程式設計

第5章 陣列
Array

蘇維宗(Wei-Tsung Su)
suwt@au.edu.tw
564D

目標

定義與使用陣列

瞭解陣列在記憶體中儲存的方式

陣列的應用

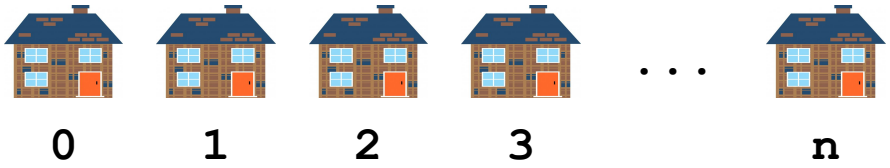
定義與使用一維陣列

One-dimensional Array

陣列

陣列(array)是一種將多筆**相同型別**的資料儲存在**連續記憶體空間**中的資料結構(data structure)。

因為資料依序儲存在連續的記憶體空間中，所以可以透過索引值(index)**隨機存取**陣列中任何一筆資料。(*就像是郵差可以透過門牌號碼送信*)



定義一維陣列

定義一維陣列的方式為

型別　陣列名稱 [陣列大小] ;

例如,

```
1. int score[60];       // 可以儲存60個整數(想想看:佔用多少記憶體空間?)
2. char name[10];       // 可以儲存10個字元(想想看:佔用多少記憶體空間?)
3. int year[];          // 編譯錯誤(沒有指定陣列大小)
```

注意: 陣列屬於**靜態記憶體配置 (static memory allocation)**, 所以程式一執行就會立刻配置所需的記憶體空間且無法再改變大小。

定義一維陣列(給予初始值)

可以在定義一維陣列時同時給予初始值，例如

```
1.  int year[5] = {107,106,105,103};
2.  int year[3] = {107,106,105,103};  // 編譯警告(元素個數超出陣列大小)
3.  char dep[5] = {'C','S','I','E','\0'};
4.  char dep[] = "CSIE";                    //陣列大小由初始 值決定(幾個元素?)
5.  char dep[4] = "CSIE";                   // 編譯成功，但有問題？
```

存取一維陣列

在C語言中，陣列第1筆資料的索引值為0。

透過索引值可以**隨機存取**陣列中任何一筆資料，例如

```
1.  int year[10] = {107, 106, 105, 104, 102}; // 有10個元素，索引值為0~9
2.  printf("%d\n", year[0]); //讀出整數陣列的第1個元素，印出107
3.  printf("%d\n", year[4]); //讀出整數陣列的第5個元素，印出???
4.  year[4] = 103;            // 將103寫入整數陣列的第5個元素
5.  printf("%d\n", year[4]); //讀出整數陣列的第5個元素，印出???
```

想想看

怎麼初始化陣列中所有元素的資料?
怎麼印出陣列中所有元素的資料

例如，將大小為100的陣列元素設定為對應的索引值。例如：

```
arr[0] = 0
...
arr[50] = 50
...
arr[99] = 99
```

```
1.  #define SIZE 100
2.  int i;
3.  int arr[SIZE];
4.
5.  //以迴圈初始化陣列
6.  for(i=0;i<SIZE;i++){
7.    arr[i] = i;
8.  }
9.
10. //以迴圈輸出陣列內容
11. for(i=0;i<SIZE;i++){
12.   printf("%d ", arr[i]);
13. }
14. printf("\n");
```

練習：投擲骰子

輸入整數*n*，以亂數產生器隨機產生1-6的數值模擬投擲1個骰子*n*次並用**陣列**記錄每面出現的次數與機率。

輸入	輸出
12	Face Count Rate 1 3 25% 2 1 8% 3 2 17% 4 2 17% 5 3 25% 6 1 8%

一維陣列在記憶體中的儲存方式

記憶體內的一維陣列

```
char dep[5] = "CSIE";
```

記憶體位址	實際記憶體位址	索引值	資料
dep	x	0	C
dep+1	x + 1*sizeof(char)	1	S
dep+2	x + 2*sizeof(char)	2	I
dep+3	x + 3*sizeof(char)	3	E
dep+4	x + 4*sizeof(char)	4	\0

```
int year[4] = {107, 106, 105};
```

記憶體位址	實際記憶體位址	索引值	資料
year	x	0	107
year+1	x + 1*sizeof(int)	1	106
year+2	x + 2*sizeof(int)	2	105
year+3	x + 3*sizeof(int)	3	?

注意：陣列名稱即為該陣列第 1個元素(索引值為0)的記憶體位址。

如何取得與印出陣列元素的記憶體位址?

```
1.  char dep[5] = "CSIE";
2.  printf("%p %c\n", dep, dep[0]);
3.  printf("%p %c\n", dep+1, dep[1]);
4.  printf("%p %c\n", dep+2, dep[2]);
5.  printf("%p %c\n", dep+3, dep[3]);
6.  printf("%p %c\n", dep+4, dep[4]);
```

記憶體位址	實際記憶體位址	索引值	資料
dep	x	0	C
dep+1	x + 1*sizeof(char)	1	S
dep+2	x + 2*sizeof(char)	2	I
dep+3	x + 3*sizeof(char)	3	E
dep+4	x + 4*sizeof(char)	4	\0

問題:其它取得陣列元素位址的方法?

如何取得與印出陣列元素的記憶體位址? (續)

```
1.  int year[4] = {107, 106, 105};
2.  printf("%p %d\n", year, year[0]);
3.  printf("%p %d\n", year+1, year[1]);
4.  printf("%p %d\n", year+2, year[2]);
5.  printf("%p %d\n", year+3, year[3]);
```

記憶體位址	實際記憶體位址	索引值	資料
year	x	0	107
year+1	x + 1*sizeof(int)	1	106
year+2	x + 2*sizeof(int)	2	105
year+3	x + 3*sizeof(int)	3	?

問題:其它取得陣列元素資料的方法?

輸入一維陣列

一維陣列的輸入

如何將使用者輸入在一行中的多筆資料存入陣列?

```
1.  int score[10]; //最多可存10個整數
2.  int size = 0;
3.  while(scanf("%d", &score[size])) { //scanf()接受變數的記憶體位址為參數
4.    size++;
5.    if(getchar() == '\n') { //如果讀到換行字元就離開迴圈
6.      break;
7.    }
8.  }
```

練習:輸入一維陣列

在一行中輸入不定數量(不超過10個)的整數，輸出使用者輸入的整數數量與整數數列。

輸入	輸出
3 5 7 5 6 7 8	3 3 5 7 4 5 6 7 8

傳遞一維陣列(位址)給函式

傳遞陣列給函式

例如，如何寫一個函式可以把整數陣列中所有元素加上*n*。

函式宣告

```
int arrAdd(int arr[], int size, int n);
```

傳遞陣列給函式(續)

函式宣告

```
int arrAdd(int arr[], int size, int n);
```

函式呼叫

```
1.  int score[5] = {55, 70, 30, 80, 45};
2.  arrAdd(score, 5, 10);   //每個人加10分，即結果為{65, 80, 30, 90, 55}
```

練習：陣列加法運算

輸入一個整數陣列與整數 n，實作一陣列加法運算函式將陣列中所有元素加 n。

輸出經過運算後的陣列的所有元素

輸入	輸出
55 70 30 80 -5	50 65 25 75
55 70 30 10	65 80 40

陣列應用:排序

泡沫排序(從小到大)

6	7	3	2
6	7	3	2
6	3	7	2
6	3	7	2
6	3	2	7
6	3	2	**7**

```
if(arr[1]>arr[0])
  swap(arr[1],arr[0]);
```

```
if(arr[2]>arr[1])
  swap(arr[2],arr[1]);
```

```
if(arr[3]>arr[2])
  swap(arr[3],arr[2]);
```

泡沫排序(續)

6	3	2	**7**
3	6	2	**7**
3	6	2	**7**
3	2	6	**7**
3	2	**6**	**7**

```
if(arr[1]>arr[0])
  swap(arr[1],arr[0]);
```

```
if(arr[2]>arr[1])
  swap(arr[2],arr[1]);
```

泡沫排序(續)

3	2	**6**	**7**
2	3	**6**	**7**
2	**3**	**6**	**7**

```
if(arr[1]>arr[0])
  swap(arr[1],arr[0]);
```

練習：奇偶對調

將整數陣列中索引值i與$i+1$的元素的數值對調，其中i=0,2,4,...。(如果元素大小為奇數，則最後一個元素不動。)

輸入	輸出
3 5 6 3 1 4	5 3 3 6 4 1
1 2 3 4 5	2 1 4 3 5

練習：奇偶對調

將整數陣列中索引值*i*與*i+1*的元素的數值對調，其中i=0,2,4,...。(如果元素大小為奇數，則最後一個元素不動。)

把對調的程式碼改為函式呼叫

輸入	輸出
3 5 6 3 1 4	5 3 3 6 4 1
1 2 3 4 5	2 1 4 3 5

練習:泡沫排序

給定一個整數陣列，利用泡沫排序法(bubble sort)對陣列元素進行排序。

輸入	輸出
3 6 1 4	1 3 4 6
4 6 3 1 3 4	1 3 3 4 4 6

練習:泡沫排序

給定一個整數陣列，利用泡沫排序法(bubble sort)對陣列元素進行排序。

把泡沫排序的程式碼改為函式呼叫

輸入	輸出
3 6 1 4	1 3 4 6
4 6 3 1 3 4	1 3 3 4 4 6

二維陣列

定義二維陣列

定義二維陣列的方式為

型別 陣列名稱 [列大小] [欄大小] ;

例如,

```
int score[2][5];        // 以二維陣列儲存 10個整數(佔用多少記憶體空間 ???)
char name[5][10];       // 以二維陣列儲存 50個字元(佔用多少記憶體空間 ???)
```

定義二維陣列(續)

```
int score[2][5];

score[1][3] = 100;

score[0][2] = 200;

score[1][1] = 300;
```

score[2][5]		欄索引				
		0	1	2	3	4
列索引	0			200		
	1		300		100	

定義二維陣列(初始值)

C語言儲存二維陣列時採**以列為主(row major)**，所以定義二維陣列時給予初始值的方式，如

```
int score[2][3] = {{10,20,30},{11,21,31}};
```

10	20	30
11	21	31

```
//列的大小可以由初始 值決定，但欄的大小必須給定
char name[][6] = {"Bob","Alice","John"};
```

B	o	b	\0	?	?
A	l	i	c	e	\0
J	o	h	n	\0	?

想想看

怎麼初始化二維陣列中所有元素的資料？
怎麼印出二維陣列中所有元素的資料

例如，將大小為10x10的陣列元素設定為對應的列索引值x欄索引值。例如：

```
arr[0][0] = 0,
…
arr[2][9] = 18,
…
arr[9][9] = 81
```

練習：課表

定義一個10x5的整數二維陣列代表自己的課表。列代表第1-10節課，欄代表禮拜1到禮拜5。陣列元素為0代表沒課，陣列元素為1代表有課。

使用者輸入兩個整數1 ~ 5詢問禮拜1第5節有沒有課，如果有輸出YES，如果沒有輸出NO。

輸入	輸出
3 7	NO
5 8	YES

	一	二	三	四	五
1		1			
2		1			
3		1	1	1	
4		1	1	1	
5		1			
6					
7	1				
8	1		1		1
9			1		1
10			1		

二維陣列在記憶體中的儲存方式

A	\0	?
B	\0	?
C	\0	?

記憶體內的二維陣列

```
char name[][3] = {"A","BC","D"};
```

可以下列方式取得元素 name[i][j] 的記憶體位址

```
1.  &name[i][j]
2.  *name + i*3 + j  //*name是name[0][0]的位址
3.  name[i]+j        //name[i]是name[i][0]的位址
4.  *(name+i) + j    //*(name+i)是name[i][0]的位址
```

記憶體位址	實際記憶體位址	索引值	資料
*name	x	0,0	A
*name+1	x + 1*sizeof(char)	0,1	\0
*name+2	x + 2*sizeof(char)	0,2	?
*name+3	x + 3*sizeof(char)	1,0	B
*name+4	x + 4*sizeof(char)	1,1	C
*name+5	x + 5*sizeof(char)	1,2	\0
*name+6	x + 6*sizeof(char)	2,0	D
*name+7	x + 7*sizeof(char)	2,1	\0
*name+8	x + 8*sizeof(char)	2,2	?

記憶體內的二維陣列(續)

```
int score[2][3] = {{10,20,30},{11,21,31}};
```

可以下列方式取得元素 `score[i][j]` 的記憶體位址

1. `&score[i][j]`
2. `*score + i*3 + j`
3. `score[i]+j`
4. `*(score+i) + j`

10	20	30
11	21	31

記憶體位址	實際記憶體位址	索引值	資料
*score	x	0,0	10
*score+1	x + 1*sizeof(int)	0,1	20
*score+2	x + 2*sizeof(int)	0,2	30
*score+3	x + 3*sizeof(int)	1,0	11
*score+4	x + 4*sizeof(int)	1,1	21
*score+5	x + 5*sizeof(int)	1,2	31

練習:矩陣純量加法

給定一個二維整數陣列(即矩陣)與輸入整數 n, 實作矩陣純量加法運算函式將矩陣中所有元素加 n。

輸入	輸出
-5	50 65 25 75
10	65 80 40 90

假設整數陣列為

```
{{55, 70}, {30, 80}}
```

輸入二維陣列

二維陣列的輸入

如何讓使用者輸入二維陣列?

```
 1.   int i,j;
 2.   int row, col;
 3.   int score[10][10];  /最多可存10x10個整數
 4.   scanf("%d %d", &row, &col); /使用者輸入二維陣列的大小
 5.   for(i=0;i<row;i++) {
 6.     int size = 0;
 7.     while(scanf("%d", &score[i][size])) { //scanf()接受變數的記憶體位址為參數
 8.       size++;
 9.       if(getchar() == '\n') { /如果讀到換行字元就離開迴圈
10.         break;
11.       }
12.     }
13.   }
```

練習:矩陣純量加法(2)

輸入一個二維整數陣列(不超過10x10)與輸入整數n, 實作矩陣純量加法運算函式將矩陣中所有元素加n。

輸入	輸出
2 3 10 20 30 40 50 60 -5	5 15 25 35 45 55
3 2 10 20 30 40 50 60 5	15 25 35 45 55 65

Q & A