CHAPTER 5

串列

- 何謂串列
- 一維串列
- 使用迴圈操作串列
- 串列常用的函式與方法
- 串列的排序
- 二維串列
- 檢測模擬試題解析

5.1 何謂串列

當一個資料常值要儲存於記憶體時，便需要指定一個變數來存放。但若有 100 個相似的資料要儲存時，那可是要使用 100 個變數來存放，這 100 個變數放在程式敘述中，要維護時是相當麻煩的。若這些變數的用途有關聯，這時候我們可以使用「串列」(List)來取代變數。串列又稱為「清單」，在其它程式語言被稱為「陣列」(Array)。

串列是儲存資料的容器，是一組經過編號的變數，在記憶體中佔用連續的記憶體位址。好像一列台車，每個車斗都標記有連續編號，若要從某個車斗取得或放置物品，只要告知車斗的編號，很快地就能找到該車斗。若將一列台車比喻為一個串列，則台車上的一串串車斗我們稱之為「元素」(element)或稱為「串列項目」，台車編號就是元素的註標值。欲存取串列中某個記憶體的資料內容時，只要告知註標值就會由存放在記憶體中的串列起始位址往下數，即可存取該記憶體的內容。

譬如：下圖是名稱為 score 的串列，含有 score[0]~score[49] 共 50 個串列元素，存放學生成績的整數資料。所有串列元素一起存放在連續的記憶體空間，若要存取串列中的資料，只要告知是第幾個資料(即註標值)，便會自動計算該資料的實際位置。

至於串列中是第幾個元素的表示方式是在該串列名稱後面的中括號內填入該變數所對應的數字編號稱為「註標」(index) 或稱「索引值」。譬如 ary[3] 表示串列名稱為 ary 串列中的第四個元素(註標由 0 開始)。所以設計程式時，若碰到需要連續輸入、處理或輸出相關資料，則使用串列取代變數，只要改變中串列括號內的數字編號即可，如此可減輕對變數逐一命名的困擾，在程式中串列元素配合迴圈使用可縮短程式的長度，使得程式變得精簡且可讀性高。

若串列中只有一組註標，稱為一維串列；有兩組註標，稱為二維串列；有三組註標，稱為三維串列；以此類推…。

5.2 一維串列

5.2.1 一維串列的建立

一維串列的建立方式是以 [ ] 運算子來存放元素資料，元素間再用逗號分隔，語法為：

```
串列名稱 = [元素 1, 元素 2, 元素 3, …]
```

[例] 建立串列名稱 score，其元素內容分別為 72, 98, 86, 76 整數資料。

```
score = [72, 98, 86, 76]
```

串列名稱　　元素資料

說明

1. 串列中各個元素資料的型別可以相同，也可以不相同。
   例：score = [72, 98, 86, 76]　　# 元素皆為整數資料型別
   例：animal = ['cat', 'dog', 'monkey', 'tiger']　　# 元素皆為字串資料型別
   例：data = ['Mary', 45, True]　　# 元素存放的資料型別不相同
2. 串列建立時，也可以不含元素資料。
   例：list1 = [ ]　　# 建立的 list1 串列是空串列

5.2.2 串列的讀取與存放

利用 [ ] 運算子填入註標，便可對串列存取其對應的元素內容；若是使用 [start:end]，便是存取註標 strat 起至註標 end-1 的串列元素。要注意的是，串列中第一個元素的註標是 0，第二個元素的註標是 1，以此類推…。

一. 讀取串列元素

讀取串列元素的語法為：

語法 1： 串列名稱[註標] 語法 2： 串列名稱[start:end:step]

1. **語法 1**：可以讀取註標值所對應的串列元素，註標不能超過範圍，否則程式執行時會產生錯誤。
2. **語法 2**：讀取 strat 起到 end-1 註標之間的串列元素，step 為間隔值。若 strat 省略，預設值為 0；若 end 省略，則預設值為串列長度；若 step 省略，則預設值為 1。

[例] 建立串列 lst1，並進行相關的串列元素讀取操作。(檔名：list1.py)

```
01 lst1 = [10, 20, 30, 40, 50, 60, 70, 80, 90]
02 print(lst1)              # 印出 [10, 20, 30, 40, 50, 60, 70, 80, 90]
03 print(lst1[0])           # 印出 10
04 print(lst1[3])           # 印出 40
05 print(lst1[2:6])         # 印出 [30, 40, 50, 60]
06 print(lst1[:6])          # 印出 [10, 20, 30, 40, 50, 60]
07 print(lst1[0:9])         # 印出 [10, 20, 30, 40, 50, 60, 70, 80, 90]
08 print(lst1[1:9:2])       # 印出 [20, 40, 60, 80]
09 print(lst1[1::2])        # 印出 [20, 40, 60, 80]
10 print(lst1[-1])          # 印出 90
11 print(lst1[-9])          # 印出 10
12 print(lst1[-6:-3])       # 印出 [40, 50, 60]
13 print(lst1[:-2])         # 印出 [10, 20, 30, 40, 50, 60, 70]
14 print(lst1[10])          # 錯誤，註標超過範圍
```

說明

1. 第 2 行：印出 lst1 串列所有元素內容。
2. 第 3~4 行：串列中的第一個元素註標是 0，第二個元素的註標是 1，以此類推…。
3. 第 5~9 行：使用 [start:end:step]，便是存取註標 strat 起，到註標 end-1 的串列元素，step 為間隔值。
4. 第 10~13 行：註標可以為負值，串列倒數第一個元素為 -1，串列倒數第二個元素為 -2，以此類推…。lst1[:-2] 與 lst1[0:-2] 相同。
5. 第 14 行：註標不能超過範圍，否則程式執行時會產生錯誤。

二. 字串的連結

將字串串列的元素使用連結字元，連接成一個長字串的語法如下：

字串 = 連結字元.join(字串串列)

[例] 將 arr2 字串串列元素連接成一個長字串 st2。(檔名：join.py)

```
01 arr2 = ['苟不教', '性乃遷', '教之道', '貴以專']
02 st2 = ' '.join(arr2)
03 print(st2)              # 印出 苟不教 性乃遷 教之道 貴以專
```

說明

第 2 行：連結字元為一個空格。連結字元會成為連結字串的一部分。

簡 例 (檔名：dynamic.py)

建立空串列後，使用者可輸入整數決定元素的數量，再依序輸入元素的內容。

結 果

```
請輸入 lst 串列的元素數量：5 Enter

請依序填入各元素的內容...
輸入第 1 個元素內容：56 Enter
輸入第 2 個元素內容：42 Enter
輸入第 3 個元素內容：71 Enter
輸入第 4 個元素內容：26 Enter
輸入第 5 個元素內容：19 Enter

lst 串列的元素內容：
56 42 71 26 19
```

程式碼

檔名： \ex05\dynamic.py

```
01 lst = []
02 count = eval(input('請輸入 lst 串列的元素數量：'))
03
04 print('請依序填入各元素的內容...')
05 for i in range(count):
06     print('輸入第 %d 個元素內容：' %(i+1), end = '')
07     num = eval(input())
08     lst.append(num)
```

```
09
10 print('lst 串列的元素內容：')
11 for x in lst:
12     print(x, end = ' ')
```

說明

1. 第 1,2 行：本例先建立一個 lst = [] 空串列，而串列的長度再由使用者依需求輸入並指定 count 變數存放。
2. 第 5~8 行：使用 for … range() 迴圈循序新增串列元素，來存放使用者所輸入的資料。
3. 第 10~12 行：使用 for … in 串列 迴圈，將 lst 串列中的元素內容一一顯示出來。

5.5 串列的排序

串列的「排序」(Sorting)就是將串列元素的多項資料，由小而大遞增或由大而小遞減來排列。

5.5.1 串列元素由小到大排列

串列的元素，可以按資料值由小到大的排列方式，重新安排元素順序。語法為：

```
串列名稱.sort()
```

[例] 建立串列名稱 score，再對該串列的元素做由小到大的排序。(檔名：sort.py)

```
01 score = [72, 98, 86, 76, 63]
02 score.sort()
03 print(score)                # 印出 [63, 72, 76, 86, 98]
```

5.5.2 串列元素反轉排列

串列的元素，可以按反方向重新排列元素的順序。語法為：

```
串列名稱.reverse()
```

[例] 建立串列名稱 class，再對該串列的元素做由反轉排列。(檔名：reverse.py)

```
01 animal=['dog','cat','monkey','fox','tiger']
02 animal.reverse()
03 print(animal)        # 印出 ['tiger', 'fox', 'monkey', 'cat', 'dog']
```

說明

串列元素若先以 sort() 方法做由小到大的排序，再以 reverse() 方法做反轉排列，就可以做到對串列元素做由大到小的排序。

5.5.3 複製串列排序

使用 sort()方法排序串列，是採就地排序方式，串列經排序後會失去原有的排列順序。若要有排序後的結果，又要保有排序前的原貌，就得使用 sorted() 函式來複製串列並排序。語法為：

```
串列名稱 2 = sorted(串列名稱 1, reverse = True|False)
```

1. 「串列名稱 1」代表排序前的原串列，「串列名稱 2」代表排序後的串列。
2. 若 reverse = True，進行由大到小排序；若 reverse = False，進行由小到大排序。

[例] 建立串列名稱 animal，對 animal 串列的元素做由小到大的排序，排序結果複製給 data 串列，而 animal 留有原順序的排列。(檔名：sorted.py)

```
animal = ['dog','cat','monkey','fox','tiger']
data = sorted(animal, reverse = False)
print('animal =', animal) # animal=['dog','cat','monkey','fox','tiger']
print('data =' , data)    # data = ['cat','dog','fox','monkey','tiger']
```

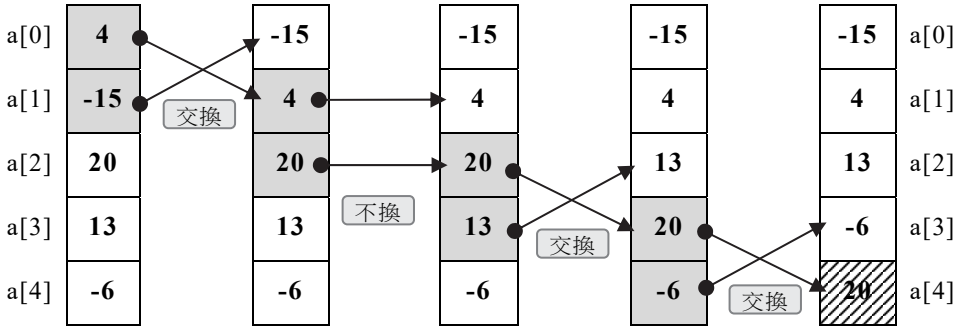
5.5.4 氣泡排序法

使用 sort()方法排序串列，固然很方便，但無法得窺串列元素間依序排列的原理及完整過程。在各種程式語言所使用的串列(陣列)元素的排序方法中，以氣泡排序法最常見。氣泡排序法是採用兩相鄰串列元素的元素值做比較，使元素的元素值由左而右排列時。若是做遞增排列時，元素值較小者排前面，元素值較大者排後面。處理的方式是由左而右進行兩兩比較，當左邊元素的元素值比右邊元素的元素值大時，即進行交換工作。在第一次排列時，元素值最大的元素會移到最右邊；第二次排列時，元素值第二大的元素移到最右邊算過來的第二位；以此類推…。最後，元素值最小的元素會排列在最左邊。

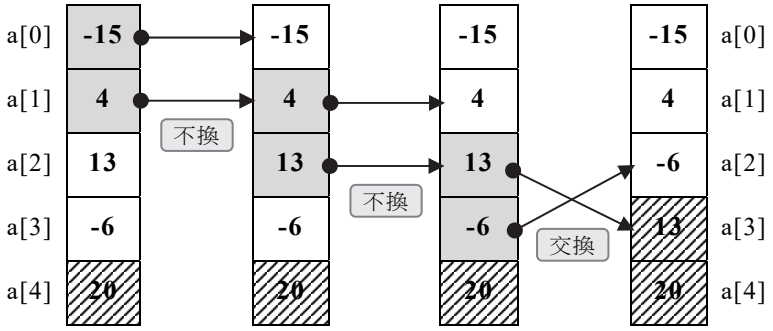
氣泡排序法的排列次數，是串列元素個數減 1。而每次排列的比較次數，是參加排序的元素數減 1。每一次排列比較後，會有一個元素值被放至正確的元素位置。例如有五個元素的串列以氣泡排序法排列，會需要排列 4 次(5-1)，比較次數為 10 次(4+3+2+1)。

以 a = [4, -15, 20, 13, -6] 整數串列為例，說明遞增排列的氣泡排序法原理，如下：

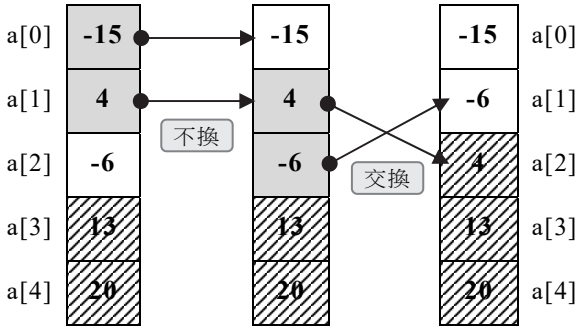
1. 第一次排列 ：
   串列 5 個元素中，兩相鄰元素值互相比較，比較後小者放前面、大者放後面，共比較 4 次，最後找出最大數 20 放至最後面的 a[4] 元素內。



2. 第二次排列 ：
   前面 4 個元素中，兩相鄰元素值互相比較，比較後小者放前面、大者放後面，共比較 3 次，最後找出第二大數 13 放至倒數第二個的 a[3] 元素內。

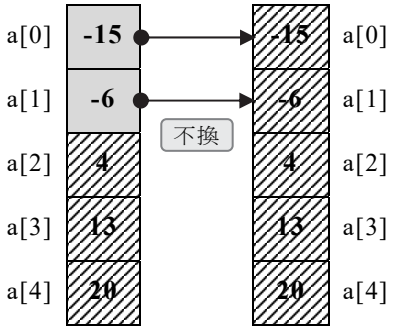


3. 第三次排列 ：
   前面 3 個元素中，兩相鄰元素值互相比較，比較後小者放前面、大者放後面，共比較 2 次，找出第三大數 4 會被放至 a[2] 元素內。



4. 第四次排列 ：

前面 2 個元素中，兩相鄰元素值互相比較，比較後小者放前面、大者放後面，共比較 1 次，其第四大數 -6 會被放至 a[1] 元素內，而最小的數會被放至 a[0] 元素內。



簡 例 (檔名：bubble.py)

將 a = [4, -15, 20, 13, -6] 整數串列，使用氣泡排序法由小到大遞增逐次排列，並顯示驗證每一次排列的結果。

結 果

```
排  序  前 : a[0] =  4  a[1] =-15  a[2] = 20  a[3] = 13  a[4] = -6
第 1 次排列: a[0] =-15  a[1] =  4  a[2] = 13  a[3] = -6  a[4] = 20
第 2 次排列: a[0] =-15  a[1] =  4  a[2] = -6  a[3] = 13  a[4] = 20
第 3 次排列: a[0] =-15  a[1] = -6  a[2] =  4  a[3] = 13  a[4] = 20
第 4 次排列: a[0] =-15  a[1] = -6  a[2] =  4  a[3] = 13  a[4] = 20
```

程式碼

檔名： \ex05\bubble.py

```
a = [4, -15, 20, 13, -6]
print('排  序  前 :', end = '')
for i in range(5):
    print('  a[%d] =%3d' %(i, a[i]), end = '')

for loop in range(1, 5):
    for index in range(0, (5-loop)):
        if a[index] > a[index+1] :
            temp = a[index]
            a[index] = a[index+1]
            a[index+1] = temp
    print()
    print('第 %d 次排列:' %loop, end = '')
    for j in range(5):
        print('  a[%d] =%3d' %(j, a[j]), end = '')
```

說明

1. 第 1 行：建立一個 a 整數串列，元素值為 [4, -15, 20, 13, -6]。
2. 第 3~4 行：顯示 a 串列排序前各元素的內容。
3. 第 6~15 行：執行氣泡排序法，共進行了四次排列。
4. 第 7~11 行：每一次排列的元素內容比較運算，將較大的數值移至註標較大的元素。
5. 第 13~15 行：每一次的排列比較完畢，將該次排列的結果顯示驗證。

5.6 二維串列

二維串列的註標有兩組，第一組註標稱為「列」(row) ，第二組註標稱為「行」(column)。凡是能以表格方式呈現的資料，都可以使用二維串列，如：座位表、課表…。二維串列中若每一列的個數都相同，就構成了一個矩陣串列。二維串列的建立語法：

```
串列名稱 = [[元素 00, 元素 01, 元素 02, …], [元素 10, 元素 11, 元素 12, …],
            [元素 20, 元素 21, 元素 22, …], ……]
```

[例] 建立宣告一個名稱為 lst 的二維串列，其串列大小為 3 列 4 行，元素存放整數資料。取得 lst 串列元素內容，使用 len() 測得串列大小。(檔名：2list.py)

```
01 lst = [[4,8,5,9],[13,16,19,15],[28,25,29,24]]
02 print(lst)            # 印出 [[4,8,5,9],[13,16,19,15],[28,25,29,24]]
03 n1 = len(lst)         # n1←3
04 print(lst[1])         # 印出 [13,16,19,15]
05 n2 = len(lst[1])      # n2←4
06 print(lst[1][2])      # 印出 19
```

說明

1. 第 1 行：所建立的二維串列元素內容如下表所示。

	第 1 行[0]	第 2 行[1]	第 3 行[2]	第 4 行[3]
第 1 列[0]	4	8	5	9
第 2 列[1]	13	16	19	15
第 3 列[2]	28	25	29	24

2. 第 3 行：用 len(lst) 函式取得 lst 串列的大小，會取得 lst 串列第一維的大小，而第一維的串列元素內容為三列串列，分別為 lst[0]、lst[1]、lst[2]，故會傳回 3 指定給 n1 變數。

3. 第 4 行：lst[1] 為 lst 串列的第 2 列串列(如上表)，其 lst[1] 串列內容為 [13,16,19,15]。
4. 第 5 行：用 len(lst[1]) 函式取得 lst[1] 串列的大小，而該串列的含有 lst[1][0]、lst[1][1]、lst[1][2]、lst[1][3] 四個元素。
5. 第 6 行：lst 串列的元素內容分別如下：
   lst[0][0] = 4 、 lst[0][1] = 8 、 lst[0][2] = 5 、 lst[0][3] = 9
   lst[1][0] = 13、 lst[1][1] = 16、 lst[1][2] = 19、 lst[1][3] = 15
   lst[2][0] = 28、 lst[2][1] = 25、 lst[2][2] = 29、 lst[2][3] = 24

簡 例 (檔名：readList.py)

使用迴圈從已初始化的串列中讀取所有元素的資料。

結 果

```
lst = [[4, 8, 5, 9], [13, 16, 19, 15], [28, 25, 29, 24]]

lst[0][0] =  4  lst[0][1] =  8  lst[0][2] =  5  lst[0][3] =  9
lst[1][0] = 13  lst[1][1] = 16  lst[1][2] = 19  lst[1][3] = 15
lst[2][0] = 28  lst[2][1] = 25  lst[2][2] = 29  lst[2][3] = 24
```

程式碼

檔名： \ex05\readList.py

```
01 lst = [[4,8,5,9],[13,16,19,15],[28,25,29,24]]
02 print('lst =', end = ' ')
03 print(lst)
04 for i in range(len(lst)):
05     print()
06     for j in range(len(lst[i])):
07         print('lst[%d][%d] = %2d' %(i, j ,lst[i][j]), end = '  ')
```

說明

1. 第 1 行：建立 lst 串列，元素內容同上例。
2. 第 4~7 行：用 len(lst) 函式取得 lst 串列第一維的大小，因第一維元素為 lst[0]、lst[1]、lst[2]，故 len(lst) = 3，所以 for 外層迴圈的 i 變數依序為 0、1、2。
3. 第 6~7 行：當 i = 0 時，len(lst[0]) = 4，故 for 內層迴圈的 j 變數依序為 0、1、2、3。當 i = 1 時，len(lst[1]) = 4，故此次 for 內層迴圈的 j 變數仍依序為 0、1、2、3。當 i = 2 時，len(lst[2]) = 4，j 變數仍依序為 0、1、2、3。

[例] 用串列生成器建立二維串列 arr，使其內容為[[0, 0], [0, 0], [0,0], [0, 0]]。
(檔名：list8.py)

```
01 A = [0 for x in range(2)]
02 print(A)                          # 印出 [0,0]
03 arr = [A for y in range(4)]
04 print(arr)                        # 印出 [[0,0],[0,0],[0,0],[0,0]]
```

說明

1. 第 1 行：使用串列生成器建立一維串列，並使元素內容為 [0, 0]。
2. 第 3 行：以已建立的一維串列 A 的元素註標做為 arr 串列的元素行註標，而 0~3(4-1) 為列註標。即使
   arr = [A[0], A[1], A[2], A[3]] = [[0, 0], [0, 0], [0, 0], [0, 0]]。
3. 若 A = [(2x+1) for x in range(2)]; arr = [A for y in range(4)]，則
   A = [1, 3]，arr = [[1, 3], [1, 3], [1, 3], [1, 3]]。
4. 若 arr = [[0 for x in range(2)] for y in range(4)]，則
   arr = [[0, 0], [0, 0], [0, 0], [0, 0]]。

簡例 (檔名：score.py)

下表為四位考生參加資優評選的成績，請完成空白部分的數據。

編號	語文	數理	智力	總分
1	87	64	88	
2	93	72	86	
3	80	88	89	
4	79	91	90	
平均				

結果

```
編號   語文   數理   智力   總分
=======================
  1     87     64     88    239
  2     93     72     86    251
  3     80     88     89    257
  4     79     91     90    260
平均   84.8   78.8   88.2
```

程式碼

檔名： \ex05\score.py

```
01 no = [1,2,3,4]                    # 編號
02 score = [[87,64,88],[93,72,86],[80,88,89],[79,91,90]]        # 成績
03 print('編號    語文    數理    智力    總分')
04 print('================================')
05 for i in range(len(no)):
06     print('%2d' %no[i], end = '      ')
07     hSum = 0
08     for j in range(len(score[i])):
09         print('%3d' %score[i][j], end = '    ')
10         hSum += score[i][j]
11     print('%3d' %hSum)
12
13 print('%s' %'平均', end = '   ')
14 for j in range(3):
15     vSum = 0
16     for i in range(len(no)):
17         vSum += score[i][j]
18     print('%4.1f' %(vSum/len(no)), end = '   ')
```

說明

1. 第 1 行：使用一維串列 no 存放編號。
2. 第 2 行：使用二維串列 score 存放 語文、數理、智力 三科目成績。
3. 第 5~11 行：使用兩層 for 迴圈將 score 二維串列元素的內容顯示。內層迴圈用 hSum 變數累加每一個編號(水平列)各科的成績總分，並在內層迴圈結束後顯示出來。
4. 第 14~18 行：使用兩層 for 迴圈計算每一科(垂直行)的分數總和 vSum。在內層迴圈結束後，用 vSum/len(no) 計算該垂直行的分數平均並顯示出來。

5.7 檢測試題解析

題 目 **(一)** (APCS 檢測試題)

輸入 5 個學生資料，資料有姓名、國文、英語成績，計算學生的平均分數後，學生的所有資料再依平均分數由高而低對應排列。

結 果

```
輸入第 1 個學生姓名： 王大維 Enter↵
輸入國文成績： 92 Enter↵
輸入英語成績： 72 Enter↵
輸入第 2 個學生姓名： 李四端 Enter↵
輸入國文成績： 78 Enter↵
輸入英語成績： 45 Enter↵
輸入第 3 個學生姓名： 林森春 Enter↵
輸入國文成績： 97 Enter↵
輸入英語成績： 84 Enter↵
輸入第 4 個學生姓名： 陳明通 Enter↵
輸入國文成績： 83 Enter↵
輸入英語成績： 73 Enter↵
輸入第 5 個學生姓名： 吳仁康 Enter↵
輸入國文成績： 82 Enter↵
輸入英語成績： 77 Enter↵

姓名     國文  英語  平均
====================================
林森春    97    84    90.5
王大維    92    72    82.0
吳仁康    82    77    79.5
陳明通    83    73    78.0
李四端    78    45    61.5
```

程式碼

檔名： \ex05\test05_1.py

```
name = ['' for x in range(5)]
score = [[0,0,0],[0,0,0],[0,0,0],[0,0,0],[0,0,0]]
# 輸入資料
for i in range(5):
    print('輸入第 %d 個學生姓名：' %(i+1), end=' ')
    name[i] = input()
    print('輸入國文成績：', end=' ')
    score[i][0] = eval(input())
    print('輸入英語成績：', end=' ')
    score[i][1] = eval(input())
    score[i][2] = (score[i][0] + score[i][1]) / 2
# 氣泡排序法
for loop in range(1,5):
    for index in range(0, (5-loop)):
```

```
15            if score[index][2] < score[index+1][2]:
16                Temp = score[index]
17                score[index] = score[index+1]
18                score[index+1] = Temp
19                nTemp = name[index]
20                name[index] = name[index+1]
21                name[index+1] = nTemp
22 # 表列
23 print()
24 print('姓名        國文    英語    平均')
25 print('====================================')
26 for p in range(5):
27      print(name[p], end = '      ')
28      print(score[p][0], end = '      ')
29      print(score[p][1], end = '      ')
30      print(score[p][2])
```

說明

1. 第 1~2 行：建立 name 串列，用來存放使用者所輸入的姓名；建立 score 串列，用來存放使用者所輸入的國文、英語分數成績。
2. 第 4~11 行：依序輸入學生姓名及國文、英語分數成績。其中 score[i][2] 用來存放國文、英語兩科的平均分數。
3. 第 13~21 行：依平均分數的遞減排序，相對應的學生姓名串列 name、成績串列 score 的元素資料隨平均分數排列時同步交換元素內容。
4. 第 24~30 行：顯示兩串列遞減排序後的所有資料。

題目 (二)　　(APCS 檢測試題)

使用串列使 4*3 矩陣順時針旋轉 90 度轉換成 3*4 的矩陣。

結果

```
矩陣旋轉前：
 1  2  3
 4  5  6
 6  7  8
10 11 12

矩陣旋轉後：
10  6  4  1
```

```
11  7  5  2
12  8  6  3
```

程式碼

檔名： \ex05\test05_2.py

```
01 a1 = [[1,2,3],[4,5,6],[6,7,8],[10,11,12]]
02 a2 = [[0,0,0,0],[0,0,0,0],[0,0,0,0]]
03
04 print('矩陣旋轉前：')
05 for i in range(4):
06     for j in range(3):
07         print('%2d' %a1[i][j], end = '  ')
08     print()
09 # a1 矩陣順時針旋轉 90 度的轉換運算成 a2 矩陣
10 for k in range(3):
11     for h in range(4):
12         a2[k][h] = a1[4-h-1][k]
13
14 print()
15 print('矩陣旋轉後：')
16 for i in range(3):
17     for j in range(4):
18         print('%2d' %a2[i][j], end = '  ')
19     print()
```

說明

1. 第 1~2 行： a1=[[1,2,3], [4,5,6], [6,7,8], [10,11,12]] 為旋轉前的 4*3 矩陣內容，而 a2 串列用來存放 a1 矩陣順時針旋轉 90 度轉換後的 3*4 矩陣內容。a2 串列的所有元素內容先預設為 0。

2. 第 10~12 行：a1 矩陣順時針旋轉 90 度的轉換運算成 a2 矩陣。觀察 [0][0]⇨[0][3]、[0][1]⇨[1][3]、[0][2]⇨[2][3]、[1][0]⇨[0][2] …，會發現轉換後第一維註標值等於原第二維註標值，而第二維註標值會由 0~3。

a1 矩陣(4*3)

[0][0] 1	[0][1] 2	[0][2] 3
[1][0] 4	[1][1] 5	[1][2] 6
[2][0] 6	[2][1] 7	[2][2] 8
[3][0] 10	[3][1] 11	[3][2] 12

⇨

a2 矩陣(3*4)

[0][0] 10	[0][1] 6	[0][2] 4	[0][3] 1
[1][0] 11	[1][1] 7	[1][2] 5	[1][3] 2
[2][0] 12	[2][1] 8	[2][2] 6	[2][3] 3

題目 (三)　(APCS 檢測試題)

下列程式片段中，若依序輸入整數 0、1、2、3、4、5、6、7、8、9，

```
arr = [0 for x in range(10)]
for i in range(10):
    arr[(i+2) % 10] = eval(input())
```

則在 arr 串列的元素值依順序為何？

(A) [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]　(B) [2, 0, 2, 0, 2, 0, 2, 0, 2, 0]

(C) [9, 0, 1, 2, 3, 4, 5, 6, 7, 8]　(D) [8, 9, 0, 1, 2, 3, 4, 5, 6, 7]

說明

1. 答案是 (D)，程式碼請參考 test05_3.py。
2. 所輸入的數值存入串列元素的註標順序為：arr[2]、arr[3] 、arr[4] … arr[9] 、arr[0] 、arr[1]。

題目 (四)　(APCS 檢測試題)

經過運算後，下列程式片段的輸出為何？

```
for i in range(100):
    b[i] = i
a[0] = 0
for i in range(1,100):
    a[i] = b[i] + a[i-1]
print(a[50] - a[30])
```

(A) 1275　(B) 20　(C) 1000　(D) 810

說明

1. 答案是 (D)，程式碼請參考 test05_4.py。
2. a[1] = b[1] + a[0] = 1 + 0 = 1
   a[2] = b[2] + a[1] = 2 + 1 = 3
   a[3] = b[3] + a[2] = 3 + 3 = 6
   ⋮
   a[30] = b[30] + a[29] = 30 + 435 = 465
   ⋮
   a[50] = b[50] + a[49] = 50 + 1225 = 1275
   故 a[50] – a[25] = 1275 - 465 = 810

題目 (五) (APCS 檢測試題)

請問下列程式片段輸出為何？

```
for i in range(1, 5):
    A[i] = 2 + i * 4
    B[i] = i * 5
C = 0
for i in range(1, 5):
    if B[i] > A[i]:
        C = C + (B[i] % A[i])
    else:
        C = 1
print(C)
```

(A) 1　(B) 4　(C) 3　(C) 333

說明

1. 答案是 (B)，程式碼請參考 test05_5.py。
2. i=1 , A[1]=6　B[1]=5　, C = 1
   i=2 , A[2]=10　B[2]=10 , C = 1
   i=3 , A[3]=14　B[3]=15 , C = 1+1 = 2
   i=4 , A[4]=18　B[4]=20 , C = 2+2 = 4

題目 (六) (APCS 檢測試題)

定義 arr 為一個串列(list)，串列元素的註標為 0 至 n-1。若要將串列中 arr[0] 的元素移到 arr[n-1]，即 arr[0]→arr[n-1]、arr[1]→arr[0]、arr[2]→arr[1]…，下列程式片段的空白處該填入何種運算式？

```
for i in range(_______):
    hold = arr[i]
    arr[i] = arr[i+1]
    arr[i+1] = hold
```

(A) n+1　(B) n　(C) n-1　(C) n-2

說明

1. i=0 時，arr[0]和 arr[1]交換，也就是 arr[1]左移一位、arr[0]右移一位。i=1 時，arr[1]和 arr[2]交換，也就是 arr[2]左移一位、arr[0]又再右移一位。最後一次迴圈 i=n-2 時，為 arr[n-2]和 arr[n-1]交換，也就是 arr[n-1]左移一位、arr[0]右移一位。
2. i 由 0～n-2，所以應該為 range(n-1)，所以答案是(C)，程式碼請參考 test05_6.py。

題目 (七)　(APCS 檢測試題)

下列程式片段執行過程的輸出為何？

```
arr = [y for y in range(10)]
print(arr)
sum = 0
for i in range(1, 9):
    sum = sum - arr[i-1] + arr[i] + arr[i+1]
print(sum)
```

(A) 44　(B) 52　(C) 54　(D) 63

說明

1. i=1 時-arr[1-1]+arr[1]+arr[1+1]，即 -0+1+2=3。i=2 時-arr[2-1]+arr[2]+arr[2+1]，即 -1+2+3=4。… i=8 時-arr[8-1]+arr[8]+arr[8+1]，即 -7+8+9=10。
2. 3+4+5+6+7+8+9+10=52，所以答案是(B)，程式碼請參考 test05_7.py。

題目 (八)　(APCS 檢測試題)

若 A 是一個可儲存 n 筆整數的串列，且資料儲存於 A[0]~A[n-1]。經下列程式碼運算後，

```
A = […,…,…,…,……]
p = A[0]
q = A[0]
for i in range(n):
    if A[i] > p:
        p = A[i]
    if A[i] < q:
        q = A[i]
```

以下何者敘述不一定正確？

(A) p 是 A 串列資料中的最大值　(B) q 是 A 串列資料中的最小值

(C) q < p　(D) A[0] <= p

說明

1. 答案是 (C)，程式碼請參考 test05_8.py。
2. 當串列元素整數內容皆一樣時，q < p 就不成立了。

題目 (九) (APCS 檢測試題)

若 A 是一個 M*N 的整數串列，如下列程式片段用來計算 A 串列每一列元素的總和。

```
01 A = [[…,……],[…,……],[…,……],……]
02 rowSum = 0
03 for i in range(M):
04     for j in range(N):
05         rowSum = rowSum + A[i][j]
06     print('The sum of row %d is %d.' %(i, rowSum))
```

以下敘述何者正確？

(A) 第一列總和是正確，但其它列總和不一定正確

(B) 程式片段在執行時會產生錯誤(run-time error)

(C) 程式片段有語法上的錯誤

(D) 程式片段會完成執行並正確印出每一列的總和

說明

1. 答案是(A)，程式碼請參考 test05_9.py。
2. 把第 2 行的 rowSum = 0 敘述放到第 3 行與第 4 行之間，程式片段就正確了。

題目 (十) (MTA 模擬試題)

下列程式碼，執行後輸出值為何？

```
01 arr1 = [11, 22]
02 arr2 = [33, 44]
03 arr3 = arr1 + arr2
04 arr4 = arr3 * 2
05 print(arr4)
```

(A) [[22, 44], [66, 88]]

(B) [22, 44, 66, 88]

(C) [11, 22], [33,44], [11,22], [33,44]]

(D) [11, 22, 33, 44, 11, 22, 33, 44]

說明

1. 答案是(D)，程式碼請參考 test05_10.py。
2. arr3 = [11, 22, 33, 44]，arr4 = [11, 22, 33, 44, 11, 22, 33, 44]

題目 (十一) (MTA 模擬試題)

一個名為 emp 的清單中有 150 個員工姓名，最後 10 名為臨時員工。若要顯示不含臨時員工的員工姓名，則這分割清單的表示法如何？

(A) emp[0:-9]　(B) emp[1:-10]　(C) emp[:-10]　(D) emp[:-9]

說明

1. 答案是 (C)。
2. emp[:-10] 與 emp[0:-10] 相同。

題目 (十二) (MTA 模擬試題)

一個名為 colors 的清單中有 256 個顏色值。若要分割這個清單，使顯示由清單的第 2 個顏色值開始，間隔 3 個顏色值後再顯示下一個顏色值，直到最後。則這分割清單的表示法如何？

(A) colors[2::4]　(B) colors[1::4]　(C) colors[::3]　(D) colors[1::3]

說明

1. 讀取串列元素的語法為：串列名稱[start:end:step]。題目是由第 2 個顏色值開始，所以 strat 為 1(起始值為 0)。因為是讀取到最後，所以 end 可以省略。題目是讀取間隔 3 個顏色值後的第 4 個顏色值，所以 step 為 4。
2. 答案是 (B)。

5.8 習題

一. 選擇題

1. 建立串列，以哪種括號來存放元素？
   ① { }　② [ ]　③ < >　④ ( )
2. arr = [17,8,-9,24,-3,20]，請問 arr[4] 是多少？
   ① -3　② 24　③ -9　④ 20
3. word = ['甲', '乙', '丙', '丁', '甲', '甲', '乙', '辛']，則 word[-3] 的內容是什麼？
   ① ‘丁’　② ‘甲’　③ ’乙’　④ ‘丙’
4. word = ['甲', '乙', '丙', '丁', '甲', '甲', '乙', '辛']; n = len(word)，則 n 的值為何？
   ① 8　② 9　③ 7　④ 10