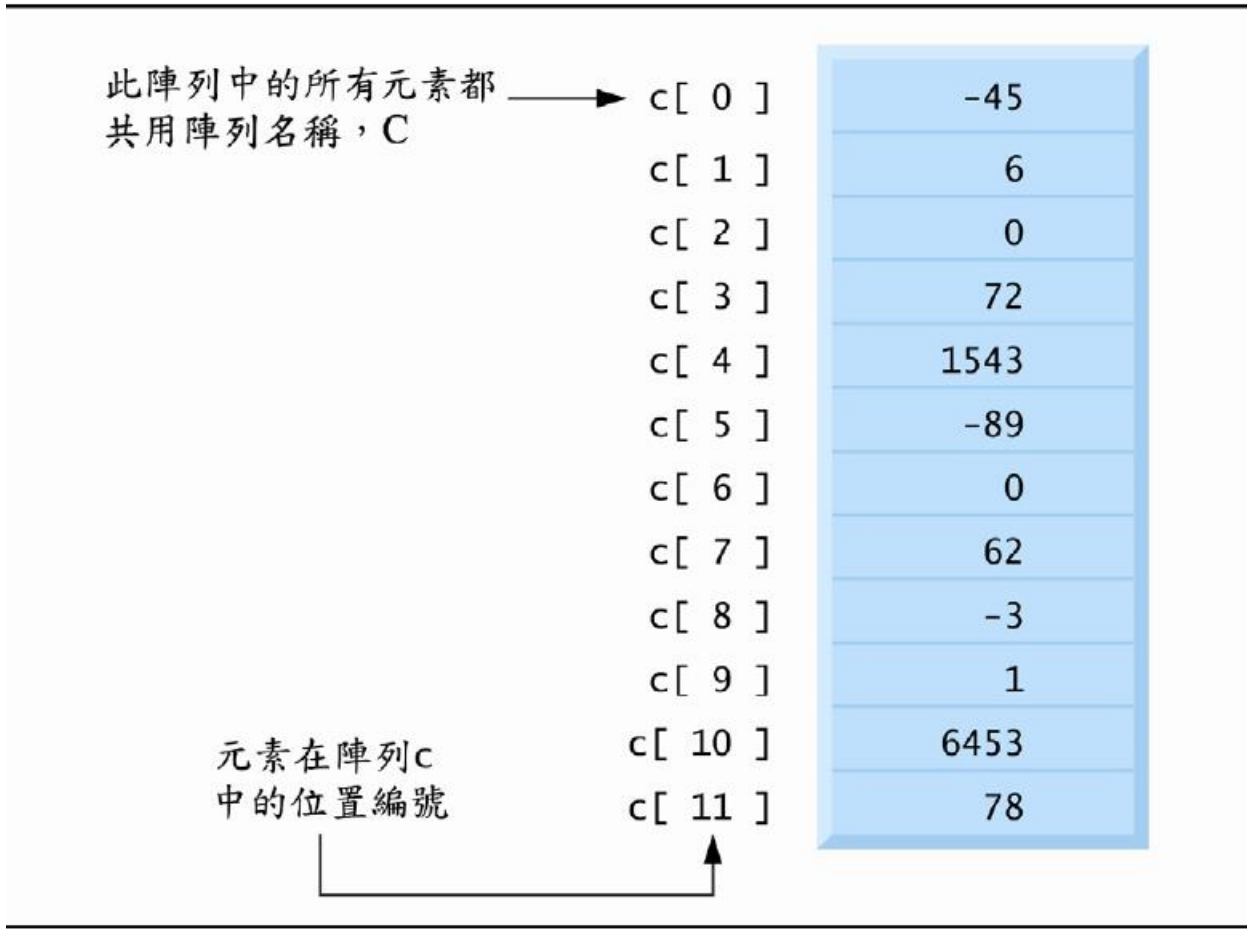
程式設計第6章　陣列一


www.parallax.com
PING)))
SHT11
HTC
10 08
MaxStream
XBee
ZX-BLUETOOTH
UNO
ARDUINO

6.2 簡介

- **陣列 (Arrays)** 是由相同型別的相關資料項所組成的資料結構。
- 陣列是一群具有相同型別的連續記憶體位置。
- int c[12]



- 若要引用陣列的某個位置或元素，我們必須指定陣列名稱，以及此元素在陣列中的**位置編號 (position number)**。
- 中括號中的位置編號正式名稱為**下標 (subscript)**。下標必須是整數或是整數運算式。
- 用來圈住陣列下標的中括號事實上被認為是C的運算子。它們的運算優先順序和函式呼叫運算子 (也就是放在函式後面，用來呼叫函式的括號) 相同。圖6.2列出截至目前為止，我們所介紹過的運算子的運算優先順序和結合性。

運算子	結合性	型別
[] () ++ *(postfix)* -- *(postfix)*	由左至右	最高
+ - ! ++ *(prefix)* -- *(prefix)* *(type)*	由右至左	一元
* / %	由左至右	乘法
+ -	由左至右	加法
< <= > >=	由左至右	關係
== !=	由左至右	相等
&&	由左至右	邏輯 AND
||	由左至右	邏輯 OR
?:	由右至左	條件
= += -= *= /= %=	由右至左	指定
,	由左至右	逗號

圖6.2　運算子的運算優先順序與結合性

6.3 定義陣列

- 陣列會佔用記憶體空間。你會指定每個陣列所需要的元素型別和元素個數，電腦會依此來預留適當大小的記憶體空間。

- 定義

```
int b[ 100 ], x[ 27 ];
```

- 為整數陣列**b**預留100個元素的位置，以及為整數陣列**x**預留27個元素的位置。矩陣分別有0至99與0至26的下標。

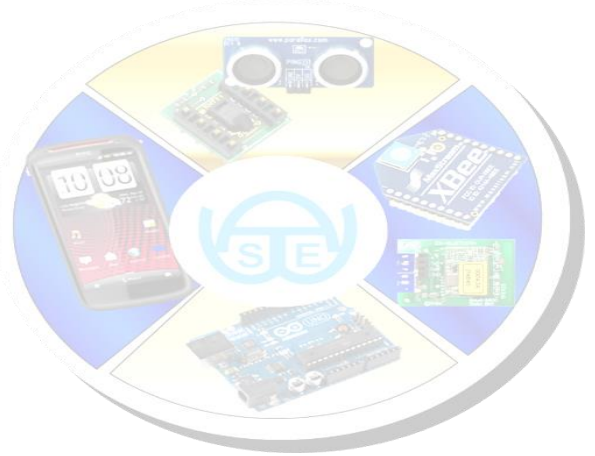
6.4 使用陣列的例子

- 定義一個陣列並且使用迴圈來初始化陣列中的元素
  - 圖6.3的程式使用了一個**for**敘述式，將10個元素的整數陣列**n**的所有元素的初始值設定為零，並且以表列的方式印出這個陣列的內容。

```
1  // Fig. 6.3: fig06_03.c
2  // Initializing the elements of an array to zeros.
3  #include <stdio.h>
4
5  // function main begins program execution
6  int main( void )
7  {
8     int n[ 10 ]; // n is an array of 10 integers
9     size_t i; // counter
10
11    // initialize elements of array n to 0
12    for ( i = 0; i < 10; ++i ) {
13       n[ i ] = 0; // set element at location i to 0
14    } // end for
15
16    printf( "%s%13s\n", "Element", "Value" );
17
18    // output contents of array n in tabular format
19    for ( i = 0; i < 10; ++i ) {
20       printf( "%7u%13d\n", i, n[ i ] );
21    } // end for
22 } // end main
```

```c
#include <stdio.h>

// function main begins program execution
int main( void )
{
   int n[ 10 ]; // n is an array of 10 integers
   size_t i; // counter

   // initialize elements of array n to 0
   for ( i = 0; i < 10; ++i ) {
      n[ i ] = 0; // set element at location i to 0
   } // end for

   printf( "%s%13s\n", "Element", "Value" );

   // output contents of array n in tabular format
   for ( i = 0; i < 10; ++i ) {
      printf( "%7u%13d\n", i, n[ i ] );
   } // end for
} // end main
```

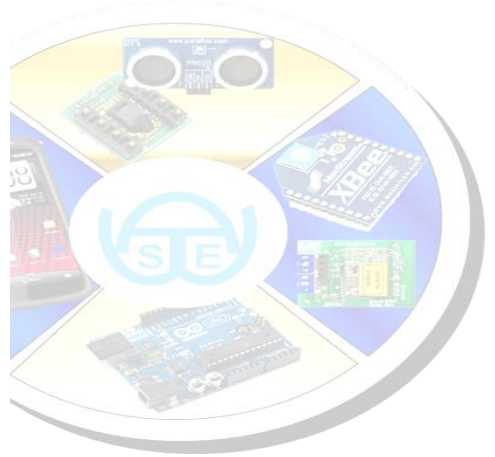
```
Element         Value
      0             0
      1             0
      2             0
      3             0
      4             0
      5             0
      6             0
      7             0
      8             0
      9             0
```



- 在定義中以初始值串列來初始化陣列
  - 陣列中的元素也可以在陣列宣告時就指定初始值，其方式就是在宣告之後**加上一個等號和大括號包住**的一串陣列**初始值 (array initializers)** 串列。

```
1  // Fig. 6.4: fig06_04.c
2  // Initializing the elements of an array with an initializer list.
3  #include <stdio.h>
4
5  // function main begins program execution
6  int main( void )
7  {
8     // use initializer list to initialize array n
9     int n[ 10 ] = { 32, 27, 64, 18, 95, 14, 90, 70, 60, 37 };
10    size_t i; // counter
11
12    printf( "%s%13s\n", "Element", "Value" );
13
14    // output contents of array in tabular format
15    for ( i = 0; i < 10; ++i ) {
16       printf( "%7u%13d\n", i, n[ i ] );
17    } // end for
18 } // end main
```

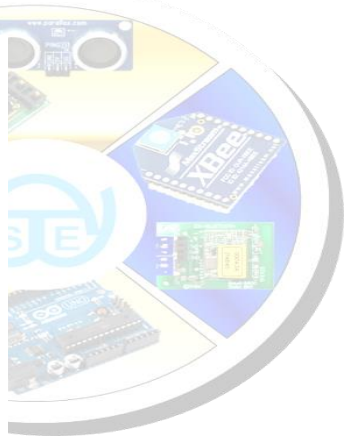
```
Element         Value
      0            32
      1            27
      2            64
      3            18
      4            95
      5            14
      6            90
      7            70
      8            60
      9            37
```



```
1  // Fig. 6.4: fig06_04.c
2  // Initializing the elements of an array with an initializer list.
3  #include <stdio.h>
4
5  // function main begins program execution
6  int main( void )
7  {
8     // use initializer list to initialize array n
9     int n[ 10 ] = { 32, 27, 64, 18, 95, 14, 90, 70, 60, 37 };
10    size_t i; // counter
11
12    printf( "%s%13s\n", "Element", "Value" );
13
14    // output contents of array in tabular format
15    for ( i = 0; i < 10; ++i ) {
16       printf( "%7u%13d\n", i, n[ i ] );
17    } // end for
18 } // end main
```

```
Element         Value
      0            32
      1            27
      2            64
      3            18
      4            95
      5            14
      6            90
      7            70
      8            60
      9            37
```

- 使用一個符號常數來指定陣列的大小，並且以計算來初始化陣列



```
1  // Fig. 6.5: fig06_05.c
2  // Initializing the elements of array s to the even integers from 2 to 20.
3  #include <stdio.h>
4  #define SIZE 10 // maximum size of array
5
6  // function main begins program execution
7  int main( void )
8  {
9     // symbolic constant SIZE can be used to specify array size
10    int s[ SIZE ]; // array s has SIZE elements
11    size_t j; // counter
12
13    for ( j = 0; j < SIZE; ++j ) { // set the values
14       s[ j ] = 2 + 2 * j;
15    } // end for
16
17    printf( "%s%13s\n", "Element", "Value" );
18
19    // output contents of array s in tabular format
20    for ( j = 0; j < SIZE; ++j ) {
21       printf( "%7u%13d\n", j, s[ j ] );
22    } // end for
23 } // end main
```

```
Element         Value
```

```
1  #include <stdio.h>
2  #define SIZE 10 // maximum size of array
3
4  // function main begins program execution
5  int main( void )
6  {
7     // symbolic constant SIZE can be used to specify array size
8     int s[ SIZE ]; // array s has SIZE elements
9     size_t j; // counter
10
11    for ( j = 0; j < SIZE; ++j ) { // set the values
12       s[ j ] = 2 + 2 * j;
13    } // end for
14
15    printf( "%s%13s\n", "Element", "Value" );
16
17    // output contents of array s in tabular format
18    for ( j = 0; j < SIZE; ++j ) {
19       printf( "%7u%13d\n", j, s[ j ] );
20    } // end for
21 } // end main
```

```
Element         Value
      0             2
      1             4
      2             6
      3             8
      4            10
      5            12
      6            14
      7            16
      8            18
      9            20
```

- 求陣列中所有元素的總和
  - 圖6.6的程式將12元素的整數陣列a的所有內含值進行加總。
  - **for**迴圈本體內的敘述式 (第16行) 為我們做了這件事。

```
1  // Fig. 6.6: fig06_06.c
2  // Computing the sum of the elements of an array.
3  #include <stdio.h>
4  #define SIZE 12
5
6  // function main begins program execution
7  int main( void )
8  {
9     // use an initializer list to initialize the array
10    int a[ SIZE ] = { 1, 3, 5, 4, 7, 2, 99, 16, 45, 67, 89, 45 };
11    size_t i; // counter
12    int total = 0; // sum of array
13
14    // sum contents of array a
15    for ( i = 0; i < SIZE; ++i ) {
16       total += a[ i ];
17    } // end for
18
19    printf( "Total of array element values is %d\n", total );
20 } // end main
```

```
1  // Fig. 6.6: fig06_06.c
2  // Computing the sum of the elements of an array.
3  #include <stdio.h>
4  #define SIZE 12
5
6  // function main begins program execution
7  int main( void )
8  {
9     // use an initializer list to initialize the array
10    int a[ SIZE ] = { 1, 3, 5, 4, 7, 2, 99, 16, 45, 67, 89, 45 };
11    size_t i; // counter
12    int total = 0; // sum of array
13
14    // sum contents of array a
15    for ( i = 0; i < SIZE; ++i ) {
16       total += a[ i ];
17    } // end for
18
19    printf( "Total of array element values is %d\n", total );
20 } // end main
```

```
Total of array element values is 383
```

圖6.6　計算陣列中所有元素的總和

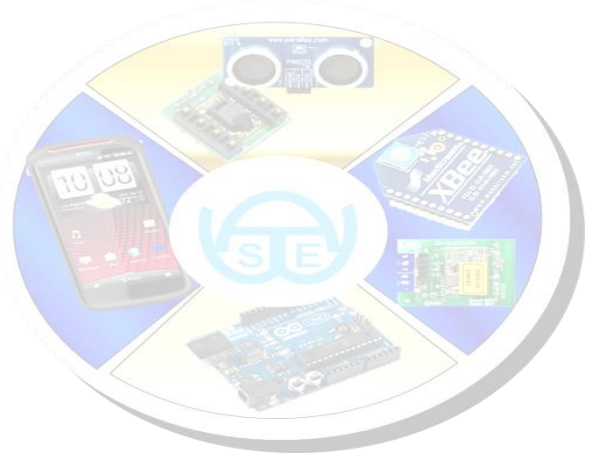
6.5 傳遞陣列給函式

- 若我們想傳遞陣列引數給某個函式的話，只需要指定陣列名稱即可，不必加任何的中括號。

- 練習1.由函式進行輸入陣列資料
  2.由函式找出最大最小值

- main(void)當中僅宣告了一個陣列，之後呼叫函式得到陣列內容，輸出得到的陣列內容
- 函式InputFunc1(int arr[ ])要求使用者輸入資料
- 函式minmax(int arr[ ])可傳入一陣列，並判斷當中的最大最小值各為何?

```
int main(void) {
  int input, ret, k;
  char check_loop;
  int arr[20]={};
  do{
     int flag = 1;
     int i;
     k=InputFunc1(arr);
     printf(" the number of values is %d \n", k);
     for(i=0;i<k;i++)
        printf("  %d  ", arr[i]);
      puts("");
    puts("\n是否繼續輸入(是1)");
    scanf(" %c",&check_loop);
  }while(check_loop == '1');
  system("pause");
  return 0;
}
```



```
int InputFunc1(int arrPtr[]){
    int i = 0, ret;
    puts(" Please input at most 20 integers. ");
        do{
                ret = scanf("%d", &arrPtr[i]);
                if (ret == 1)
                  i++;
        } while (i<20 && ret == 1);
   return i;
}
```