Chapter 10 ： 傳遞陣列至函式

我們可以將一整個陣列的資料傳遞到函式中做處理，首先來看一維陣列的傳遞。

10.1 傳遞一維陣列至函式

格式:

函式定義: void function(int [ ], int);
或 void function(int arrayname[ ], int size);

函式呼叫: function(arrayname, size);

說明:

定義輸入參數爲陣列之函式，需使用[ ]來宣告所傳入的參數爲一陣列。如有使用函式原型宣告，於宣告時可以不指定陣列參數名，於函式定義時指定即可。呼叫函式時，只需用輸入「陣列名」來呼叫。

由於一函式需可處理任何長度的陣列，而不限於某固定長度的陣列，故[ ]中不需定義陣列的長度。然而，我們於函式中經常必須要知道陣列的長度，以便能對輸入陣列之所有元素做運算，故陣列之長度，需當作輸入陣列之另一個伴隨參數。

傳遞陣列的效果相當於之前的「傳位址法」及「傳參考法」，亦即主程式中的陣列與函式中的陣列參數對應到相同的記憶體空間，故於函式之中對所傳入的陣列作改變，亦等於對主程式中之相對陣列作改變。

例:

```
void function(int x[], int size )
{
      x[3] = …..;
}
```

```
void main()
{
    int x[] = {1,2,3,4,5};
    int size = 5;
    function(x, size);
}
```

程式範例: cpp_ex49.cpp

注意要點:

1. 之前提過，一維陣列之名稱爲一指標,其固定指到此陣列的第一個元素，因此，除了上述的傳遞法外，我們亦可使用「指標」的方式來做傳遞。

2. 由於可由函式內更改主程式中陣列的資料值，如不欲此情形發生，可將函式的陣列參數定義爲 const (唯讀指標)。

程式範例: cpp_ex50.cpp　cpp_ex51.cpp

10.2　傳遞多維陣列至函式

格式:

函式定義:　　void function(int [ ][2][3], int);
　　或　void function(int arrayname[ ][2][3], int size);

函式呼叫:　　function(arrayname, size);

說明:

定義有輸入參數爲陣列之函式，使用[ ]來宣告傳入的參數爲陣列，[ ]的數目代表所輸入陣列之維數。如使用函式原型宣告，於原型宣告時可以不給陣列參數名。

函式宣告時，其所有內層之維數都必須要給長度，只有最外層的那一維數(最左邊那一維)不需給長度。

呼叫時，所傳入陣列之爲度數目與內層維數的長度必須與函式之定義相符，不符時會產生錯誤。

例:

```
void function(int x[][3], int size )
{
    x[1][2] = …..;
}

void main()
{
    int x[2][3] = {{1,2,3},{4,5,6}};
    int size = 2;
    function(x, size);
}
```

程式範例: cpp_ex52.cpp

注意要點:

我們亦可使用指標的方式來做多維陣列之傳遞，但較複雜，不建議使用。

程式範例: cpp_ex53.cpp ， cpp_ex54.cpp