

# C 程式語言 3

# 認識陣列 – 使用多個變數

- 上述表格是小考成績，我們可以宣告5個int整數變數來儲存這5次成績，如下所示：

```
int test1 = 71;
```

```
int test2 = 83;
```

```
int test3 = 67;
```

```
int test4 = 49;
```

```
int test5 = 59;
```

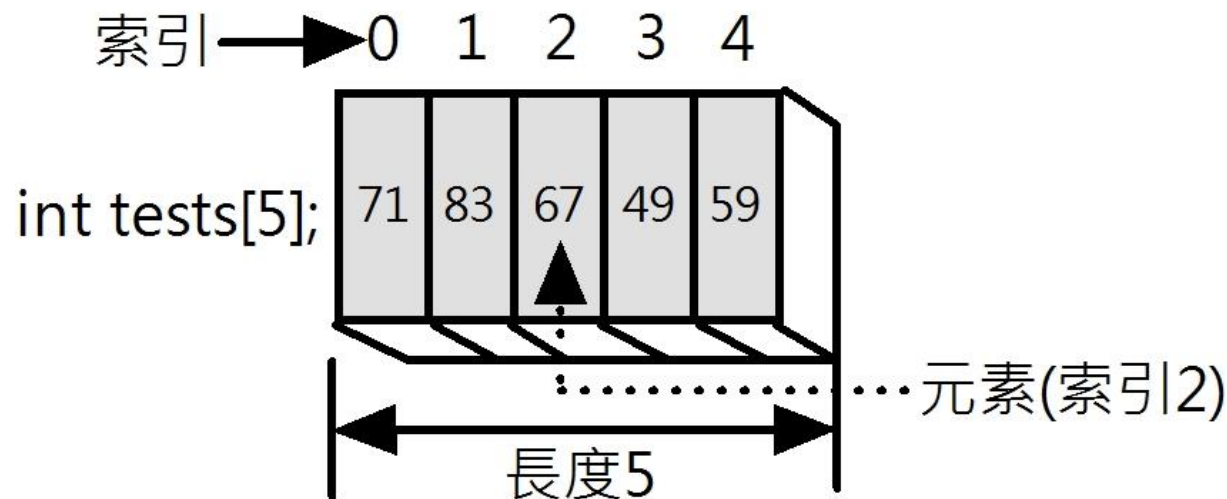
- 上述程式碼宣告5個變數和指定初值，5個數量還好，如果是一班50位學生的成績，我們需要50個變數；如果一個公司有500位員工時，在程式中就需要宣告大量變數，如此會造成程式碼變的十分複雜。

# 認識陣列 – 找出共同特性

- 讓我們再次觀察上述小考成績的5個變數，其擁有的共同特性，如下所示：
  - 變數的資料型態相同都是int。
  - 變數有循序性，擁有順序的編號1~5。

# 認識陣列 - 使用陣列

- 陣列（**Array**）是一種儲存大量循序資料的結構，我們可以將上述相同資料型態的5個**int**變數集合起來，使用一個名稱**tests**代表，如下圖所示：



# 一維陣列

- 陣列是相同型態之元素所組成的集合
- 在 C 語言中，陣列使用前必須先宣告：

一維陣列的宣告格式

資料型態 陣列名稱[個數];

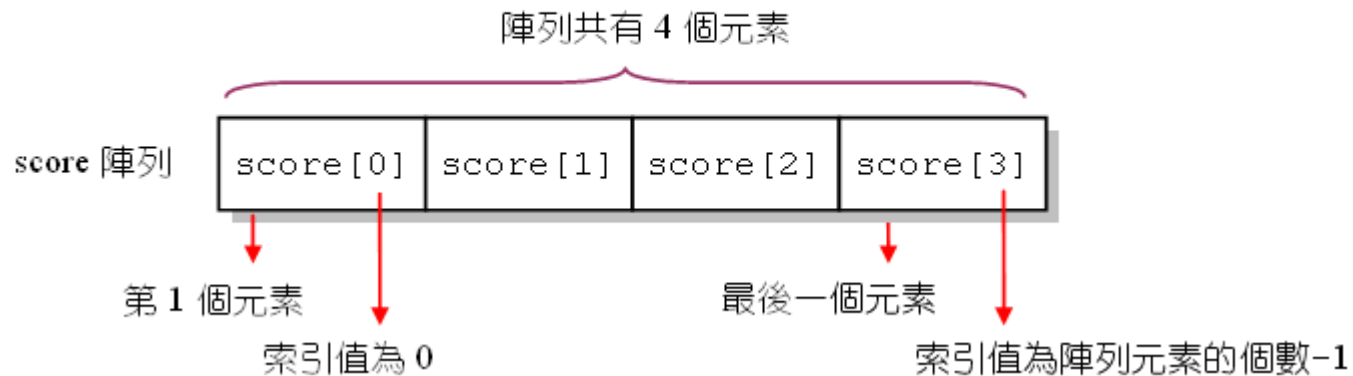
- 下面是一維陣列宣告的範例：

```
int score[4];          /* 宣告整數陣列score，可存放4個元素 */  
float temp[7];        /* 宣告浮點數陣列temp，可存放7個元素 */  
char name[12];        /* 宣告字元陣列name，可存放12個元素 */
```

# 陣列的索引值

- 陣列中的元素是以索引值來標示存放的位置
- 陣列索引值的編號必須由0開始

```
int score[4];
```



# 陣列的範例

- 一維陣列的基本操作：

```
#include <stdio.h>
#include <stdlib.h>
int main(void) {
    int i,score[4];          /* 宣告整數變數i與整數陣列score */
    score[0]=78;             /* 設定陣列的第一個元素為78 */
    score[1]=55;             /* 設定陣列的第二個元素為55 */
    score[2]=92;             /* 設定陣列的第三個元素為92 */
    score[3]=80;             /* 設定陣列的最後一個元素為80 */

    for(i=0;i<=3;i++)
        printf("score[%d]=%d\n",i,score[i]); /* 印出陣列的內容 */

    system("pause");
    return 0;
}
```

# 陣列初值的設定

- 一維陣列初值的設定格式：

一維陣列初值設定的格式

資料型態 陣列名稱[個數n]={初值1, 初值2, ..., 初值n};

- 初值設定的範例：

- **int** score[4]={78, 55, 92, 80};

- **int** score[]={60, 75, 48, 92};      /\* 省略元素的個數 \*/



# sizeof 關鍵字

- 查詢陣列所佔的記憶空間

查詢整個陣列所佔的位元組

```
sizeof (陣列名稱)    /* 查詢陣列所佔的位元組 */
```

# 陣列的應用－最大與最小值

```
#include <stdio.h>
#include <stdlib.h>
int main(void) {
    int A[5]={7,48,30,17,62};
    int i,min,max;
    min=max=A[0];                /* 將max與min均設為陣列的第一個元素 */

    for(i=0;i<5;i++)
    {
        if(A[i]>max)              /* 判斷A[i]是否大於max */
            max=A[i];
        if(A[i]<min)              /* 判斷A[i]是否小於min */
            min=A[i];
    }
    printf("The maximum value of the array element :%d\n",max);
    printf("The minimum value of the array element :%d\n",min);

    system("pause");
    return 0;
}
```

# 二維陣列

- 二維陣列的宣告

二維陣列的宣告格式

資料型態 陣列名稱 [列的個數] [行的個數];


- 二維陣列宣告的範例：

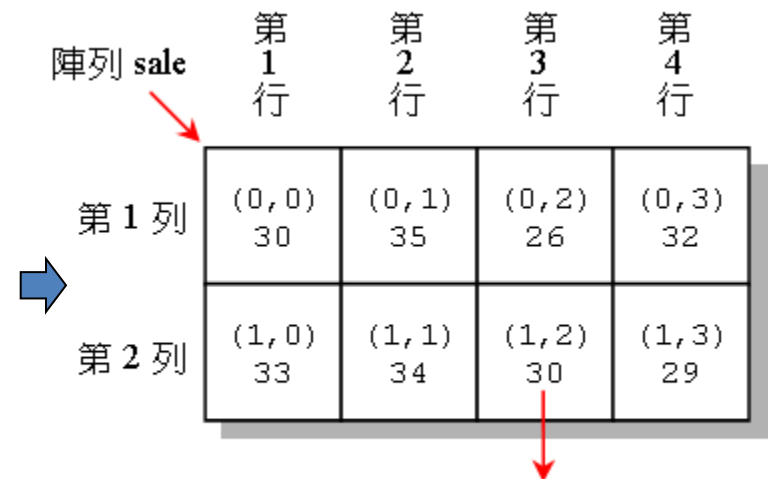
```
int data[10][5];    /* 可存放10列5行個整數 */  
float score[4][3]; /* 可存放4列3行個浮點數 */
```

# 表格與二維陣列



- 二維的表格很適合用陣列來儲存：業績以二維陣列表示

業務員	2004 年銷售量			
	第一季	第二季	第三季	第四季
1	30	35	26	32
2	33	34	30	29



每一格代表一個元素，每個元素皆為 **int** 型態

```
➡ int sale[2][4]={{30,35,26,32},  
                  {33,34,30,29}};
```

# 二維陣列元素的存取

- 利用巢狀迴圈依序輸入二維陣列的元素：

```
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
    int i,j,sale[2][4],sum=0;

    for(i=0;i<2;i++)
        for(j=0;j<4;j++)
        {
            printf"sales%d %d quarter results:",i+1,j+1);
            scanf("%d",&sale[i][j]);          /* 輸入銷售量 */
        }
}
```

# 二維陣列元素的存取

```
printf("***Output***");
for(i=0;i<2;i++)                                /* 輸出銷售量並計算總銷售量 */
{
    printf("\nsales%d quarter results:",i+1);
    for(j=0;j<4;j++)
    {
        printf("%d ",sale[i][j]);
        sum+=sale[i][j];
    }
}
printf("\n2015 results for the total sales volume of %d cars \n",sum);

system("pause");
return 0;
}
```

-The End-