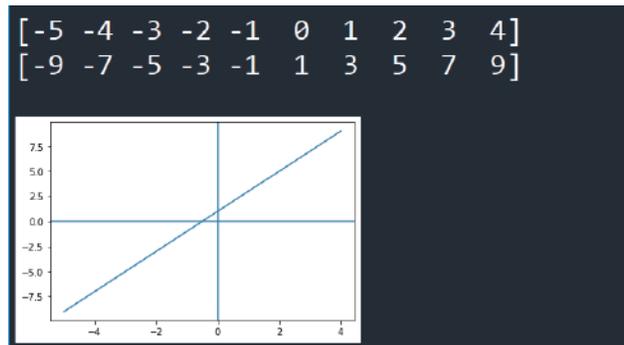
我們人類繪製直線是使用直尺，但電腦並沒有直尺，電腦如何繪製直線呢？答案就是要先建立直線方程式，然後再根據直線方程式一點一點密集繪製，這些密集的点，看起來就是直線。例如，設有直線方程式如下：

```
y=ax+b
```

要繪製 x 從 -5 到 5 的直線，程式如下：

```
import matplotlib.pyplot as plt
import numpy as np
x = np.arange(-5, 5, 1) #-5開始到5，間隔是1，不含結束點5，間隔可
                        以實數，且x為串列
print(x) #[-5 -4 -3 -2 -1 0 1 2 3 4]
y = 2*x + 1 # y為串列
print(y) #[-9 -7 -5 -3 -1 1 3 5 7 9]
plt.plot(x,y) #使用直線連接以上座標
plt.axhline(y=0) #繪出x軸
plt.axvline(x=0) #繪出y軸
plt.show() #於螢幕輸出結果
```

執行結果如圖 7-8：



★ 圖 7-8 直線圖

`arange()` 方法是產生一個 list (串列)，本例從 -5 開始到 5，間隔是 1，不含結束點 5，且間隔可以實數。請留意圖 7-8 第一、第二列數字是以下程式的輸出結果。

```
x = np.arange(-5, 5, 1) #-5開始到5，間隔是1，不含結束點5，且x為串列
print(x) #[-5 -4 -3 -2 -1 0 1 2 3 4]
y = 2*x + 1 # y為串列
print(y) #[-9 -7 -5 -3 -1 1 3 5 7 9]
```

👉 自我練習

1. 請分別繪製以下聯立直線方程式。

(1) $3x+y=4$ $6x+2y=8$	(2) $3x+y=4$ $3x+y=8$	(3) $3x+y=4$ $2x-y=1$
---------------------------	--------------------------	--------------------------

▶ 二次曲線

前面 x 僅一次方，圖形是直線，若 x 是兩次、或三次，就形成曲線， x 二次會是拋物線，二次方係數為正就凹向上，會有極小值；二次方係數為負就凹向下，會有極大值，請觀察以下圖形。

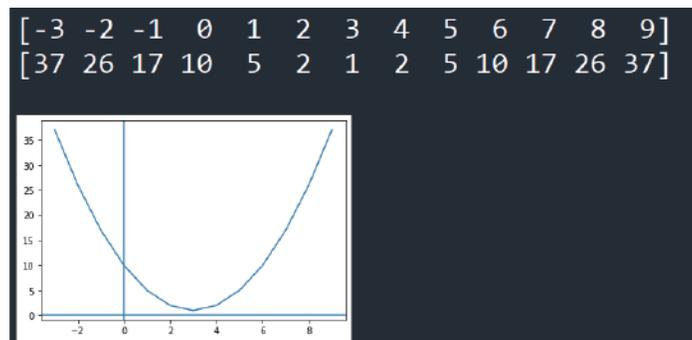
$$y = (x-3)^2 + 1 \quad (x=3 \text{ 有極小值} 1)$$

$$= x^2 - 6x + 10$$

繪製以上圖形的程式如下：

```
import matplotlib.pyplot as plt
import numpy as np
x = np.arange(-3, 10, 1) #數字是我多次修正的結果，這樣可以表現圖形
                           最有變化的部分
print(x)
y = x * x - 6 * x + 10
print(y)
plt.plot(x, y)
plt.axhline(y=0) #x軸
plt.axvline(x=0) #y軸
plt.show()
```

以上程式執行結果如圖 7-9：(x 從 -3 到 10，這是我慢慢觀察圖形，所調整出來的範圍，這樣才能畫出函數最有變化的範圍)



★ 圖 7-9 一元二次方程式圖形

以下是一元三次方圖形（一元三次方程式是一元二次方程式的推廣，超出高職商科範圍，僅供參考），一元三次方圖形大部分有兩個臨界點，或稱相對極值點。例如：

$$y=2x^3-3x^2-12x+3$$

將以上對 x 微分，得到斜率方程式如下：

$$dy/dx=6x^2-6x-12$$

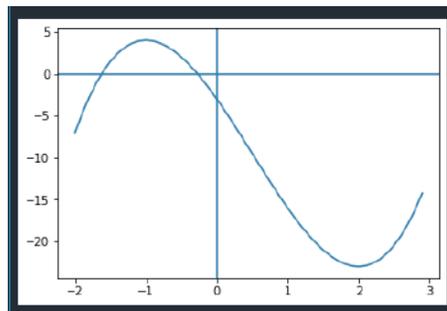
設其斜率為零如下：

$$dy/dx=6x^2-6x-12=0$$

解出 $x=-1$ 或 $x=2$ 時斜率為零，此點即為臨界點，撰寫程式驗證如下：

```
import matplotlib.pyplot as plt
import numpy as np
x = np.arange(-2, 3, 0.1) # 這是我慢慢觀察圖形，所調整出來的範圍
y=2*x**3-3*x**2-12*x-3
plt.plot(x,y)
plt.axhline(y=0) #x軸
plt.axvline(x=0) #y軸
plt.show()
```

執行結果如圖 7-10：



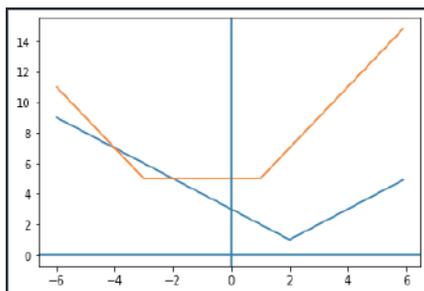
★ 圖 7-10 一元三次方程式圖形

▶ 絕對值函數

方程式有一個絕對值，就會有一個轉折點，請看以下程式。

```
import matplotlib.pyplot as plt
import numpy as np
x = np.arange(-6, 6, 0.1)
y1=abs(x-2)+1#一個折點
y2=abs(x+3)+abs(x-1)+1#兩個折點
plt.plot(x,y1)
plt.plot(x,y2)
plt.axhline(y=0)#x軸
plt.axvline(x=0)#y軸
plt.show()
```

以上執行結果如圖 7-11：



★ 圖 7-11 絕對值曲線圖

▶ 指數函數

x 是指數，例如：

$$y=a^x \quad a>0, a<>1, x \text{ 可為任意實數}$$

稱為指數函數，此為高一數學。此一函數的第一個重點是，要討論定義域與值域，所有 x 值所成的集合稱為定義域，其值可從負無限大到正無限大。第二個重點是，所有 y 值所成的集合稱為值域，值域僅會大於 0，驗證程式如下：

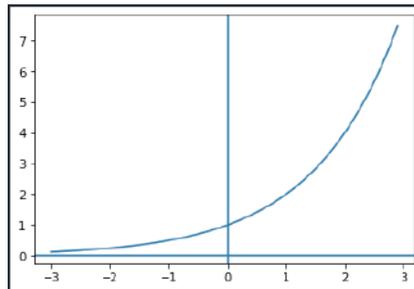
```
import matplotlib.pyplot as plt
import numpy as np
x = np.arange(-3, 3, 0.1) #x從-3~3，每次遞增0.1，不含結束點3
print(x) #-3,-2.9,-2.8...0...2.8,2.9
```

```

y=2**x #**次方運算子
plt.plot(x,y) #以x,y串列為資料，繪出圖形
plt.axhline(y=0) #繪出x軸
plt.axvline(x=0) #繪出y軸
plt.show() #於螢幕顯示此圖形

```

執行結果如圖 7-12 所示，請留意 $x=0$ 時 y 一定等於 1。



✦ 圖 7-12 指數函數圖

► 三角函數

任意三角形有三個邊，三個角，科學家為了找出其邊與角的關係，就定義出三角函數。例如已知三角形任兩邊與其所夾角度，就可透過餘弦定理找出第 3 邊的長度。所以現代人蓋六角庭，蓋橋樑、蓋房子，可以將所有木頭或鋼板長度通通事先裁切好，到了現場就是組裝而已。但是古代的工匠就窘了，他們沒有三角函數，所以是一面作，一面量下一根木頭長度，一面裁切，那速度當然慢。以下是 $\sin()$ 函數的定義，變數 x 是角度，單位是徑度量。

```
y=sin(x)
```

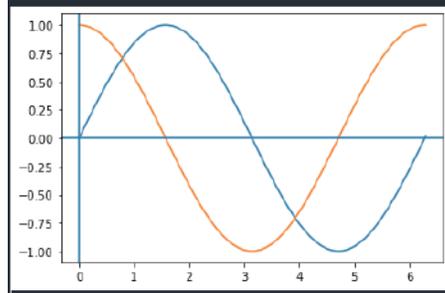
x 定義域是 0 到 2π ，值域在正負 1 之間，所以撰寫程式如下：

```

import matplotlib.pyplot as plt
import numpy as np
x = np.arange(0, 2*np.pi+0.1, 0.1) #x從0~2pi遞增，每次遞增0.1
y1=np.sin(x) #計算sin(x)
y2=np.cos(x) #計算cos(x)
plt.plot(x,y1) #繪出sin(x)圖形
plt.plot(x,y2) #繪出sin(x)圖形
plt.axhline(y=0) #繪出x軸
plt.axvline(x=0) #繪出y軸
plt.show() #於螢幕輸出圖形

```

執行結果如圖 7-13。



★ 圖 7-13 三角函數圖

7-3 自定函式

若內定函式沒有可用的函式，當然可以自己寫，這樣可以讓程式模組化，增加程式維修的便利性。例如，以下就是很簡單的範例，請先鍵入此程式，觀察執行結果。請留意 Python 的縮排有意義，也代表函式的範圍；冒號 (:) 也是 Python 的語法，不可省略。

```
#自訂函式
def add(a,b): #參數a接收m，參數b接收n
    print(a,b)
    c=a+b
    return c #將結果c傳回
#主程式
m=3;n=4
p=add(m,n) #主程式的m,n稱為引數
#呼叫自訂函式add，m傳給函式的a，n傳給函式的b
#函式傳回值由主程式p接收
print(p)#7
```

此程式 `add(a,b)` 稱為自訂函式，由主程式 `add(m,n)` 呼叫才執行，且接收傳遞而來兩個引數，`a` 接收 `m` 之值，`b` 接收 `n` 之值，主程式的 `m,n` 稱為引數 (Argument)，函式的 `a,b` 稱為參數 (Parameter)，自訂函式 `add`，將兩個參數相加，由 `c` 儲存，並將結果 `c` 傳回，由主程式 `p` 接收。以下我們分別說明 Python 函式的宣告與定義、函式參數傳遞、函式傳回值。

⚙️ 函式的宣告與定義

Python 函式的宣告與定義語法如下：

```
def 函式名稱([參數1, 參數2, 參數3...]) :
    [敘述區塊]
    [pass]
    [return 回傳1,回傳2,回傳3...]
```

以上語法說明如下：

1. 函式名稱可以是任何合法識別字，例如：aa、a1、add、inputdata 等。但盡量用函式功能命名，可讀性才高，例如：add、sum、inputdata 等。
2. 函式名稱當然不能用 Python 保留字，例如：for、if。
3. 函式的呼叫就直接使用函式名稱加上所需傳遞的引數就可以。例如，

```
add(3,4) #將(3,4)傳給以上add()函式，參數a得到3，參數b得到4
print(a)#7
m=3;n=4
add(m,n) #將(m,n)傳給以上add()函式，參數a得到m，參數b得到n
```

4. 函式的位置要放在主程式的前面，以下是全部程式列印。

```
#函式
def add(a,b):
    c=a+b
    return c
#主程式
p=add(3,4)
print(p)#7
```

5. Python 因為縮排關係，也不允許空函式，所以若函式沒內容（有可能先定義好介面，待後續再寫入程式），也是先放一個 pass。例如：

```
def aa():
    pass
#主程式
aa()
```

6. 函式名稱若與 Python 內建函式同名，則此自訂函式可改寫內建函式功能。例如：`len()` 是內建函式，可求物件的長度，以下則又使用 `len` 定義使用者自訂函式，等於改寫 `len()` 函式功能。

```
def len(a,b):#a接收m，b接收n
    c=a+b
    return c #將結果c傳回
p=len(3,4)
print(p)#7
```

⚙️ 函式參數傳遞

1. 參數可以是任意數量或無，例如：`def add()`、`def add(a)`、`def add(a,b)`。
2. 引數的傳遞若不指名，則是按照引數位置順序接收。例如：

```
def add(a,b):#a接收m，b接收n
    print(a,b)
m=3;n=4
p=add(m,n)
```

3. 主程式引數若指名，則按照指定的名稱傳遞。例如：

```
def add(a,b):#a接收n，b接收m
    print(a,b)
m=3;n=4
p=add(a=n,b=m) #於主程式指定引數與參數的接收方式
```

4. 函式參數若有預設值，則當主程式沒給此引數時，可用預設值替代。

```
def add(a,b=0):#本例參數b有預設值0
    c=a+b
    return c
m=3;n=4
print(add(m,n)) #7    有給2個參數
print(add(m)) #3    第2個參數b沒給，參數b用預設值0
```

5. 引數若是不定數量，請在參數前加上 (*) 號，例如：

```
def add(*a):      # *代表可接收不定數量的引數
    c=0
    for a1 in a:
        c=c+a1
    return c
#主程式
print(add()) #0   可以沒有引數
print(add(3)) #3  1個引數
print(add(3,4)) #7 2個引數
```

6. 引數可以任何資料型態，單一變數、串列 (list)、tuple、dict、set 等等都可以當作引數。而且資料型態是可變的，完全依照使用者傳入的資料型態（此即為物件導向「多型」的概念）。例如：請鍵入以下程式，寫出執行結果。

```
def aa(a):
    print(a)
#主程式
aa(3) #數值
aa('a') #字元
aa("Mary") #字串
aa(True) #布林值
aa([3,4,5]) #list
aa((3,4,5)) #tuple
aa({"book":"書","pen":"筆"}) #dict
aa({3,4,5}) #set
```

7. 以下引數則是傳遞 list。

```
def add(a) :
    c=0
    for a1 in a:
        c=c+a1
    return c
#主程式
b=[3,4,5]
print(add(b)) #12
```

⚙️ 函式傳回值

Python 函式可用參數或 `return` 傳回值給主程式，分別說明如下：

⚙️ 引數傳回值

使用參數傳回值的說明如下：

1. 函式的參數值的改變不傳回者，稱為傳值呼叫。單一變數、`tuple`、等都是傳值呼叫。例如：以下程式，參數 `a` 是單一變數，`a` 值雖在函數被修改，但並不回傳。

```
def aa(a):  
    print(a) #3  
    a=4  
#主程式  
m=3  
aa(m)  
print(m) #3
```

2. `tuple` 不論在主程式或函式都不能修改其值，所以是單向傳遞。
3. 參數若是串列、`dict`、`set`，則是傳址呼叫，傳址呼叫表示此參數在主副程式共用位址，任一方的改變，都是改變同一變數，也就是參數在函式的改變，會傳回主程式，是一種雙向傳遞，此稱為傳址呼叫。
4. 以下程式是串列的傳址呼叫，串列若在函式有變更其值，也會將變更結果回傳。

```
def bb(a):  
    print(a) # [3,4,5]  
    a[1]=6 #修改串列的內容  
#主程式  
m=[3,4,5] #list 本書翻譯為串列  
bb(m)  
print(m) # 串列有傳回修改結果  
#[3,6,5]
```

5. 因為串列是傳址呼叫，若不想接收被修改的結果，請改為傳串列副本到函式，如以下程式的 `bb(m[:])`。

```
def bb(a):
    print(a) # [3,4,5]
    a[1]=6 #修改串列的內容
#主程式
m=[3,4,5] #list 本書翻譯為串列
bb(m[:]) #傳副本
print(m) # 本例是傳串列副本，所以沒傳回
#[3,4,5]
```

6. 以下程式是 dict 的傳址呼叫。

```
def aa(a):
    print(a) # {'book': '書', 'pen': '筆'}
    a["book"]="黃金" #修改 dict的內容
#主程式
m={"book":"書","pen":"筆"} #dict
aa(m)
print(m) # dict 有傳回
#{'book': '黃金', 'pen': '筆'}
```

7. 以下程式是 set 的傳址呼叫。

```
def aa(a):
    print(a) #{3,4}
    a.add(6)
#主程式
m={3,4}
aa(m)
print(m) #{3,4,6}
```

8. C/C++ 可用修飾字改變參數為傳值或傳址，Python 則不行。也就是 Python 是由資料型態種類決定其為傳值或傳址，而且沒有修飾字改變其傳遞方式。

⚙️ return 傳回值

以上是用參數傳回函式值，Python 也可用 return 傳回值，return 傳回值的語法如下：

```
[return 回傳1,回傳2,回傳3...]
```

1. 傳回值可以是任何資料型態，個數也沒有限制，傳回幾個都可以，但是主程式也要使用相同個數的變數接收回傳值。例如，以下程式未傳回任何值。

```
def aa():  
    pass  
#主程式  
aa()
```

2. 以下程式傳回一個單一變數，主程式也要用一個變數接收回傳值。

```
def aa():  
    b="Mary"  
    return b  
#主程式  
c=aa()  
print(c)
```

3. 以下程式傳回一個單一變數與一個串列，主程式也要用相同數量變數接收回傳值。

```
def aa() :  
    c=1  
    d=[2,3]  
    return c,d #傳回個數沒有限制  
a,b=aa() #a,b個數要與return c,d一致  
print(a) #1  
print(b) #[2,3]
```

範例7-3a

試寫一程式，計算 C_n^m 的值。

提示： $C_n^m = \frac{m!}{n!(m-n)!}$ 。

$m! = 1 * 2 * 3 \cdots * m$ (此為高中數學)

👉 程式列印

方法一，未使用函式，階乘共需撰寫 3 次：

```
m=5
n=2
s1=1
for i in range(1,m+1):
    s1=s1*i
s2=1
for i in range(1,n+1):
    s2=s2*i
s3=1
for i in range(1,m-n+1):
    s3=s3*i
print(s1/(s2*s3))#10
```

方法二：使用函式。本例階乘共需寫三次，所以寫成函式 fsum。

```
def fsum(a):
    s=1
    for i in range(1,a+1):
        s=s*i
    return s
m=5
n=2
s1=fsum(m)
s2=fsum(n)
s3=fsum(m-n)
print(s1/(s2*s3))#10.0
```

👉 自我練習

1. 請寫一個程式，內含四個函式，分別可執行兩個整數的加減乘除等四個功能。

7-4 全域變數與區域變數。

在主程式宣告的變數稱為全域變數 (Global Variable)，在函式宣告內的變數稱為區域變數 (Local Variable)，全域變數可在任意地方取用，區域變數則僅能在自己函式內存取。例如：以下程式的 `c` 稱為全域變數，變數 `a,d` 稱為區域變數，變數 `d` 雖然名稱相同，但也彼此獨立。

```
def aa() :
    a=3
    d=2
    print(c)#6
def bb():
    d=3
    print(c)#6
#主程式由此開始
c=6
```

以下程式的變數 `c` 也彼此獨立。這樣有一個好處，自己要用的變數，自己宣告，那才不會誤用到全域變數。

```
def aa() :
    c=5
    print(c)#5
c=6
aa()
print(c)#6
```

► global

函式內僅能讀取全域變數的值，無法修改其值，若一定要修改其內容，則一定要使用 `global` 宣告。例如，以下程式僅可讀取全域變數 `c`。

```
def aa() :
    print(c) #6
c=6
aa()
print(c)#6
```

以下程式就不行，因為函式中有修改主程式的 `c`。

```
def aa() :
    print(c) #無法讀取
    c=5
    print(c)
c=6
aa()
print(c)
```

在函數裡，若一定要修改主程式的變數值，則要用 `global` 宣告，所以上述程式修改如下：

```
def aa() :
    global c
    print(c) #6
    c=5
    print(c)#5
c=6
aa()
print(c)#5
```

以上 `c` 加上 `global`，就稱為全域與區域共用，則函式的執行結果也會回傳主程式。(在區域變數用 `global` 特別宣告的好處是，只有特別標注的變數，才會影響主程式的值，這樣區域變數才不會誤用主程式的變數值，造成無法預期的錯誤。)

7-5 函式應用實例演練

範例7-5a

判斷任意點 `D` 是在三角形 `ABC` 內或外？

已知 `ABC` 三點座標，請寫一程式，可以輸入 `D` 點座標，並判斷任一點 `D` 是在三角形 `ABC` 內或外？

👉 執行結果

```
10.0 5.0 2.0 3.0
IN
```

👉 運算思維

D 若在三角形內，則三角形 ABC 面積 = $\triangle DAB + \triangle DBC + \triangle DAC$ 的面積。

因為三角形面積要計算四次，所以，先以 aa() 函式寫好，每次計算面積，只要呼叫 aa() 函式就好，程式如下：

```
def aa(x1,y1,x2,y2,x3,y3):
    return abs((x1*y2+x2*y3+x3*y1-x2*y1-x3*y2-x1*y3))/2
#三角形ABC，A(x1,y1)，B(x2,y2)，C(x3,y3)
x1=0;y1=0
x2=0;y2=5
x3=4;y3=0
x4=2;y4=1#D點
a=aa(x1,y1,x2,y2,x3,y3)#原來三角形面積
a1=aa(x4,y4,x1,y1,x2,y2)#DAB面積
a2=aa(x4,y4,x1,y1,x3,y3)#DAC面積
a3=aa(x4,y4,x2,y2,x3,y3)#DBC面積
print(a,a1,a2,a3)
if a==(a1+a2+a3):
    print("IN")
else:
    print("NOT IN")
```

👉 補充說明

1. 本例變數的取法要有技巧，不然程式不好寫。
2. 本例若沒有使用函式，計算三角形面積要寫四次，程式不但很冗長，也沒有一致性，錯在哪裡也不曉得。

範例 7-5b

主客場籃球賽制。(108 年 APCS 第二梯次試題)

1. 連打兩場，主場球隊兩場皆贏輸出 Win。
2. 兩場皆輸，輸出 Loss。
3. 其餘輸出 Tie。
4. 輸入格式：每場 4 節得分，主隊先輸入。以下分別是主隊、客隊兩場比

賽的四小節得分

```
15 20 25 30
15 16 17 18
5 10 15 20
5 6 7 8
```

👉 執行結果

```
input a b c d:15 20 25 30
input a b c d:15 16 17 18
input a b c d:5 10 15 20
input a b c d:5 6 7 8
Win
```

👉 程式列印

參考解答如下：輸入資料與計算總分共要執行四次，所以將輸入資料與計算總分寫成函式 `inputdata()`。

```
def inputdata():
    p=input('input a b c d:')
    q=p.split(' ')#本例測資是以空白分隔，所以要以split分解，請看8-1節
    print(q)
    r=[int(q1) for q1 in q]#將字串轉為數值
    print(r)
    return sum(r)#計算4小節總和
a1=inputdata()
b1=inputdata()
a2=inputdata()
b2=inputdata()
num=0
if a1>b1 :
    num=num+1
if a2> b2:
    num=num+1
if num==2:
    print('Win')
```

```

else:
    if num==0:
        print('Loss')
    else:
        print('Tie')

```

⚙️ 選單服務

較大的程式，通常有選單讓使用者選擇所要的服務，此時就可以將每一個子功能寫成一個函式，請看以下範例說明。

範例 7-5c

電腦擇友程式。假設有電腦擇友資料如下：

編號	姓名	性別	學歷	年收入(萬)	身高	體重	出生年
a01	Mary	0	6	120	158	50	76
b01	Jhon	1	5	100	170	70	76
b02	Tom	1	5	90	168	66	77
b03	Jack	1	4	80	172	90	74

請完成以下子功能。

- (1) 姓名查詢。輸入姓名，輸出其全部資料。
- (2) 性別查詢。
- (3) 依學歷查詢。

本例四個子功能，就用四個函式，這樣彼此獨立，就可由數個人合力完成一個大的程式專案。

```

n=4
a=[['a01','Mary',0,6,120,158,50,76],
    ['b01','Jhon',1,5,100,170,70,76],
    ['b02','Tom',1,5,90,168,66,77],
    ['b03','Jack',1,4,80,172,90,74]]
def a1():
    name=input('input name:')
    for i in range(n):
        if name==a[i][1]:
            for j in range(8):

```

```

        print(a[i][j],end=' ')
    return
def a2():
    sex=int(input('input 性別:'))
    for i in range(n):
        if sex==a[i][2]:
            for j in range(8):
                print(a[i][j],end=' ')
            print()
def a3():
    edu=int(input('input 學歷:'))
    for i in range(n):
        if a[i][3]>=edu:
            for j in range(8):
                print(a[i][j],end=' ')
            print()
while True:
    b=input('1:姓名查詢。 2:性別查詢。3:學歷查詢 4:結束:')
    if b=='1':
        a1()
    elif b=='2':
        a2()
    elif b=='3':
        a3()
    else:
        break

```

範例7-5d

以函式實現電子琴。

👉 執行結果



👉 程式列印

前面範例 6-2d 的自我練習已經複習電腦如何發聲、如何以選擇結構完成電子琴，本範例則使用「函式」重做電子琴，如此可提高程式結構化，參考程式如下：

```
from tkinter import * #載入視窗模組，此不是本書範圍，先體驗
import ctypes #載入發音模組
p = ctypes.windll.kernel32 #產生發聲物件
    #先使用函式定義所有按鍵功能
def a1(e): #定義a1函式
    p.Beep(523,200) #發Do音
def a2(e):
    p.Beep(587,200)
def a3(e):
    p.Beep(659,200)
def a4(e):
    p.Beep(698,200)
def a5(e):
    p.Beep(784,200)
def a6(e):
    p.Beep(880,200)
def a7(e):
    p.Beep(988,200)
def a8(e):
    p.Beep(1046,200)
tk=Tk() #產生視窗，此不是本書範圍，先體驗
tk.geometry("300x50+50+70") #視窗大小，此不是本書範圍，先體驗
enyl=Entry(tk) #產生輸入元件盒，此不是本書範圍，先體驗
enyl.bind('1',a1) #bind()方法可綁定按鍵1，且執行a1函式
enyl.bind('2',a2) #bind()方法可綁定按鍵2，且執行a2函式
enyl.bind('3',a3)
enyl.bind('4',a4)
enyl.bind('5',a5)
enyl.bind('6',a6)
enyl.bind('7',a7)
enyl.bind('8',a8)
enyl.pack() #元件配置方式，此不是本書範圍，先體驗
tk.mainloop() #於螢幕輸出此視窗，此不是本書範圍，先體驗
```

7-6 本章內容摘要

1. 程式要模組化才能提高後續維護性。
2. Python 內建函式庫分為內建函式與模組。例如：以下分別使用內建函式與模組完成次方運算。

```
a=2
print(pow(a,3)) #8使用內建函式
import math as m
print(m.pow(a,3)) #8.0使用內建math模組
```

3. Python 函式的宣告與定義語法如下：

```
def 函式名稱([引數1, 引數2, 引數3...]) :
    [敘述區塊]
    [pass]
    [return 回傳1,回傳2,回傳3...]
```

4. 在主程式宣告的變數稱為全域變數，在函式宣告內的變數稱為區域變數，全域變數可在任意地方取用，區域變數則僅能在自己函式內存取。

課後評量

(填充題，請寫出程式執行結果)

題號	題目	
1	<pre>a="123" b="456" print(a+b) print(int(a)+int(b)) print(a*3)</pre>	
2	<pre>print(round(555.555,1))</pre>	
3	<pre>a=divmod(8,3) print(a) #請留意型態</pre>	
4	<pre>print(eval("4*5+5**2"))</pre>	
5	<pre>a=-3.14 print(abs(a)) import math as m print(m.fabs(a))</pre>	
6	<pre>import numpy as np b=np.roots([1,-4,-12])#x**2-4x-12=0 print(b)</pre>	
7	<pre>import numpy as np c=np.array([[2,1],[1,-1]]) d=np.array([4,-1]) ans=np.linalg.solve(c,d)</pre>	
8	<pre>import matplotlib.pyplot as plt import numpy as np x = np.arange(-3, 2, 1) #-3開始到2，間隔是1，不含結束點 2 print(x) y = 3*x -1 print(y)</pre>	
9	<pre>def add(a,b) :#a接收m，b接收n c=a+2*b return c #將結果c傳回 p=add(3,4)</pre>	

10	<pre>def add(a,b) :#a接收n，b接收m print(a,b) m=3;n=4 p=add(a=n,b=m)</pre>	
11	<pre>def add(a,b=0) :#a接收n，b接收m c=a+b return c print(add(3)) #3 沒給引數，用預設值0</pre>	
12	<pre>def add(*a) :#a接收n，b接收m c=0 for a1 in a: c=c+a1 return c #主程式 print(add()) #0 可以沒有引數 print(add(3)) #3 1個引數 print(add(3,4)) #7 2個引數</pre>	
13	<pre>def aa(a): a=4 #主程式 m=3 aa(m) print(m)</pre>	
14	<pre>def bb(a): a[1]=6 #修改串列的內容 #主程式 m=[3,4,5] #list 本書翻譯為串列 bb(m) print(m)</pre>	
15	<pre>def bb(a): a[1]=6 #修改串列的內容 #主程式 m=[3,4,5] #list 本書翻譯為串列 bb(m[:]) print(m)</pre>	

16	<pre>def aa(a): print(a) # {'book': '書', 'pen': '筆'} a["book"]="黃金" #修改 dict的內容 #主程式 m={"book": "書", "pen": "筆"} #dict aa(m) print(m)</pre>	
----	---	--

課後習題

1. 請寫一程式，使用函式完成 $1! + 2! + 3! + 4! + \dots + n!$ (n 可由使用者輸入)
2. 請寫一程式，使用函式完成 $1+(1+2)+(1+2+3)+\dots+(1+2+3+\dots+n)$ (n 可由使用者輸入)
3. 請寫一程式，使用函式完成 $1+(1+1/2)+(1+1/2+1/3)+\dots+(1+1/2+1/3+\dots+1/n)$ (n 可由使用者輸入)
4. 於範例 6-2a，請先自行撰寫求和、求平均、求極大值、求極小值等函式，並呼叫自己的函式完成程式功能。
5. 於範例 6-2d，請先自行撰寫十進位轉 N 進位函式，然後呼叫此函式，完成程式功能。
6. 於範例 6-3d，請先撰寫串列相加與串列輸出函式，然後呼叫函式完成程式。
7. 生產排程系統實例探討。(摘自工科程式設計技藝競賽 111 年試題)

說明：小青畢業後，應徵到南港公司生產課工作，其第一個工作主要是設計一「雙向且循環之生產排程系統」，南港公司原來的排程系統就像水管一樣，是單向的，只能由左邊進，右邊出，這樣的排程，很沒有效率，公司希望改為雙向的，即由左邊進，右邊出之外，也可以由右邊進，左邊出，也就是，左邊同時可以進出，右邊也同時可以進出。而且還要可以循環，即最左邊和最右邊是接在一起的，例如，生產線有 5 位作業員，雖然像水管一樣，坐成一直線，其編號由左而右為 0, 1, 2, 3, 4，循環即此 5 位作業員像是坐在一個圓桌，作業員編號 0 和編號 4 是在隔壁。請您幫小青寫一個「雙向且循環之生產排程系統」來完成南港公司生產課交付的工作。

輸入說明：

可以讓使用者輸入生產排程選項和物件編號。

輸出說明：

輸出從左邊和右邊加入或刪除物件後的左邊和右邊作業員的編號。若生產線是空的，要顯示「生產線空的！」，若是生產線是滿的，要顯示「生產線滿了！」

程式功能和要求：

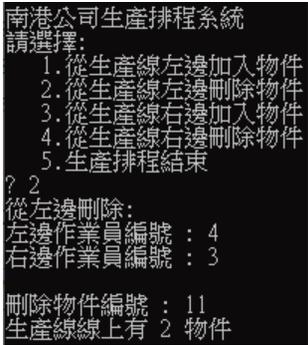
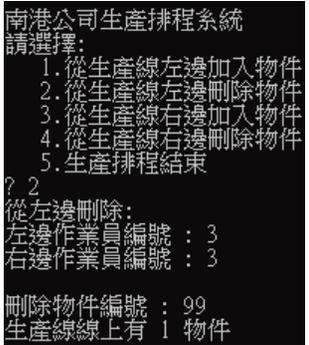
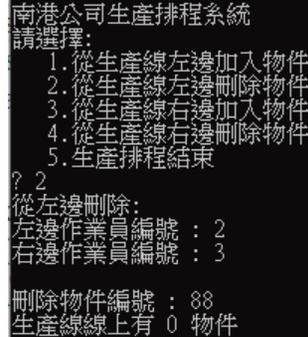
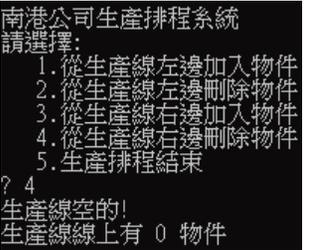
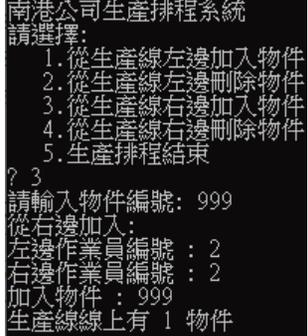
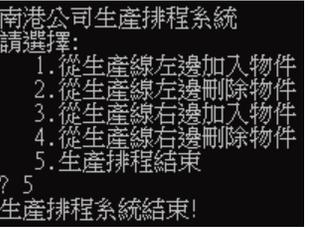
請利用上述說明，假設生產線有 5 位作業員，寫一個程式，能完成以下功能要求：

- (1) 能讓使用者輸入生產排程選項和物件編號，同時具有防呆功能，防止生產線管理人員輸入錯誤。
- (2) 能讓生產線管理人員，正確地由生產線左邊進行加入和刪除物件功能，同時顯示目前生產像狀況（詳見輸出說明）。
- (3) 能讓生產線管理人員，正確地由生產線右邊進行加入和刪除物件功能，同時顯示目前生產像狀況（詳見輸出說明）。
- (4) 生產線排程系統，具有循環功能（詳見題目說明）

程式執行範例：

防呆 1	防呆 2
<pre> 南港公司生產排程系統 請選擇： 1. 從生產線左邊加入物件 2. 從生產線左邊刪除物件 3. 從生產線右邊加入物件 4. 從生產線右邊刪除物件 5. 生產排程結束 ? 0 輸入錯誤! 請重新輸入! </pre>	<pre> 南港公司生產排程系統 請選擇： 1. 從生產線左邊加入物件 2. 從生產線左邊刪除物件 3. 從生產線右邊加入物件 4. 從生產線右邊刪除物件 5. 生產排程結束 ? 6 輸入錯誤! 請重新輸入! </pre>
<p>從左邊加入物件 11</p>	<p>從左邊加入物件 22</p>
<pre> 南港公司生產排程系統 請選擇： 1. 從生產線左邊加入物件 2. 從生產線左邊刪除物件 3. 從生產線右邊加入物件 4. 從生產線右邊刪除物件 5. 生產排程結束 ? 1 請輸入物件編號: 11 從左邊加入: 左邊作業員編號 : 0 右邊作業員編號 : 0 加入物件 : 11 生產線線上有 1 物件 </pre>	<pre> 南港公司生產排程系統 請選擇： 1. 從生產線左邊加入物件 2. 從生產線左邊刪除物件 3. 從生產線右邊加入物件 4. 從生產線右邊刪除物件 5. 生產排程結束 ? 1 請輸入物件編號: 22 從左邊加入: 左邊作業員編號 : 1 右邊作業員編號 : 0 加入物件 : 22 生產線線上有 2 物件 </pre>

<p>從左邊加入物件 33</p> <pre> 南港公司生產排程系統 請選擇: 1. 從生產線左邊加入物件 2. 從生產線左邊刪除物件 3. 從生產線右邊加入物件 4. 從生產線右邊刪除物件 5. 生產排程結束 ? 1 請輸入物件編號: 33 從左邊加入: 左邊作業員編號 : 2 右邊作業員編號 : 0 加入物件 : 33 生產線線上有 3 物件 </pre>	<p>從左邊刪除物件 33</p> <pre> 南港公司生產排程系統 請選擇: 1. 從生產線左邊加入物件 2. 從生產線左邊刪除物件 3. 從生產線右邊加入物件 4. 從生產線右邊刪除物件 5. 生產排程結束 ? 2 從左邊刪除: 左邊作業員編號 : 1 右邊作業員編號 : 0 刪除物件編號 : 33 生產線線上有 2 物件 </pre>
<p>從右邊加入物件 99</p> <pre> 南港公司生產排程系統 請選擇: 1. 從生產線左邊加入物件 2. 從生產線左邊刪除物件 3. 從生產線右邊加入物件 4. 從生產線右邊刪除物件 5. 生產排程結束 ? 3 請輸入物件編號: 99 從右邊加入: 左邊作業員編號 : 1 右邊作業員編號 : 4 加入物件 : 99 生產線線上有 3 物件 </pre>	<p>從右邊加入物件 88</p> <pre> 南港公司生產排程系統 請選擇: 1. 從生產線左邊加入物件 2. 從生產線左邊刪除物件 3. 從生產線右邊加入物件 4. 從生產線右邊刪除物件 5. 生產排程結束 ? 3 請輸入物件編號: 88 從右邊加入: 左邊作業員編號 : 1 右邊作業員編號 : 3 加入物件 : 88 生產線線上有 4 物件 </pre>
<p>從右邊加入物件 77</p> <pre> 南港公司生產排程系統 請選擇: 1. 從生產線左邊加入物件 2. 從生產線左邊刪除物件 3. 從生產線右邊加入物件 4. 從生產線右邊刪除物件 5. 生產排程結束 ? 3 請輸入物件編號: 77 從右邊加入: 左邊作業員編號 : 1 右邊作業員編號 : 2 加入物件 : 77 生產線線上有 5 物件 </pre>	<p>從右邊加入物件 66，但生產線滿了！</p> <pre> 南港公司生產排程系統 請選擇: 1. 從生產線左邊加入物件 2. 從生產線左邊刪除物件 3. 從生產線右邊加入物件 4. 從生產線右邊刪除物件 5. 生產排程結束 ? 3 請輸入物件編號: 66 生產線滿了! 生產線線上有 5 物件 </pre>
<p>從右邊刪除物件 77</p> <pre> 南港公司生產排程系統 請選擇: 1. 從生產線左邊加入物件 2. 從生產線左邊刪除物件 3. 從生產線右邊加入物件 4. 從生產線右邊刪除物件 5. 生產排程結束 ? 4 從右邊刪除: 左邊作業員編號 : 1 右邊作業員編號 : 3 刪除物件編號 : 77 生產線線上有 4 物件 </pre>	<p>從左邊刪除物件 22</p> <pre> 南港公司生產排程系統 請選擇: 1. 從生產線左邊加入物件 2. 從生產線左邊刪除物件 3. 從生產線右邊加入物件 4. 從生產線右邊刪除物件 5. 生產排程結束 ? 2 從左邊刪除: 左邊作業員編號 : 0 右邊作業員編號 : 3 刪除物件編號 : 22 生產線線上有 3 物件 </pre>

<p>從左邊刪除物件 11</p>	<p>從左邊刪除物件 99，99 是由右邊加入的！</p>
	
<p>從左邊刪除物件 88，88 是由右邊加入的！</p>	<p>從右邊刪除物件，但是生產線是空的！</p>
	
<p>從右邊加入物件 999</p>	<p>輸入 5，生產排程系統結束！</p>
	

提示：

1. 本例可以建立串列、左邊作業員編號、右邊作業員編號、物件數量等變數。

a=[0]*5 # 動態儲存所加入物件編號

lf=4 # 左邊作業員編號

rf=0 # 右邊作業員編號

objsum=0 # 物件數量

2. 仔細觀察以上執行結果，左邊加入、左邊刪除的作業員編號變化，此即為以上 `lf` 的變化，且其變化為圓形。
3. 仔細觀察以上執行結果，右邊加入、右邊刪除的作業員編號變化，此即為以上 `rf` 的變化，且其變化為圓形。
4. 仔細觀察物件數量的變化，此即為以上 `objsum` 的變化。

綜合應用實例

學習綱要

- ⊞ 8-1 字串的運算
- ⊞ 8-2 時間的運算
- ⊞ 8-3 本章內容摘要

學習目標

- ⊞ 1. 能實作字串的比對、擷取等運算。
- ⊞ 2. 能實作時間的長度與前後等運算。

8-1 字串的運算

Python 並沒有細分字元或字串型態，用單引號『b='Horng'』或雙引號『a="Horng"』圍起的字元，通通是字串，例如：

```
a="Horng"
print (a)
```

要從字串取字元，可以同串列的取單一元素。例如：

```
print(a[0])#H
b='Horng'
print(b)
print(b[0])
```

⚙️ 字串與串列、字串與集合互轉

請鍵入以下程式，並觀察執行結果。

```
a="AABBCCC"
b=list(a) #字串轉串列
print(b) #['A', 'A', 'B', 'B', 'C', 'C', 'C']
c="".join(b) #串列轉字串
print(c) #AABBCCC
d=set(a) #字串轉集合
print(d) #{'A', 'C', 'B'} #已經剔除相同字元
e="".join(d) #集合轉字串
print(e) #ACB
```

子字串也可用子串列方法處理。例如，請鍵入以下程式，並觀察執行結果。

```
a='abcdef'
print(a[0]) #a取索引0
print(a[-1]) #f 取倒數第1個
print(a[-2]) #e取倒數第1個
print(a[0:3]) #abc 取索引0~2
print(a[2:]) #cdef 取索引2開始到結束
print(a[-2:]) #ef 取倒數兩個
```

```
print(len(a)) #6 計算長度
print(max(a)) #f 取最大元素
b=list(a) #轉為串列 ['a', 'b', 'c', 'd', 'e', 'f']
print(b)
c="".join(b) #轉為字串 abcdef
print(c)
```

► split()

幾乎所有程式語言都是使用 split() 分割字串，Python 也不例外。例如：

```
a="There are four people in my family."
b=a.split(" ")#以空白為分割依據。
print(b)
print(b[0])
print(b[1])
```

以上 b 是一個串列，輸出如下圖：

```
['There', 'are', 'four', 'people', 'in', 'my', 'family. ']
There
are
```

► 測資含有空白

有些考試測試資料在同一列含有多筆資料，中間以空白隔開，例如

```
2 3 4
```

此時可用 split 分解資料，例如：

```
d=input("input a b c:") #本例假設輸入 2 3 4
print(d) #2 3 4
a,b,c=d.split(' ') #以空白分割資料
print(a+b+c) #字串型態 234
a=int(a);b=int(b);c=int(c)
print(a+b+c) # 9 數值型態輸出
```

► map()

map() 函式是 Python 內建資料型態轉換函式，其語法如下：

```
map(function, iterable, ...)
```

以上語法非常彈性，本書並不一一介紹，茲將以上字串分割與型態轉換程式使用 `map()` 重寫如下：

```
d=input("input a b c:") #本例假設輸入 2 3 4
a,b,c=map(int,d.split(' '))
print(a+b+c) # 9 數值型態輸出
```

👉 自我練習

1. 請寫一程式，讓使用者可以輸入任意數量的整數，整數與整數之間以空白隔開，輸入後可計算其和、平均、極大值與極小值。
2. 請寫一程式，可以計算所輸入句子的字數。例如，'There are four people in my family.' 表示共有 7 個字。
3. 請寫一程式，可以計算所輸入文章的句數。例如，' This is a book. There are four people in my family.' 表示共有 2 句。

▶ `replace(ch1,ch2)`

將 `ch1` 子字串由 `ch2` 取代。例如：以下程式可將全部的 `are` 都置換為 `is`。

```
a="There are four people in my family."
b=a.replace("are","is")
print(b)#There is four people in my family.
```

若 `ch2` 是空白，則是刪除 `ch1` 字串。例如：以下程式可刪除 `is`。

```
a="This is a book."
b=a.replace("is","") #將所有 is 全部刪除
print(b) #Th a book.
a=a.replace(" is","") #僅刪除單字is，請留意有無空白的差異
print(a) #This a book
```

▶ `startswith()`、`endswith()`

傳回字串是否由某一子字串開頭或結束。

```
a="洪國勝"
print(a.startswith("洪"))#True
print(a.endswith("勝"))#True
```

範例8-1a

統一發票對獎。請寫一程式，可以幫使用者核對統一發票中獎情形。

執行結果

六獎

程式列印

1. 將中獎號碼以字串儲存，如以下程式的 a1~a4。
2. 以下我們僅核對 1 張發票，以變數 b 儲存。

```

a1='59647042'#特別獎號碼
a2='01260528'#特獎號碼
a3=['01616970','69921388','53451508']#頭獎號碼
a4=['710','585','633']#增開六獎的號碼
#b='31616970' #測試資料1，欲核對的發票號碼
b='11111710' #測試資料2，欲核對的發票號碼
c=''
if b==a1 :
    c='特別獎'
elif b==a2 :
    c='特獎'
else:
    #逐一核對頭獎
    for i in range(len(a3)): #本例頭獎有三個
        d=a3[i]
        if b==d:
            c='頭獎'
        elif b[1:]==d[1:]:#大家都取末七位數，末七位數相同稱為二獎
            c='二獎'
        elif b[2:]==d[2:]: #大家都取末六位數，末六位數相同稱為三獎
            c='參獎'
        elif b[3:]==d[3:]: #大家都取末五位數，末五位數相同稱為四獎
            c='四獎'
        elif b[4:]==d[4:]: #大家都取末四位數，末四位數相同稱為五獎
            c='五獎'
        elif b[5:]==d[5:]: #大家都取末三位數，末三位數相同稱為六獎
            c='六獎'
    for i in range(len(a4)): #增開六獎

```

```

        if b[5:]==a4[i]:
            c='六獎'
print(c)

```

👉 自我練習

1. 以上範例，我們僅核對一張發票，但我們通常很多發票，請將使用者的 6 張發票以串列儲存，並由電腦核對中獎情形。
2. 請寫一程式，可以找出個字串所含的大寫字元、小寫字元、數字的個數。例如，'AcFgk7' 傳回大寫 2，小寫 3，數字 1

範例8-1b

中華民國的身分證字號有其特定的編碼原則。第一個字是大寫的英文字母，其餘 9 個字必須為數字，但在套用編碼原則時，第一個英文字母將會先依下表被轉換為數字：

字母	A	B	C	D	E	F	G	H	J	K	L	M	N
數字	10	11	12	13	14	15	16	17	18	19	20	21	22
字母	P	Q	R	S	T	U	V	X	Y	W	Z	I	O
數字	23	24	25	26	27	28	29	30	31	32	33	34	35

轉換後的身分證字號 (共 11 位數字) 每一位數均有固定的權重 (Weight)，由左往右依序為「1, 9, 8, 7, 6, 5, 4, 3, 2, 1, 1」。判斷身分證字號是否正確的方法為：各位數字與其相對應的權重相乘後再加總，加總後的結果若為 10 的倍數則身分證字號即屬正確。請根據以上說明，寫一程式，驗證身份證號碼是否正確。

👉 執行結果

```

65
10
10157387560
170
RIGHT

```

👉 程式列印

```
#a='0157387560' #測試號碼1
a='A157387560'#測試號碼2
#請留意，題目的字母沒有按照順序，以下我重新調整，由A~Z以串列儲存如下
b=[10,11,12,13,14,15,16,17,34,18,19,20,21,22,35,23,23,25,26,27,
  28,29,32,30,31,33]
c=[1,9,8,7,6,5,4,3,2,1,1]#權重
a0=ord(a[0])#取ASCII編碼
print(a0)#65 A的ASCII編碼是65
a1=b[a0-65]#將第一個字母轉為數字a1
print(a1)#10
a2=a[1:]#取末9碼157387560
a=str(a1)+a2#成功將第一個字母轉為數字
print(a)#10157387560 此時共11位數
#使用迴圈，依照11個數字的權重累計其和
s=0
for i in range(11):
    s=s+int(a[i])*c[i]
print(s)#170
if s %10 ==0:
    print("RIGHT")
else :
    print("Worng")
```

👉 自我練習

1. 於範例 8-1b，我們僅檢查一個身份證號碼，請您將 6 個身份證號碼以串列儲存，並一一檢查其是否正確。
- ※ 2. 請問如何自動產生一個身份證號碼，且此號碼可以通過以上檢查。

範例8-1c

加法。大整數加法。若有兩個 1 千位數的數字，要執行精密加法，請問要如何完成？

👉 執行結果

```
[0, 5, 4, 3, 2, 1]
[0, 5, 4, 3, 2, 1]
[0, 0, 9, 6, 4, 2]
24690
```

👉 運算思維

電腦當然無法儲存 1000 位數，通常是先以字串儲存，再轉為串列，然後使用以下國小方法，逐一進行每一個位數的相加，也要自行處理進位問題。

$$\begin{array}{r} 1567 \\ + 835 \\ \hline 2402 \end{array}$$

1. 以上逐位數的相加程式如下，個位數我放在索引 1：

```
a='12345' #被加數
b='12345' #加數
a1=[0] #本例索引0不用，先放0
b1=[0]
c1=[0]
alen=len(a)
blen=len(b)
#使用串列儲存被加數，個位數放在索引1
for i in range(0,alen):
    a1.append(int(a[alen-i-1]))
print(a1) #[0, 5, 4, 3, 2, 1]
#使用串列儲存加數，個位數放在索引1
for i in range(0,blen):
    b1.append(int(b[blen-i-1]))
print(b1) #[0, 5, 4, 3, 2, 1]
#從個位數，逐一位數相加
carry=0#進位
for i in range(1,alen+1):
    t=a1[i]+b1[i]+carry
    carry=t //10
    c1.append(t %10) #超過10的要進位
print(c1) #[0, 0, 9, 6, 4, 2]
for i in range(1,alen+1) :
    print(c1[alen-i+1],end='')
```

👉 自我練習

1. 同範例，請練習減法。

2. 以上題目，我們是假設長度相同，但若長度不一樣呢？（方法很多，最簡單而便宜行事的作法是，以長的為標準，將短的變長，通通化為一樣長）

範例8-1d

子字串搜尋。最簡單的字字串搜尋，就是循序法搜尋，以下示範如何使用循序法比對子字串出現的位置。

👉 執行結果

```
ab
bc
found in index 1
```

👉 程式列印

1. 以下程式，可以從來源字串依照子字串長度，取出每一個子字串。

```
a='abcdef'#來源字串
b='bc'#欲比對子字串
found=False
for i in range(len(a)-len(b)+1):
    c=a[i:i+len(b)]
    print(c)# ab bc cd de ef
```

2. 以下程式，加上字串比對，就可以找到相同子字串。

```
a='abcdef' #來源字串
b='bc' #欲比對子字串
found=False
#使用迴圈，逐一於a字串，從索引0開始，每次取b字串長度的字元，本例為2
for i in range(len(a)-len(b)+1):
    c=a[i:i+len(b)]
    print(c)
    if c==b :
        found=True
        break
if not found :
    print ('not found')
else:
    print ('found in index %d' %i)
```

範例8-1e

猜單字遊戲。

本例我寫一個猜單字遊戲，遊戲過程如下：

1. 輸出底線的個數提示單字的字母數，例如，單字是 apple，就輸出 5 個底線『_ _ _ _ _』。
2. 使用者每次僅能輸入一個 a 到 z 的字母。本例假設輸入『p』。
3. 電腦將單字中合乎此字母的通通現形，所以本例將會輸出『_ p p _ _』
4. 重複步驟 2 與 3，直到全部字母猜對為止。

執行結果

```

_ _ _ _ _
請輸入一個字母：a
a _ _ _ _
請輸入一個字母：e
a _ _ _ e
請輸入一個字母：p
a p p _ e
請輸入一個字母：l
a p p l e
BinGo

```

程式列印

```

a='apple'
s=len(a)
b=[]
c=[]
#使用迴圈，逐一取a字串的每個字元放入b串列
#c串列則放入相對應數量的_
for i in range(s) :
    b.append(a[i])
    c.append('_') #c串列放入相對應數量的 _
#輸出c串列所有字元
for i in range(s) :
    print('%c '% c[i],end='')
s1=0 #答對字元計數器
while (s1<s) : #只要沒有完全猜對，重複迴圈
    d=input('請輸入一個字母：')

```

```

#使用迴圈，逐一核對，將相同的字元取代原來底線
for i in range(s):
    if d==b[i]:
        c[i]=d
        s1=s1+1 #答對字元計數器
#輸出c串列所有字元，以便使用者繼續猜測
for i in range(s) :
    print('%c '% c[i],end='')
print()
print("BinGo")

```

👉 自我練習

1. 以上範例僅用一個單字，請擴大此題目，找來 50 個單字，每次以亂數出現一個單字。

範例8-1f

英文默寫。英文要能進步，就是要強迫自己能將看過的句子覆誦一次，您想想看，中文別人講一次，我們通常可以覆誦一次，其實英文也要這樣自我要求。以下是我寫的程式，只要將一篇文章先輸入或複製貼上，程式就會逐一出現每一個句子、每個句子的第一個單字、並將該句剩餘的單字以底線『_』標示，讓使用者填空輸入，輸入完電腦還能自動批改。

👉 執行結果

```

Action _____
請輸入句子0: Action speak louder than word.
錯了1 單字
正確句子是Action speak louder than words.
Wasting _____
請輸入句子1: Wasting time is robbing time.
錯了1 單字
正確句子是Wasting time is robbing oneself.
Never _____
請輸入句子2: Never say die.
全對

```

👉 運算思維

1. 將整篇文章複製，以字串儲存。
2. 將文章以點(.) 分解為若干句子。例如，3 個句字有 3 個點，那分解後有 4 句，最後一句是空白，所以本例程式撰寫就減 1。
3. 本例將每一個句子的第 1 個單字輸出，且每一個單字以底線『_』標示，這樣使用者比較好作答，電腦也比較方便批改。
4. 逐一由使用者輸入整個句子。
5. 本例假設使用者所輸入單字數與題目一樣。例如，題目提示 5 個單字，使用者就輸入 5 個單字。
6. 逐一按順序比對兩個句子的單字，並統計正確單字的個數，也就是第 1 個單字與第 1 個單字比對，第 2 個單字與第 2 個單字比對。
7. 若全部單字都對，輸出『全對』，否則輸出正確單字的個數。
8. 輸出正確的句子。

👉 程式列印

```

a="Action speak louder than words.Wasting time is robbing
  oneself.Never say die."
b=a.split('.')#整篇文章以「.」分解
#print(len(b))
#print(b)#由輸出發現，3個點將文章分為4句，最後1句空白，所以以下就減1
#使用迴圈，逐一輸出每一句英文
for i in range(len(b)-1):
    c=b[i]+'.' #每個句子將'.'補回來
    #每個句字再以空白分解
    d=c.split(' ')
    dlen=len(d)
    #print(dlen)
    print(d[0],end=' ')#輸出第1個單字
    for j in range(1,dlen) :#依照單字個數，輸出底線
        print('_____',end=' ')
    #print('.')
    e=input("請輸入句子%d:"%i)#由使用者輸入整句
    f=e.split(' ')#分解使用者輸入的句子
    #print(len(f))
    s=0
    #計算答對幾個單字

```

```

for j in range(dlen):#逐一每個單字比對
    if (d[j]==f[j]):
        s=s+1
if s==dlen :
    print("全對")
else:
    print("錯了%d 單字" %(dlen-s))
print("正確句子是%s"%c)

```

👉 自我練習

1. 本例假設使用者輸入的單字數與提示的單字數一致，若使用者輸入太長或太短都會造成執行錯誤，請問您如何修改程式補救。

範例8-1g

英文聽寫程式的研究。

👉 執行結果

將英文課文複製，然後貼到以下變數 a，電腦就會逐一唸出每一句，且電腦提示每一句的長度，使用者就可開始輸入聽到的單字，然後電腦還會檢查這些單字是否正確。以下是執行結果：

```

請輸入句子0:Action speak louder than words.
全對
正確句子是Action speak louder than words.

請輸入句子1:Wasting time is rob one.
錯了2 個單字
正確句子是Wasting time is robbing oneself.

請輸入句子2:Never say die.
全對

```

👉 運算思維

1. 我們於第一章曾經強調，Python 之所以強，就是有源源不絕的好用套件不斷發表，上網搜尋『Python 文字轉語音』，找到套件如下 (<https://codertw.com/%E7%A8%8B%E5%BC%8F%E8%AA%9E%E8%A8%80/368251/>)：

```
import win32com.client
speaker = win32com.client.Dispatch("SAPI.SpVoice")
a="Hello, it works!"
speaker.Speak(a)
a="洪國勝，您好"
speaker.Speak(a)
```

2. 以上程式真的可以發音，也就是視障者想聽某段文章，都可以將任何中英文文章複製，放到變數 a，就可以自動播放。或是，您也可將英文課本複製下來，貼到變數 a，電腦就會用語音輸出，讓您不斷練習聽力。
3. 根據以上程式，我將以上程式改寫如下，就可完成英文聽寫程式，希望可以幫助大家學習英文。

```
import win32com.client
speaker = win32com.client.Dispatch("SAPI.SpVoice")
speaker.Speak("Hello, 各位同學大家好，現在開始聽寫測驗")
#將課文複製，然後貼來以下變數a, 電腦就會逐一唸出每一句。
a="Action speak louder than words.Wasting time is robbing
  oneself.Never say die."
b=a.split('.')#整篇文章分解
#print(len(b))
#print(b)#由輸出發現，b的長度是4，所以以下就減1
#使用迴圈，逐一唸出每一句英文與輸出底線
for i in range(len(b)-1):
    c=b[i]+'.' #每個句子將'.'補回來
    speaker.Speak(c)#您要念幾遍，就寫幾次
    speaker.Speak(c)
    #每個句字再以空白分解
    d=c.split(' ')
    dlen=len(d)
    #print(dlen)
    #print(d[0],end=' ')
    #使用迴圈，依照句子長度輸出底線
    for j in range(dlen) :
        print('____',end=' ')
    #print('.')
    #由使用者輸入句子
    e=input("請輸入句子%d:"%i)
    f=e.split(' ')
```

```

#print(len(f))
s=0
#計算答對幾個單字
for j in range(dlen):
    if (d[j]==f[j]):
        s=s+1
if s==dlen :
    print("全對")
else:
    print("錯了%d 個單字" %(dlen-s))
print("正確句子是%s"%c)

```

4. 讀檔功能。以上是資料和程式寫在一起，這樣對不會程式的人還是困擾，以下我們則要程式和資料分離，將文章先放在記事本，以 a0.txt 存檔（請與 Python 程式放在同一資料夾），如下圖：



5. 撰寫以下程式，就可讀檔，且整篇發音。

```

import win32com.client
speaker = win32com.client.Dispatch("SAPI.SpVoice")
f='a0.txt'
with open(f,'r',encoding='utf-8') as f1:
    data=f1.read()
f1.close()
print(data)
speaker.Speak(data)

```

自我練習

1. 請將以上英文聽寫測驗程式套用讀檔程式 (with open() as f1)，這樣就可以將程式與資料分開，每次只要改變資料就可以，不用改程式。

範例8-1h

猜數字

有一種遊戲稱為幾 A 幾 B 的猜數字遊戲，兩個人對玩，互相猜對方預先寫下的四位數（四位數中的阿拉伯數字不可重覆），若所猜的數字與對方位置相同者為 A，數字相同，位置不對，則稱為 B。例如對方預寫的數字為 6713，若猜 6731 則應回應 2A2B，若猜 7652 則應回應 0A2B。試寫一程式，電腦自動產生四位數的亂數，再讓使用者猜的一種遊戲程式，電腦應逐一回應使用者已猜的狀況。其次，四位數亂數，數字不能重複，例如，1001、5566、1022 等都不行。

 **執行結果**

```
1709
輸入四位數：9071
0 a 4 b
輸入四位數：2345
0 a 0 b
輸入四位數：6789
2 a 0 b
輸入四位數：1790
2 a 2 b
輸入四位數：1709
4 a 0 b
BinGo
```

 **程式列印**

```
import random
c=[0,1,2,3,4,5,6,7,8,9]
#將以上數字打亂
for i in range(len(c)):
    d=random.randint(0,9)
    c[i],c[d]=c[d],c[i]
#取前四個數字，此即為亂數
e=str(c[0])+str(c[1])+str(c[2])+str(c[3])
print(e)
right=False
```

```

a=0
b=0
#使用while 迴圈由使用者重複猜測
while (not right):#只要沒完全答對，重複迴圈
    f=input('輸入四位數：')
    #使用for迴圈，計算a與b的數量
    for i in range(4):
        if f[i]==e[i]:
            a=a+1
        for j in range(4):
            if f[j]==e[i]:
                b=b+1
    b=b-a
    print('%d a %d b'%(a,b))
    if a==4:
        right=True
    a=0
    b=0
print("BinGo" )

```

※範例8-11

人工智慧與刪除法。

先將所有可能的答案通通列出來，再逐步將所有不可能的答案一一刪掉，最後剩下的值即為所求，此稱為刪除法。例如，同上題，剛剛是人腦猜，人腦有人腦的運算思維，但是電腦有電腦的運算思維，既然是電腦程式設計，那就要利用電腦的強項，才能事半功倍，以下示範電腦用刪除法來猜，使用者必須逐一回應電腦已猜的狀況。電腦猜數字若使用以上刪除法，則演算法如下：

- (1) 使用串列列出 0000 至 9999 的四位數。
- (2) 逐一刪除阿拉伯數字重複者，例如 1001 或 3343 等。
- (3) 於剩下的可能數字中，挑最小的當作猜值，本例為 0123。(使用者應回應幾 A 幾 B，本例假設使用者的數字是 4237，所以回應 0A2B)
- (4) 於剩下的可能值中，使用臆測值，本例為 0123 逐一比對，將結果是 0A2B 者，蒐集起來，也就是不是 0A2B 者全部刪除。(為什麼呢？因

為答案 4237 也會是 0A2B，將會被保留，此即為刪除法，逐一將不可能的答案，先行刪除。

(5) 重複 (3)、(4) 兩個步驟，直到剩下最後一個，就是答案了，或使用者回應 4A0B 為止。

👉 執行結果

下圖就是我預設 4237，電腦逐一猜值的過程。本例電腦 5 次就猜出結果，真的很強。其次，若您中間有任一過程回答錯了，例如，1045 應該 1B，您回答為 2B，那串列就會刪光，所以還要補足這一缺憾，避免程式當掉，這請讀者自行練習。

```
Computer guess:0123
input na , nb:0,2
Computer guess:1045
input na , nb:0,1
Computer guess:2356
input na , nb:0,2
Computer guess:3274
input na , nb:1,3
Computer guess:4237
input na , nb:4|
```

👉 程式列印

```
a=[]
b=[]
#使用串列列出0000、0001、0002至9999的四位數
for i in range(10):
    for j in range(10):
        for k in range(10):
            for l in range(10):
                a.append(str(i)+str(j)+str(k)+str(l))
print(len(a))
#輸出a串列
```

```
for i in range(len(a)):
    print(a[i])
#於a串列逐一蒐集數字不重複者
for i in range(len(a)):
    flag=True
    c=a[i]
    for j in range(3):
        for k in range(j+1,4):
            if c[j]==c[k]:
                flag=False
                break
    if flag:#沒有重複才蒐集
        b.append(c)
for i in range(len(b)):
    print(b[i])
a=b[:] #串列的複製，已經刪除數字重複者
b=[]#重新清為空串列
print(len(b))
right=False
#使用while 迴圈，由使用者開始猜測
while (not right):
    #每次都從串列索引0開始與答案比對
    d=a[0]
    print ('Computer guess:%s' %d)
    a1,b1=eval(input('input na , nb:')) #使用者輸入符合a,b的個數
    if a1==4 :
        right=True
        print("Bingo")
        break
#於a串列的每個數字與輸入答案比對，計算a,b的數量
for i in range(1,len(a)):
    a2=0
    b2=0
    f=a[i]
    for j in range(4):
        if f[j]==d[j]:
            a2=a2+1
    for k in range(4):
        if f[j]==d[k]:
            b2=b2+1
```

```

b2=b2-a2
    #將相同a 且相同b的數字蒐集起來，放入b串列
    if a1==a2 and b1==b2 :
        b.append(f)
#已經完成一階次的比對，b串列蒐集完畢
a=b[:] #將b串列取代a串列
b=[] #b串列清空，重複迴圈，直到答對

```

8-2 時間的運算

Visual Basic 有時間型態，所以兩個時間的間隔長度或前後判斷，可以大致如下：

```

Private Sub Command1_Click()
    a = #9/10/2019#
    b = #9/7/2019#
    Print a - b '3 計算兩個日期間隔
    Print (a > b) 'True 判斷時間前後
End Sub

```

但是 Python 並沒有時間型態，所以關於時間的處理，通通要自己來。日常關於時間的處理，大致有時間的表示、兩個時間的間隔，兩個時間的前後判斷，請看本節說明。

⚙️ 時間與資料結構

電腦如何記錄時間呢？內部是不是用年、月、日、時、分、秒六個變數表示呢，答案是否定的。例如：

```

import time #載入時間模組
print(time.time())#1657865839.2869213

```

以上是 Python 的 time 模組的 time() 方法，傳回 1970/1/1 起的秒數。當要給人看時，再轉為人類的時間模式。例如：

```

print(time.asctime())#傳回現在時間Wed Nov 6 15:57:20 2022

```

範例8-2a

時鐘程式。請寫一個可以顯示『hh:mm:ss』的 24 小時時鐘程式。

執行結果

```
23:58:57
23:58:58
23:58:59
```

運算思維

1. 時鐘就是每秒秒數加 1，所以資料結構就是指派秒數的變數為 t。
2. 要讓程式不間斷執行，就要一個無窮迴圈如下：

```
while 1: #while True也可以
```

3. 要讓程式依照時序連續相加，無限循環，且不溢位，就要聯想取餘數運算『%』。

```
while 1:
    t=(t+1)%(24*60*60)
```

4. 本例 t 會在 0~24*60*60 之間，然後要給人看時，再將秒數轉為『時:分:秒』，所以程式如下：

```
import time
t=23*60*60+58*60+56
while (1):
    t=(t+1)%(24*60*60)
    h=t//3600#時
    m=(t-h*3600)//60#分
    s=t %60#秒
    print('%2d:%2d:%2d'% (h,m,s))
    time.sleep(1)#暫停1秒
```

5. 本例有人會將資料結構先指派時、分、秒等三個變數

```
import time
h=23
m=58
```

```

s=55
while (1):
    s=s+1
    if s>=60:
        s=0
        m=m+1
        if m>=60:
            m=0
            h=h+1
            if h>=24:
                h=0
    print('%2d:%2d:%2d'% (h,m,s))
    time.sleep(1)#暫停1秒

```

這樣雖然也可以，但是這樣對於往後時間的長度計算與時間的前後判斷會很麻煩，請繼續看以下說明。

自我練習

1. 請設計一個程式，可以輸入一個分鐘數，例如，10 分鐘，程式一執行電腦就一直顯示秒數倒數計時。例如，顯示 09:59、09:58...

範例8-2b

時間的間隔長度計算。請寫一程式，可以計算兩個時間的間隔長度。

運算思維

1. Python 的資料結構沒有時間型態，所以時間就以字串表示。例如，輸入資料是

```

a='03:20'
b='05:10'

```

2. 使用字串的 split 分解如下：

```

a='03:20'
b='05:10'
h1,m1=map(int,a.split(':')) # map() 函數為資料型態轉換函數
h2,m2=map(int,b.split(':'))

```

3. 關於兩個時間差的計算，小學是用直式算法，若分鐘不夠要向『時』借位。但電腦就強了，因為電腦計算能力強，可直接通通化為分鐘，再直接相減，不管數字多大，總能迅速搞定。其次，若是兩個日期差，則可通通轉為日，再直接相減。所以學電腦程式設計，就要慢慢調整成電腦的運算思維。

```
a='03:20'
b='05:10'
h1,m1=map(int,a.split(':'))
h2,m2=map(int,b.split(':'))
t=(h2*60+m2)-(h1*60+m1)
print(t)#150
h=t//60
m=t%60
print("%2d:%2d"%(h,m))# 1:50
```

👉 自我練習

1. 停車場停車計費。同上範例，若停車場每小時 30 元，不足 1 小時以 1 小時計，請計算停車費用。測試資料如下：

```
a='03:20',b='05:10' 輸出1小時50分，停車費60元
```

2. 請寫一程式，可以比較兩個時間的前後關係。例如，輸入資料與輸出結果如下表。

輸入第一個時間	輸入第二個時間	輸出
a='03:20'	b='05:10'	True
a='03:20'	b='02:30'	False

提示：通通轉為分鐘數。

3. 請寫一程式，可以比較第三個時間是否在前面兩個時間中間。例如，輸入資料與輸出結果如下表。

輸入第一個時間	輸入第二個時間	輸入第三個時間	輸出
a='03:20'	b='05:10'	c='04:10'	True
a='03:20'	b='05:10'	b='01:10'	False

提示：通通轉為分鐘數。

4. 勞動基本法第 24 條如下：

雇主延長勞工工作時間者，其延長工作時間之工資，依下列標準加給：

- 一、延長工作時間在二小時以內者，按平日每小時工資額加給三分之一以上。
- 二、再延長工作時間在二小時以內者，按平日每小時工資額加給三分之二以上。
- 三、依第三十二條第四項規定，延長工作時間者，按平日每小時工資額加倍發給。

假設某員工每日工時 8 小時，每小時工資 300 元，請寫一個程式，可以依其簽到與簽退時間，計算當日加班費。因為勞基法也規定「前項雇主延長勞工之工作時間連同正常工作時間，一日不得超過十二小時，但因突發事件再延長者，需令員工補修或給雙倍工資，本例採用雙倍工資計算」。測試資料如下表，本例假設總工時不會超過 24 小時：

編號	簽到時間	簽退時間	總工時	正常工時	4/3倍工資 (400)工時	5/3倍工資 (500)工時	2倍工資 (600)工時	加班費
1	08:00	17:00	9	8	1	0	0	400
2	08:00	19:00	11	8	2	1	0	1300
3	0800	21:30	13.5	8	2	2	1.5	2700
4	22:00	07:00	9	8	1	0	0	400

範例 8-2c

假設 2 月永遠是 28 天，請寫一程式，可以指派一個日期 $b='3/1'$ ，且轉為今年第幾天。例如，3 月 1 日，就先累加 1,2 兩月的日數 (31+28)，再加上 3 月的日數 (1)，所以是第 60 天。

演算法

1. 先將每個月的天數以串列儲存如下，索引 0 可用可不用，本例不用，就放 0。

```
a=[0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]
```

2. 將日期使用字串分解函式 `split()` 分解。
3. 將日期的字串型態使用 `map()` 函式轉為數值型態。

👉 程式列印

```
a=[0,31,28,31,30,31,30,31,31,30,31,30,31]#每月的天數，索引0不用
b='3/1' #待分解日期
#分解為m,d
m,d=map(int,b.split('/'))
print(m,d)
num=0
#累加前面月份天數
for i in range(1,m):
    num=num+a[i]
#加上本月天數
num=num+d
print(num)#60
```

範例8-2d

同上題，假設 1 月 1 日是星期二，可求得輸入日期是星期幾？（本例先不考慮閏年，也就是 2 月是 28 天）

👉 運算思維

這就與生活上的推算相同，因為每 7 天 1 輪，這就要聯想取餘數『%』運算，然後規劃餘 0、餘 1、餘 2 的對應星期。所以規劃星期的查表如下：

```
w=['日','一','二','三','四','五','六']
```

👉 程式列印

```
a=[0,31,28,31,30,31,30,31,31,30,31,30,31]
b='4/1' #待分解日期
#分解為m,d
m,d=map(int,b.split('/'))
print(m,d)
num=0
#累加前面月份天數
for i in range(1,m):
```

```

    num=num+a[i]
#加上本月天數
num=num+d
print(num)
w=['日','一','二','三','四','五','六']
f=2 #1月1日星期幾
e=(num+f-1) %7
print(w[e])#一

```

自我練習

1. 同上範例，輸出指定月份的月曆。
2. 同上題，可以輸入兩個日期，計算其相差的天數。例如，輸入

```

2018 9 30
2018 10 1

```

則輸出 1。(此為國內高中生程式解題系統 (<https://zerojudge.tw/>) 基礎題。

提示：這題要先界定年份的最小值當成一個參考起點，本例就設為 2018 1 1，然後大家都先求到起點的日期，再相減就可以。

3. 同上題，但可比較兩個日期的先後。例如，應該還書日期是 2018 9 30，但是 2018 10 1 來還，表示逾期。

提示：這題要先界定年份的最小值當成一個參考起點，本例就設為 2018 1 1，然後大家都先求到起點的日期，再比較大小就可以。

範例8-2e

可以輸入天數，但轉為日期輸出，例如輸入 365，能得到 12 月 31 日。(本例先簡化題目，假設 2 月通通 28 天)

執行結果

```
365 is m=12 d=31
```

👉 運算思維

1. 將每個月的天數以串列儲存。
2. 從 1 月開始，只要天數大於每月天數，就扣掉，所以程式如下：

👉 程式列印

(本例就是典型 while 迴圈，因為程式設計階段不知道執行次數)

```
#將每月天數使用串列儲存
a=[0,31,28,31,30,31,30,31,31,30,31,30,31]
day=365 #待求解天數
m=1 #欲求月份
d=day #日期
#使用不定while迴圈逐一扣掉每月天數
while d>a[m]:
    d=d-a[m]
    m=m+1
print('%d is m=%d d=%d'%(day,m,d))
```

👉 自我練習

1. 同上範例，但考慮閏年。提示：閏年與否可以寫程式判斷也可以用現成 calendar 模組。
2. calendar 模組有閏年、月曆等功能，以下程式可檢驗是否閏年、輸出月曆、輸出年曆，請自己練習與體驗。

```
import calendar as c
print (c.isleap(2019))
print (c.month(2019,9))
print (c.calendar(2019))
```

※範例8-2f

假設銀行是以「每日餘額 * 日利率」計算每日的利息，然後累加這些利息，每年分兩次給息，分別是 06/20 與 12/20，某人存提款記錄如下，請計算 06/20 可得到多少利息，假設年利率是 1%。

編號	日期	存款	提款	餘額	備註
1	03/06	20000		20000	
2	03/16	30000		50000	
3	04/24		10000	40000	
4	06/20				

👉 執行結果

```
65
10
39
57
121.36986301369863
```

👉 運算思維

1. 共分 3 段處理利息，第 1 段是 20000 元，日期是 03/06 ~ 03/16 共 10 天，每日利息是 $20000 \times 0.01 / 365$ ，10 天利息就是 $20000 \times 0.01 / 365 \times 10$ 。
2. 同理，處理第 2、3 兩段。
3. 然後累加以上 3 段的利息，就是 06/20 的利息。
4. 將有存提款日期的餘額使用二維串列儲存如下

```
a=[['03/06',20000],['03/16',50000],['04/24',40000],['06/20',0]]
#存提款餘額記錄
b=[0,31,28,31,30,31,30,31,31,30,31,30,31] #每月天數
s=0 #利息總和
m,d=map(int,a[0][0].split('/')) #將日期分解為m,d
sum1=0
#計算第1筆紀錄是一年裡的第幾天，本例是03/06
#先加3月前面完整月份的天數
for j in range(1,m):
    sum1=sum1+b[j]
#加上3月的天數
sum1=sum1+d
print(sum1)#65
#使用迴圈，逐一計算每一區段的利息
for i in range(1,4):
    m,d=map(int,a[i][0].split('/'))
```

```

#計算m/d是第幾天
sum2=0
for j in range(1,m):
    sum2=sum2+b[j]
sum2=sum2+d
sum=sum2-sum1 #計算每兩個日期的間隔天數
print(sum)
#累加每一段日期的利息
s=s+a[i-1][1]*(sum)*0.01/365 #每區段利息=每日餘額*日利率*天數
sum1=sum2#移動參考日期
print (s)

```

⚙️ 路口紅綠燈

路口的紅綠燈如何設計燈號的變化，這一問題要讓電腦處理，也是要先想好如何將資料數位化，這種問題通常將燈號的變化和時間建立函數關係，就稱為時序圖。(將燈號以秒為單位切割，然後記錄每秒的燈號狀態，稱為時序圖)

範例8-2g

紅綠燈程式。假設有單向紅綠燈時序如下，請設計程式完成其功能。

時序	燈號
0	綠燈
1	綠燈
2	綠燈
3	綠燈
4	綠燈
5	黃燈
6	黃燈
7	紅燈
8	紅燈
9	紅燈
10	紅燈

👉 執行結果

```
綠燈  
綠燈  
綠燈
```

👉 運算思維

1. 您要讓程式不間斷執行，就要一個無窮迴圈如下：

```
while 1:
```

2. 要讓程式依照時序無限循環，且不溢位，就要聯想取餘運算『%』。

```
while 1:  
    t=(t+1)% 11#本例共11個時序
```

3. 全部程式如下：

```
import time  
a='綠燈'  
b='黃燈'  
c='紅燈'  
d=''  
t=0  
#無窮迴圈，依照秒數給予燈號  
while 1:  
    if t<=4:  
        d=a  
    elif (t==5 or t==6):  
        d=b  
    else:  
        d=c  
    print(d)  
    time.sleep(1)  
    t=(t+1)% 11
```

4. 這一題，也可以藉助串列，先將所有情況以串列儲存，然後每一個時間點，就輸出其對應值。

```
import time
a=['綠燈','綠燈','綠燈','綠燈','綠燈','黃燈','黃燈','紅燈','紅燈',
  ,'紅燈','紅燈']
t=0
while 1:
    print('%d,%s' %(t,a[t]))
    time.sleep(1)
    t=(t+1)% 11
```

👉 自我練習

1. 觀察十字路口雙向紅綠燈，規劃其時序，寫出其程式。(也就是您要同時輸出兩個方向的燈號)
- ※ 2. 假設某一車站南下火車時刻表如下：

編號	發車時間	停靠站
1	06:30	台中、台南、左營
2	06:50	新竹、台中、台南、左營
3	07:10	台中、左營
4	07:20	桃園、台中、嘉義、台南、左營

請設計一個程式，此程式都會依照時間，顯示下一班火車時刻表與停靠站。例如，現在時間是 06:28，就是輸出『06:30 開往台中、台南、左營』

3. 依時間調整的紅綠燈。
假設某一紅綠燈依時間規劃成四種顯示模式，如下表所示：

模式	時間	紅燈時間	綠燈時間
1	20:00~07:00	閃紅燈	閃黃燈
2	07:00~09:00	30	40
3	09:00~17:00	20	20
4	17:00~19:00	30	40
5	19:00~20:00	20	20

8-3 本章內容摘要

1. 子字串的提取可同子串列的提取方式。例如：

```
a='abcdef'
print(a[0]) #a取第0個
print(a[-1]) #f 取倒數第1個
print(a[-2]) #e取倒數第2個
print(a[0:3]) #abc 取索引0~2
print(a[2:]) #cdef 取索引2開始到結束
print(a[-2:]) #ef 取倒數兩個
print(len(a)) #6 計算長度
print(max(a)) #f 取最大元素
b=list(a) #轉為串列 ['a', 'b', 'c', 'd', 'e', 'f']
print(b)
c="".join(b) #串列轉為字串 abcdef
print(c)
```

2. Python 沒有時間型態，關於時間的計算可使用 map 函數分解如下：

```
a="08:10:20"
h,m,s=map(int,a.split(":"))
print(h,m,s)
```

3. Python 沒有日期型態，關於時間的計算可使用 map 函數分解如下：

```
a="112/03/20"
y,m,d=map(int,a.split("/"))
print(y,m,d)
```

4. 關於電腦對於時間的記錄有兩種表示，分別是 1970/1/1 日起的秒數與給人類看的時間格式，如以下程式。

```
import time #載入時間模組
print(time.time()) #1657865839.2869213 #傳回1970/1/1起的秒數
print(time.asctime()) #傳回現在時間Wed Nov 6 15:57:20 2022
```

課後評量

(填充題，請寫出程式執行結果) 以下試題為「全國高級中等學校 111 學年度商業類學生技藝競賽程式設計組學科試題」，剛好與本書範圍相同，所以本書將題目轉為填充題如下：

題號	題目	執行結果
1	<pre>a, b = 11, 22 print(a b)</pre>	
2	<pre>print(6 & 4 9)</pre>	
3	<pre>print(1 ^ 1 ^ 1, 0 ^ 0 ^ 1, sep="")</pre>	
4	<pre>print(sum(range(10)) % 4)</pre>	
5	<pre>print((99 % 17) % 4)</pre>	
6	<pre>a, b = 0, 1 a = a ^ b b = a ^ b a = a ^ b print(a, b)</pre>	
7	<pre>print(4 >> 1) print(8 >> 2) print(16 >> 3) print(31 >> 3) print(32 >> 4)</pre>	
8	<pre>values = [i for i in range(2, 11)] print(sum(values) % 4)</pre>	
9	<pre>c = 0 for i in range(1, 7): c += i % 3 print(c % 4)</pre>	
10	<pre>a = 12 b = a - 5 while a % b > 0: b -= 1 print(b % 4)</pre>	
11	<pre>ans = 0 for i in range(1, 11): ans = ans + ans * i print(ans % 4)</pre>	

12	<pre>j, n = 0, 10 for i in range(1, n + 1): for k in range(i + 2, n - 1): j = j + 1 print(j % 4)</pre>	
13	<pre>j, n = 0, 10 for i in range(1, n + 1): for k in range(i + 2, n - 1): j = j + 1 print(j % 4)</pre>	
14	<pre>total = 0 for i in range(3): for j in range(4): total += 1 print(total % 4)</pre>	
15	<pre>total = 0 for i in range(3): for j in range(i, 4): total += 1 print(total % 4)</pre>	
16	<pre>print(len(range(10)))</pre>	
17	<pre>total, i = 0, 0 while i <= 5: i = i + 1 for j in range(1, i + 1): if j % 2 == 1: total += 1 else: total += 2 print(total % 4)</pre>	
18	<pre>a, s = 33, 0 while a > 0: s += a % 16 a = a >> 2 print(s % 4)</pre>	

19	<pre> a = [92, 85, 80, 100, 78, 41, 34, 90, 60, 59] count_1, count_2 = 0, 0 for i in range(10): if a[i] >= 80: count_1 += 1 if a[i] < 60: count_2 += 1 print((count_1 + count_2) % 4) </pre>	
20	<pre> count = 0 data = [1, 2, 3] data_length = len(data) for i in range(data_length): for j in range(data_length): for k in range(data_length): if data[i] == data[j] or data[i] == data[k]: count += 1 print(count % 4) </pre>	
21	<pre> dp = [0 for _ in range(10)] dp[0], dp[1], dp[2] = 1, 2, 2 for i in range(3, 8): dp[i] = dp[i - 2] + dp[i - 3] print(dp[7]) </pre>	
22	<pre> a = 10 b = a - 1 while a % b != 3: a += 1 b -= 1 print(b % 4) </pre>	
23	<pre> a = [[0] * 4 for _ in range(4)] for i in range(4): for j in range(4): if j % 2 == 0: a[i][j] = 2 else: a[i][j] = 1 for i in range(1, 4): for j in range(1, 4): a[i][j] = a[i - 1][j] + a[i][j - 1] print(a[3][3] % 4) </pre>	

24	<pre>s, i, j = 0, 2, 0 while i <= 5: while j < 3: s += 1 j += 1 i += 1 print(s)</pre>	
25	<pre>a, b, c = 12, 7, 18 while b > 0: c = a % b a = b b = c print(a + b + c)</pre>	

課後習題

1. 選擇題評分。某次考試共有 10 題選擇題，標準答案與 12 位學生答案如下，(1) 請寫程式幫忙評分。(2) 請問每一題答對人數。

標準答案	ACDDBAACCD
1	AAAAAAAAAAA
2	BBBBBBBBBBB
3	CCCCCCCCCCC
4	DDDDDDDDDDD
5	ABCDABCDAB
6	AABBCCDDAA
7	AAABBBCCCD
8	ACDDBAACCD
9	ACDDBAACCC
10	ACDDBAACCD
11	ACDDBAAAAA
12	AAAAAAACCD

- ※ 2. 請寫程式，算出以下資料，(1) 共有多少個連續的『11』？(2) 共有幾群連續『11』？哪一群連續『11』最長。例如：(提示：本題因為連續數字長度不固定，可選用 while 迴圈。)

```
00110001011111011
```

表示共有 5 個連續『11』，共 3 群，第 2 群連續『11』最長，共 3 個連續『11』。

- ※ 3. 簡單計算機。請寫一個程式，可以當作簡單計算機，可以計算輸入兩個數的 +, -, *, /，輸入格式可為「1023+23」、「321-2」、「23*420」、「96/4」等，僅 4 種運算子，但數字長度不限。(此題請不要使用 eval() 函式)(提示：本題可嘗試使用 while 迴圈，亦可使用 split() 分割。)
- ※ 4. 簡易計算機。請寫一個程式，可進行簡單連續計算，且僅含加法。例如：可輸入 23+450+2+2345。(此題請不可使用 eval() 函式)
- ※ 5. 簡易計算機。請寫一個程式，可進行簡單連續計算，且僅含加、減、乘、除法等運算，且一律由左至右運算。例如：可輸入 23+450-2*3/2。(此題請不可使用 eval() 函式)

MEMO

附錄一

商科技藝競賽程式設計試題

技藝競賽是高職生三年成果的最佳技藝展現機會，從 111 年起程式設計職種的題目由 Visual Basic 改為 Python，由考古題發現，只要熟讀本書，就能解出七成以上的題目，所以本書在此分享解題技巧，希望同學們對自己的學習要有信心，且爭取前往參加比賽。其次，競賽題目有點長，本書限於篇幅，僅簡單描述，請同學自行到全國高級中等學校技藝競賽平台 (<https://sci.me.ntnu.edu.tw>) 下載。

Problem A1 (民國年)

請寫一支程式，將輸入的西元年份轉換成民國年份。

測試資料

Sample Input 1	Sample Output 1
1912	1

執行結果

```
input a number:1912
1
```

運算思維

1. 以 a 變數儲存所輸入單筆數值資料。

2. 將 a 值減去 1911。
3. 輸出 a 值，即為所求。

程式列印

```
a=int(input("input a number:"))
a=a-1911
print(a)
```

Problem A2 (閏年(0結束))

西元年份除以 4 不可整除為平年。a normal year。

西元年份除以 4 可整除，且除以 100 不可整除，為閏年。a leap year。

西元年份除以 100 可整除，且除以 400 不可整除，為平年。a normal year。

西元年份除以 400 可整除，為閏年。a leap year

測試資料

Sample Input 1	Sample Output 1
1992	a leap year
1991	a normal year
1700	a normal year
2400	a leap year
0	

執行結果

```
input number:1992
a leap year(2)

input number:1991
a normal year(1)

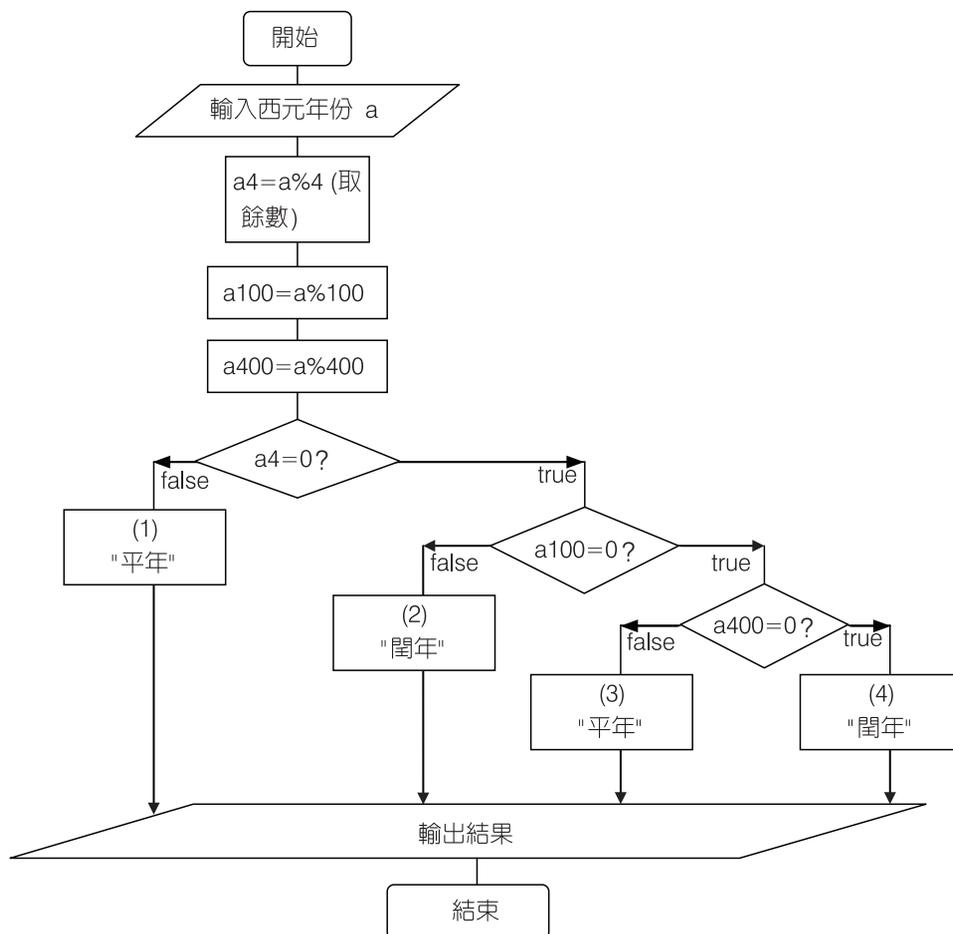
input number:1700
a normal year(3)

input number:2400
a leap year(4)

input number:0
```

👉 運算思維

- 整理題意如下：西元的閏年為每 400 年必須有 97 次閏年，其規劃方式如下：
 - 4 的倍數。依此條件共有 100 次。
 - 於 (1) 的條件，扣掉 100 的倍數。依此條件，共有 96 次。
 - 於 (2) 的條件，再加回 400 的倍數。所以共有 97 次。
 - 以上演算分析，流程圖分析如下：



- 此題測試資料筆數不固定，而是以 0 結束，所以可用不定 while 迴圈讀入資料。

👉 程式列印

```
#先讀一筆資料
a=int(input("input number:"))
#只要讀入資料不為0，繼續讀取資料
while a!=0:
    #print(a)
    a4=a%4 #除以4的餘數
    a100=a%100 #除以100的餘數
    a400=a%400 #除以400的餘數
    #根據流程圖寫出程式
    if a4==0:
        if a100==0:
            if a400==0:
                d='a leap year(4) '
            else :
                d='a normal year(3) '
        else:
            d='a leap year(2) '
    else :
        d='a normal year(1) '
    print(d)
    #重複繼續輸入資料
    a=int(input("input number:"))
```

👉 補充說明

1. 以上閏年的 if 判斷，也可以簡化如下：

```
if (a % 4 == 0 and not(a % 100 == 0) or (a % 400 == 0 )):
    d="a leap year"
else:
    d="a normal year"
```

2. 因為題目規定僅可以使用 Python 內建函式與模組，所以也可以使用 calendar 模組的 isleap() 方法，參考程式如下：

```
import calendar as c
print(c.isleap(1700))
```

Problem B (壞的遙控器)

壞的遙控器上所有按鍵全壞了，只剩下「下一個頻道」按鍵可以用。

目前電視有 200 個頻道，編號為 0~199。給現在正在看的頻道，請問使用者得按幾次「下一個頻道」，使用者才能切換到新的頻道。

👉 測試資料

Sample Input 2	Sample Output 2
199 2	3

👉 執行結果

```
input a b:199 2
3
```

👉 程式列印

```
#讀入2筆資料
d=input("input a b:")
#以空白分割資料
a,b=map(int,d.split(' '))
if b>a : #如果後面數字大於前面數字
    n=b-a #直接減
else: #否則表示該頻道在目前頻道的前面
    n=200-a+b #先按到0(數字200)，再加上b
print(n)
```

Problem C (翻轉數字)

輸入正整數，並將其數字由左到右的次序全部翻轉。

▶ Input File Format

輸入一行，包含一個正整數 N ， $1 \leq N \leq 2^{31}$ 。

▶ Output Format

輸出翻轉過後的數字。前面有 0 的話應消除。

 **測試資料**

Sample Input 3 230400	Sample Output 3 4032
--------------------------	-------------------------

 **執行結果**

```
input a:230400
4032
```

 **程式列印**

```
#讀入1筆0到2的31次方(2147483648)的數字
a=int(input("input a:"))
c=''
#因為數字長度不定，所以使用不定while迴圈，從個位數開始，每次取1個位數
while a>0: #只要a>0則重複迴圈，每次從個位數取1個數字
    b=a%10 #取餘數
    c=c+str(b) #餘數先字串化再串接
    a=a//10 #除以10取整數
print(int(c)) #字串轉數值，前導0就自動消失
```

Problem D (不重複隨機亂數產生器)

產生 N 個 1 到 1000 之間的隨機亂數 $R(N \leq 100, 1 \leq R \leq 1000)$ ，對於其中重複的數字，只保留一個，把相同的數字去掉。然後再把這些不同的數字從小到大排序。

► Input File Format

每筆輸入檔案包含二列，第 1 列為 1 個正整數 N ，表示亂數的個數。第 2 行有 N 個 1 到 1000 之間的隨機亂數，用空格隔開的正整數。

► Output Format

對於其中重複的數字，只保留一個，把相同的數字去掉。計算不重複隨機亂數的個數，同時再把這些不同的數字從小到大排序輸出。

▶ Sample Input 1

11
20 40 32 67 40 20 89 300 404 13 13

提示：20 40 32 67 40 20 89 300 404 13 13，40 20 和 13 為重複的數字。

▶ Sample Output 1

8
13 20 32 40 67 89 300 404

👉 執行結果

```
input number:11
串列資料:20 40 32 67 40 20 89 300 404 13 13
8
[32, 67, 40, 300, 13, 20, 404, 89]
13 20 32 40 67 89 300 404
```

👉 程式列印

```
#讀取資料個數
a=int(input("input number:"))
#讀取連續資料(請複製測試資料來貼，不用自己鍵入資料)
b=input("連續資料:").split(" ") #b為串列
c=[int(b1) for b1 in b] #逐一轉數值
print(c) #確認資料是否正確
d=set(c) #將串列轉set，則重複資料自動被刪除
print(len(d)) #依題目要求，輸出串列長度
e=list(d) #將集合轉回串列
print(e) #確認資料是否正確
e.sort() #由小而大排序資料
#依題目要求，輸出結果
for e1 in e:
    print(e1,end=' ')
```

Problem E (重複隨機亂數統計)

產生 N 個 1 到 1000 之間的隨機亂數 $R(N \leq 100, 1 \leq R \leq 1000)$ ，現在要統計對這些隨機亂數各自出現的次數，正整數按照從小到大的順序輸出統計結果。

► Input File Format

每筆輸入檔案包含二列，第 1 列為 1 個正整數 N ，表示亂數的個數。第 2 行有 N 個 1 到 1000 之間的隨機亂數，用空格隔開的正整數。

► Output Format

每組輸出包含 m 列 (m 為 N 個正整數中不相同數的個數)，按照正整數 (隨機亂數) 從小到大的順序輸。每列輸出兩個整數，分別是正整數 (隨機亂數) 和該數出現的次數，用空格隔開正整數和次數。

► Sample Input 1

```
11
20 40 32 67 40 20 89 300 404 13 13
```

► Sample Output 1

```
13 2
20 2
32 1
40 2
67 1
89 1
300 1
404 1
```

 執行結果

```
13 2
20 2
32 1
40 2
67 1
89 1
300 1
404 1
```

👉 程式列印

```
#讀取資料個數
a=int(input("input number:"))
#讀取連續資料(請複製測試資料來貼,不用自己鍵入資料)
b=input("連續資料:").split(" ")
c=[int(b1) for b1 in b] #逐一轉數值
print(c) #確認資料是否正確
d=set(c) #將串列轉set,則重複資料自動被刪除
print(len(d)) #依題目要求,輸出串列長度
e=list(d) #將集合轉回串列
print(e) #確認資料是否正確
e.sort() #由小而大排序資料
#依題目要求,輸出每筆資料與出現次數
for e1 in e:
    print(e1,c.count(e1)) #資料,出現次數
```

Problem F (字元刪除)

輸入兩個字元字串（為大小寫的英文字母），從第一字元字串中刪除第二個字元字串中所有的字元。

輸入：第一字元字串 = "They are students"，第二個字元字串 = "aeuou"

輸出："Thy r stdnts"

👉 輸出結果

```
a They re students
e Thy r studnts
i Thy r studnts
o Thy r studnts
u Thy r stdnts
Thy r stdnts
```

👉 程式列印

本例測試資料先使用指派，請同學依測試資料格式補齊程式。

```
a="They are students" #來源字串
b="aeiou" #第二個字串
#使用迴圈,將a字串逐一刪除b字串中的每一字元
for i in range (len(b)):
    c=b[i] #逐一取出b字串的每個字元
```


👉 運算思維一

1. 每次處理一組。
2. 抓數字。
3. 抓字母。
4. 輸出此組。

👉 程式列印

```
a='1A1P1C1S3AC14KAD'
i=0
alen=len(a)
b=0
#使用while迴圈，每次處理一組，1A、1P稱為一組
while i<=alen-1:
    #找數字
    b=''
    while i <= alen-1 and a[i].isdecimal():
        b=b+a[i] #串接此數字
        i=i+1 #移動索引
    b=int(b) #將數字轉字串
    #print(b)
    #找字母
    c=''
    while i <= alen-1 and a[i].isupper():
        c=c+a[i] #串接此字元
        i=i+1 #移動索引
    #print(c)
    #輸出此一組
    for j in range(b):
        print(c,end='')
```

👉 運算思維二

1. 每次處理一組。
2. 每組第 1 個字元一定是數字，抓完繼續探索是否下一個字元是否也是數字，例如 14、或 152。
3. 數字抓完，就是抓 1 個字母，抓完也是繼續探索下一個字元是否字母。
4. 輸出此一組。

👉 程式列印

```

a="1A1P1C1S3AC14KAD"
n=len(a)
p=0
#使用while迴圈，每次處理一組，1A、1P稱為一組
while p<n-1:
    #抓數字
    b=a[p] #每組第1個字元一定是數字
    x=eval(b)#轉為數字
    #繼續嘗試找數字
    while ord(a[p+1])>=48 and ord(a[p+1])<=57: #探索下一個字元是否數字
        p=p+1 #進入迴圈，表示此為數字，所以移動索引，準備提取此字元
        b=a[p] #提取此字元
        y=eval(b) #轉為數字
        x=10*x+y #累加此數字
    #沒繼續找到數字，也要移動索引
    p=p+1
    c=a[p] #字母
    #繼續嘗試找字母
    while p+1<=n-1 and ord(a[p+1])>=65 and ord(a[p+1])<=90:
        c=c+a[p+1] #串接此字元
        p=p+1
    #沒繼續找到字母，也要移動索引
    p=p+1
    #輸出此一組
    for i in range(x):
        print(c,end="")

```

👉 補充說明

此一 while 迴圈技術，是 APCS 與技藝競賽常考試題，請大家多練習，才能熟練結構化程式設計——不能任意使用 goto 的技術。

👉 自我練習

1. 請寫一個程式，可以處理以下算術運算。本例假設數字長度不限，運算子只有「+,-,*,/」，且運算由左而右，且都是正數開頭。本例不可使用 Python 的 eval() 函式。

```
a='3+20+300='
a='300-3+20='
a='20*3/4+100='
```

Problem H (分組反轉字串)

請分組反轉字串，考慮下列字串：

```
"EFUIEHVOAUCQWNCNWVBNXDAHCBWBGIX"
```

本字串長度為 32，如果我們把該字串分成 8 個群組及長度為 4，"EFUI EHVO AUCQ WNCN WVBN XDAH CBWB GIWX"，對每個群組內的字串 (4 個字元) 分別由左到右的次序做反轉 "IUFE OVHE QCUA NCNW NBVW HADX BWBC XWIG"，會得到另一個新字串："IUFEOVHEQCUANCNWN BVWHADXBWBCXWIG" 給定群組數目及字串，請分組反轉字串。提示：

要知道這字串有幾個字串。

它要分成幾組。

每一組字把它由左到右的次序反轉 (倒序輸出)。

測試資料

Sample Input 1

3 ABCEHSHSH

5 FAOETASINAHGRIONATWONOQAONARIO

0

提示：數字 5，是指分成五組，不是每五個字分成一組。

Sample Output 1

CBASHEHSH

ATEOAFGHANISTANOIRAQONOWOIRANO

執行結果

以下僅完成 1 組資料。

```
ABCEHSHSH
ABC
CBA
EHS
SHE
HSH
HSH
['CBA', 'SHE', 'HSH']
CBASHEHSH
```

程式列印

以下僅完成一組資料，完整程式請同學依照測試資料格式自行擴充。

```
a="ABCEHSHSH" #測試資料
print(a)
num=3 #假設分3組
n=len(a)//num #n為每組的長度
b=[] #準備儲存每一組字元
k=0
#依照分組方式，逐一完成每一組的反轉
for i in range(num):
    #依序取每一組
    c=''
    for j in range(n):
        c=c+a[k]
        k=k+1
    #輸出此組
    print(c)
    #將此組反轉
    d=''
    for j in range(n-1,-1,-1):
        d=d+c[j]
    #輸出此組
    print(d)
    #串接此組
    b.append(d)
print(b) #輸出串列，確認結果
#將串列轉為字串
strb = "".join(b)
#依照題目要求，輸出反轉後的字串
print(strb)
```

👉 自我練習

1. 以上僅完成一組資料，請依照題目輸入格式，完成輸出要求。

Problem I (字元出現次數)

給你一列文字，請你找出各字元出現的次數。

▶ Input File Format

每筆測試資料一列。每列最大長度為 1000。

▶ Output Format

對每一列輸入，請輸出各字元的 ASCII 值及其出現的次數；請根據出現的次數由小到大輸出；如果有 2 個以上的字元有相同的次數，則 ASCII 值較大的先輸出。參考 Sample Output。

👉 測試資料

Sample Input 3	Sample Output 3
AAABBc32433@@	99 1
	52 1
	50 1
	66 2
	64 2
	96 3
	65 3
	51 3

👉 執行結果

```
99 1
52 1
50 1
66 2
64 2
96 3
65 3
51 3
```

👉 運算思維

1. 以下程式可初始化二維陣列，且逐一新增每一列的資料。

```
b=[]
for i in range(4):
    b.append([]) #新增一行
    for j in range(3):
        b[i].append(j) #逐一新增每一欄
print (b)
```

```
[[0, 1, 2], [0, 1, 2], [0, 1, 2], [0, 1, 2]]
```

👉 程式列印

```
a= "AAABBC32433@@```" #測試資料
b=set(a) #字串轉set
blen=len(b) #求集合長度
print(b)
d=[] #準備儲存每一字元出現次數
i=0
#使用迴圈計算每一個字元出現的次數
for b1 in b:
    c=a.count(b1) #傳回此字元出現次數
    print(b1,ord(b1),c)
    d.append([]) #新增一行
    #逐一新增每一欄，分別是該字元ASCII、該字元出現次數
    d[i].append(ord(b1)) #該字元ASCII
    d[i].append(c) #該字元出現次數
    #本例以字元出現少者優先輸出，字元相同時，再以ASCII 高的優先
    #所以新增以下排序欄位，作為排序依據
    #以上排序第二順位ASCII與第一順位次數相反，所以第二順位ASCII要取顛倒
    #ASCII最高是255，所以取256
    d[i].append(c*256+(256-ord(b1))) #此欄位為排序依據
    i=i+1
    #以上每筆記錄，三個欄位分別是該字元ASCII、出現次數、排序用
print(d)
#將二維串列依照新增的排序欄位，由小而大排序，排序請看範例6-5a
for i in range(0,blen-1):
    for j in range(0,blen-i-1):
        if d[j][2]>d[j+1][2]:
```

```

        for k in range(3): #條件成立時，3個欄位都要交換
            d[j][k],d[j+1][k]=d[j+1][k],d[j][k]
print(d)
#已經排序完成，逐一輸出每個字元ASCII與該字元出現次數
for i in range(blen):
    print(d[i][0],d[i][1]) #ASCII,次數

```

Problem J (質數)

輸出給定兩個數字之間的所有質數。

👉 執行結果

```

1000003
1000033
1000037
1000039
1000081

```

👉 程式列印

以下僅找出一筆資料範圍的質數。實際測試資料含指定資料個數，請同學自行用迴圈完成。

```

#指派質數範圍
a=1000000
b=1000090
#使用迴圈逐一檢視範圍內每一個數字是否質數
for i in range(a,b+1) :#a,b兩者都含
    n=int(i**(1/2)) #每一個數字所需檢查因數的上限
    flag=True
    #逐一檢查每一個數字是否有因數
    for j in range (2,n+1):
        if i % j ==0: #找到一個就離開
            flag=False
            continue
    if flag: #如果是質數，則輸出此數
        print(i)

```

Problem K (質因數)

寫一個程式算出一個正整數有多少個不同的質因數。例如：
 $45=3*3*5$ ，所以 45 有 2 個質因數 (3 和 5)。

👉 執行結果

2

👉 程式列印

```
a=45
num=0
#使用迴圈逐一找出其因數
for i in range(2,a//2+1):
    n=0
    while( a%i==0 ): #只要i是因數，還要繼續找
        n+=1
        a=a//i
    if(n>0): #代表i是因數
        num=num+1
print(num)
```

Problem L (二進制bit為1)

計算 1 到 N 之間整數的二進制表示中，bit 為 1 的數量。

▶ Input File Format

輸入一列，包含一個正整數 N， $1 \leq N \leq 10^{15}$ 。

▶ Output Format

計算 1 到 N 之間整數的二進制表示 bit 為 1 的數量。提示：1...7 的二進制表示為 1、10、11、100、101、110、111，所以 bit 為 1 的數量為 12。

👉 執行結果

[1, 1, 2, 1, 2, 2, 3]
12

👉 程式列印

以下程式，先指派一個測試資料，請同學自己擴充為題目要求的輸入格式。

```
a=7 #待求解整數
b=[] #用來儲存每個整數1的個數
#使用迴圈逐一找出每個整數二進位1的個數
for i in range(1,a+1): #列出每個數字
    n=0
    c=i
    #逐一將每個數字轉為二進位，並累加1的個數
    while c>0: #只要c>0 則重複迴圈
        d=c % 2
        n=n+d #累加1的個數，不要判斷，因為判斷耗時
        c=c//2
    b.append(n) #以串列儲存每個整數1的個數
print(b) #輸出串列
print(sum(b)) #累加此串列元素（每個整數所含1的個數），即為所求
```

MEMO

附錄二

本書中英文詞彙索引對照表

英文	中文	頁碼
Advanced Placement Computer Science 縮寫為 APCS	大學程式設計先修檢測	6-59
Argument	引數	7-18
Binary Search	二分搜尋法	6-48
Bubble Sort	氣泡排序法	6-36
Build-in Function	內建函式 (有些書翻譯成公用函式)	7-2
Counting Sort	計數排序法	6-41
dict	字典	6-51
frozenset	凍結集合	6-57
Global Variable	全域變數	7-26
list	串列	6-2
List Generator	串列生程式	6-33
Local Variable	區域變數	7-26
Module	模組 (有些書翻譯成套件或類別)	7-2
Parameter	參數	7-18
Search	搜尋	6-46
Sequential Search	循序搜尋法	6-46
set	集合	6-55
Sort	排序	6-36
tuple	有些書翻譯為元組，但不貼切，本書不翻譯	6-50

MEMO

國家圖書館出版品預行編目 (CIP) 資料

程式語言與設計 / 洪國勝, 蔡懷文, 蔡懷介, 陳蘊慈編著 . --

初版 . -- 高雄市 : 泉勝出版有限公司, 民 112.04

冊 ; 公分

技術型高級中等學校商業與管理群

ISBN 978-986-99632-4-4(上冊 : 平裝). --

ISBN 978-986-99632-5-1(下冊 : 平裝). --

ISBN 978-986-99632-6-8(全套 : 平裝)

1.CST: 電腦教育 2.CST: 電腦程式設計 3.CST: Python(電腦程式語言) 4.CST: 技職教育

528.831

112002894

十二年國民基本教育

技術型高級中等學校商業與管理群

程式語言與設計 (下冊)

審定字號：技審字第 112018 號

書號：SB002

編著者：洪國勝、蔡懷文、蔡懷介、陳蘊慈

責任編輯：洪國勝

排版封面設計：陳彥慈（芒現）

發行所：泉勝出版有限公司

地址：高雄市左營區華夏路 708 號 17 樓

電話：0900757159 LINE ID：5abook

網址：www.goodbooks.com.tw

Email:aa163677@yahoo.com.tw

訂購方式：電話：0900757159 LINE ID：5abook Email:aa163677@yahoo.com.tw

初版日期：112 年 4 月

版次：初版第一刷

定價：280

ISBN：978-986-99632-5-1

◎書內程式、圖片、資料的來源已盡查明之責且標注出處，若有疏漏，在此致歉，並請通知我們，我們將於再版時修訂。

◎若發現書有缺頁、倒裝、嚴重污損，請直接盡快與本公司聯繫，我們會盡快更換。

◎版權所有，請勿翻印。

