陣列介紹

Array for C language

說明：

陣列(Array), 用以保存多筆相同型態的資料，以中括號表示。宣告陣列時，系統會在記憶體中分配一個連續的空間，因此才能夠透過索引值直接算出資料所在位置。

*注: 陣列的索引值是由 0開始計算, 例如宣告陣列大小為10, 那麼索引範圍應該是：0～9。

1、宣告範例

```c
#include <stdio.h>

int main() {

    //整數陣列宣告, 其大小為10, 未進行初始化
    int array[10];  //相當於有10筆整數為一組的資料(可以保存10筆整數資料於array中)

    //也可以這樣宣告陣列: 指定內容,透過資料數目來決定陣列大小(此處陣列大小為 4)
    int array2[] = { 1, 2, 3, 4 };

    //指定陣列大小 & 初始化
    //注: 初始化資料筆數不可以超過前面宣告的陣列大小10
    int array3[10] = { 1, 2, 3, 4 };     //沒初始化的空間依舊沒有初始化

    //指定陣列大小 & 初始化整個陣列內容
    int array4[10] = {0};    //這並非表示單獨初始化索引 0號位置,而是表示初始化整個陣列都為 0

    return 0;
}
```

2、陣列輸出方式：

```
8   //也可以這樣宣告陣列: 指定內容,透過資料數目來決定陣列大小(此處陣列大小為 4)
9   int array2[] = { 1, 2, 3, 4 };
10
11  //輸出指定索引中的資料(索引範圍0~9)
12  //由於宣告時是整數陣列,因此陣列的內容為整數,需要用整數型態%d來輸出結果
13  printf("array[%d] = %d\n", 0, array2[0]);   //output-> array[0] = 1
14  printf("array[%d] = %d\n", 1, array2[1]);   //output-> array[0] = 2
15  printf("array[%d] = %d\n", 2, array2[2]);   //output-> array[0] = 3
16  printf("array[%d] = %d\n", 3, array2[3]);   //output-> array[0] = 4
17
```

*注:

1、輸出陣列內容時必須指定索引值，不然系統不知道你要的是哪一筆資料。

2、中括號中的索引值為 負數 或 超出了宣告時的大小, 則程式碼會報錯(錯誤訊息-> Error: array(str) Out of range!)

3、利用迴圈輸出陣列內容：

```
8   //也可以這樣宣告陣列: 指定內容,透過資料數目來決定陣列大小(此處陣列大小為 4)
9   int array2[] = { 1, 2, 3, 4 };
10
11  //使用迴圈輸出陣列內容
12  //for迴圈結構請參考迴圈介紹
13  //for迴圈範圍由 0~3, 一共會循環4次, i++表示: 每次循環i都會加一
14  for (int i = 0; i < 4; i++) {
15      printf("array[%d] = %d\n", i, array2[i]);
16  }
```

4、除了以上範例中所使用到的整數型態陣列，也可以宣告其他型態陣列。

只有char陣列需要特別注意(請參考字元字串介紹)，其他陣列型態的使用上都和整數陣列一致。

```
5   //各種型態陣列宣告
6   int array5[10];
7   float array6[10];
8   double array7[10];
9   char array8[10]; //字元陣列即為 字串
```

進階說明：

陣列是連續的記憶體空間，因此系統分配記憶體給陣列時，會通過以下公式來確定空間大小：

型態大小 * 陣列大小 = 實際大小

int array[10]; 的實際大小：
int * 10 = 4Bytes * 10 = 40Bytes。
系統會在記憶體中尋找一片連續的空間且必須超過40Bytes的連續空間，否則系統會告知內存不夠。

高階說明：

所有資料型態在電腦中都是通過記住Address來取得對應　地址Address　中的內容。C語言中能夠直接控制Address的方式只有　指標pointer，這也就意味著所有資料型態在電腦背後在一開始初始化型態格式後，會返回一個地址，這個地址會保存在宣告名稱中，透過這個地址就可以訪問其內容。陣列也是如此，但唯一不同的是，陣列保存的地址僅為陣列0號索引的地址：

int array[10]; //假設系統分配給陣列地址為 13。

其他索引欄位會通過 索引0號的陣列起始地址來計算：

陣列起始地址 + 索引號 * 型態大小 = 指定索引內容位置

array[1] 的地址 = 13 + 1 * 4Bytes = 17 //所以索引1號的實際位置在記憶體中的17