

【第六章】

陣列與字串

講師: 李根逸 (Ken-Yi Lee), E-mail: feis.tw@gmail.com



課程大綱

- 陣列介紹 [P179]
- 陣列的使用 (1) - 取代多個變數宣告 [P181]
- 陣列的初始化 [P182]
- 陣列的使用 (2) - 搭配 **for** [P183]
- 陣列的使用 (3) - 取代 **switch** [P184]
- 在函式間傳送陣列 [P187]
- 字串介紹 [P189]
- 二維陣列介紹 [P194]

陣列

- 陣列是將一群相同資料型態的變數放在同一個名稱下的結構

- ▶ 在記憶體中，陣列會使用一段連續的空間

- 陣列宣告方式：

- ▶ 資料型態 陣列名稱[元素個數];

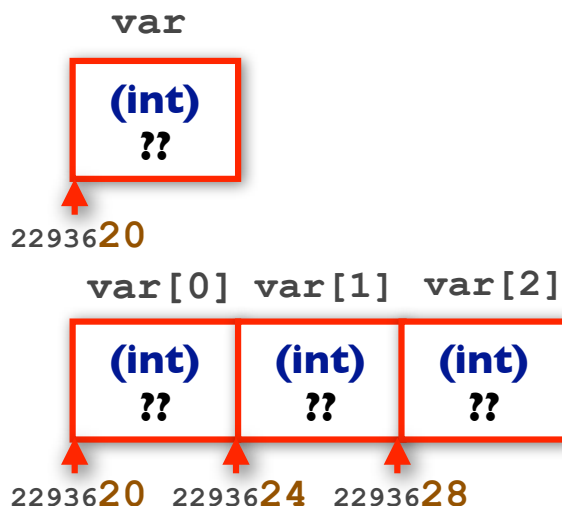
- 陣列與一般變數的差別：

- ▶ 宣告一般變數：

- `int var;`

- ▶ 宣告陣列：

- `int var[3];`



陣列的使用 (1)

求三數的最大值 (一般版)

```
int a = 3;
int b = 5;
int c = 0;
int max = a;
if (b > max) {
    max = b;
}
if (c > max) {
    max = c;
}
printf("最大值是 %d.\n", max);
```

max3_1.cpp

求三數的最大值 (陣列版 I)

```
int v[3];
v[0] = 3;
v[1] = 5;
v[2] = 0;
int max = v[0];
if (v[1] > max) {
    max = v[1];
}
if (v[2] > max) {
    max = v[2];
}
printf("最大值是 %d.\n", max);
```

max3_2.cpp

《範例》 公正骰子統計

- 有一程式 (**dice_2.cpp**) 可以統計電腦用亂數產生 **6000** 次骰子點數後，**1 ~ 6** 各點數分別出現了幾次。試著用陣列去重寫為另一程式 (**dice_3.cpp**)
 - ▶ 提示：使用 `long rand()` 函式 [`stdlib.h`] 產生亂數
 - ▶ 提示：每次骰子的點數可以用 `rand() % 6 + 1` 來得到
 - ▶ 提示：原本我們需要六個整數變數去記錄各個點數已經出現了幾次，現在可以用一個六元素的整數陣列

陣列的初始化

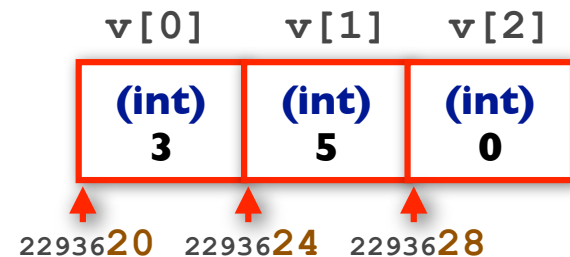
- 已知長度的陣列宣告時具有一個特殊的初始化語法：

- ▶ 資料型態 陣列名稱[元素個數] = {第一個元素的值, 第二個元素的值, 第三個元素的值, ... , 最後一個元素的值};

- 在初始化時未指定值的元素會被自動指定為 0

- ▶ 例如：

```
int v[3] = {3, 5};
```



- ▶ 在有初始化的時候，一維陣列的長度可以省略不寫，此時陣列長度就是有初始化的個數

```
int v[] = {3, 5, 0};
```

相當於 `int v[3] = {3, 5, 0};`

參考 `dice_5.cpp`

陣列的使用 (2) [for]

求三數的最大值 (陣列版2)

```
int v[3] = {3, 5, 0};

int max = v[0];
if (v[1] > max) {
    max = v[1];
}
if (v[2] > max) {
    max = v[2];
}
printf("最大值是 %d.\n", max);
```

求三數的最大值 (陣列 **x** 迴圈版)

```
int v[3] = {3, 5, 0};

int max = v[0];
for (int i = 1; i < 3; ++i) {
    if (v[i] > max) {
        max = v[i];
    }
}
printf("最大值是 %d.\n", max);
```

迴圈與陣列是絕配！使用陣列時請搭配迴圈服用

因為陣列的索引是由 **0** 開始數，所以迴圈的計數器範圍跟直覺會有所不同

max3_3.cpp

dice_6.cpp

陣列的使用 (3) [switch]

■ 使用陣列來取代 **switch**:

- ▶ 陣列可以看成是一個表格，我們利用不同的索引來讀取或寫入表格內不同格子的值

```
switch(dice) {  
    case 1: count[0]++; break;  
    case 2: count[1]++; break;  
    case 3: count[2]++; break;  
    case 4: count[3]++; break;  
    case 5: count[4]++; break;  
    case 6: count[5]++; break;  
}
```

相當於一行 `count[dice-1]++;`

參考 **dice_7.cpp**

《範例》 統計金額

- 試修改程式 **coin_1.cpp** 為 **coin_2.cpp**，利用陣列讓使用者先輸入賣出商品的總數後，分別輸入每個賣出商品的編號。輸入完後，顯示賣出的總金額

請輸入商品個數： 5

1
5
5
4
1

總共 411 元

price[5]

商品編號	商品金額
1	90
2	75
3	83
4	89
5	71

《範例》 長條圖

- 試寫一程式 (**hist.cpp**) 讓使用者輸入學生個數後，接著分別輸入每個學生的成績。輸入完後，顯示學生成績的統計圖：

▶ 級距為 (0~59, 60~69, 70~79, 80~89, 90~100)

請輸入有幾位學生： 5

第 1 位的成績是： 80

第 2 位的成績是： 85

第 3 位的成績是： 90

第 4 位的成績是： 95

第 5 位的成績是： 75

0~ 59 :

60~ 69 :

70~ 79 : *

80~ 89 : **

90~100 : **

在函式間傳送陣列

- 在呼叫函式時，將陣列內容一個一個複製給函式太耗費時間了！

- ▶ 在 C 裡面陣列名稱可以轉型為一個指向陣列開頭元素的記憶體位址



- ▶ 概念上，我們傳送給函式的參數只是陣列名稱的位址
 - 一維陣列函式的參數可以不需要宣告大小：**int max(int v[]);**
 - 如果存取到陣列實際大小以外的地方，不會發生編譯錯誤但是會產生不可預期的結果

《範例》 求最大值

- 試修改 **max_1.cpp** 為 **max_2.cpp**，寫一可以讓使用者輸入五個整數，回傳最大值的函式 **max1v**。

- ▶ **int max1v(int [5]);**

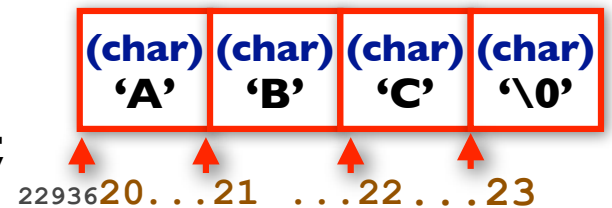
- 試修改 **max_2.cpp** 為 **max_3.cpp**，寫一可以讓使用者輸入任意個整數，回傳最大值的函式 **max2v**。

- ▶ **int max2v(int [], int N);**

字串

■ 字串是一種用 '\0' 表示結尾的字元 (char) 陣列

- ▶ 字元常數表示語法 (單引號): 'A'
- ▶ 字串初始化時表示語法 (雙引號): "ABC"
 - "ABC" 相等於 { 'A', 'B', 'C', '\0' } (\0 字元表示字串結尾)
- ▶ 宣告字串變數 (字元陣列):
 - **char** 字串名稱[字元陣列長度] = "字串內容";
 - 在有初始化的時候, 字串的長度可以省略不寫
 - 字元陣列的長度應該是字串內容的長度加一 (因為要加上一個 '\0')



■ 使用 scanf 與 printf

- ▶ scanf 跟 printf 可以用 %s 來讀取或顯示字串

```
char string[] = "Hello world";  
printf("%s", string);  
scanf("%s", string); [使用 scanf 時不用 &]
```

注意的是 **scanf** 讀入字串時, 讀到空白就會以為是字串的結尾

《範例》 字串長度

- 試寫一程式 (**strlen**) 讀取一字串後顯示該字串的長度

- ▶ 提示: `'\0'` 字元表示字串結束

請輸入一個字串 (最大長度為 100): `len`
`"len"` 的長度為 3

《補充》 C 字串函式庫

■ C 字串函式庫 **<string.h>** 中提供了許多字串處理相關的函式：

▶ 字串一 **strcpy**(字串一, 字串二)

■ 將字串二的內容複製到字串一內

▶ 字串一 **strcat**(字串一, 字串二)

■ 將字串二的內容銜接在字串一後面

▶ **int strcmp**(字串一, 字串二)

■ 此函式會比較字串一與字串二的大小，相等時回傳 0，字串一較大時回傳正值，字串一較小時回傳負值

▶ **size_t strlen**(字串)

■ 此函式會回傳字串的長度

■ 字串的結尾必定是 NULL ('\0') 字元，不包含在長度內。

《範例》 成績輸入

- 試寫一程式 (**max_grade.cpp**)，讓使用者分別輸入學生的英文姓名與成績，不停輸入直到當姓名為 **quit** 時結束，顯示最高分的學生名字與成績。

```
Please enter the name : John
Please enter the grade: 80
Please enter the name : Mary
Please enter the grade: 70
Please enter the name : Stephen
Please enter the grade: 90
Please enter the name : David
Please enter the grade: 60
Please enter the name : quit
```

```
Best: Stephen (90)
```


《補充》 排序

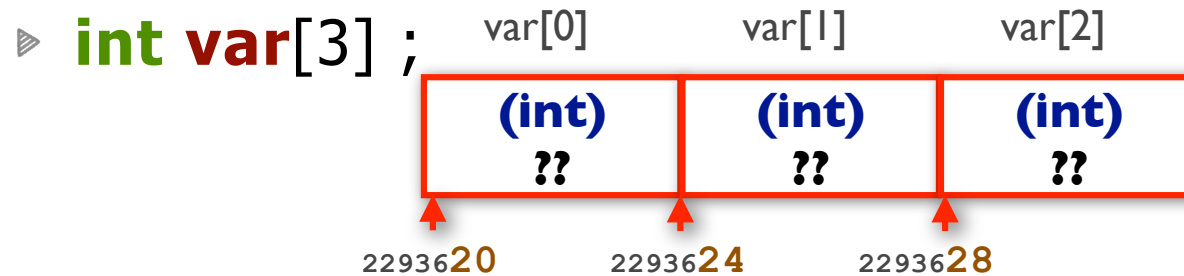
- 排序 (**sort**) 是一個常見的工作。要對一個陣列內容排序有很多種作法。我們下面示範其中一種：
 - ▶ 交換或選擇排序法：每次將最小的值換到最左邊

i=0	9	7	5	10	3	8	4
i=1	3	7	5	10	9	8	4
i=2	3	4	5	10	9	8	7
i=3	3	4	5	10	9	8	7
...							

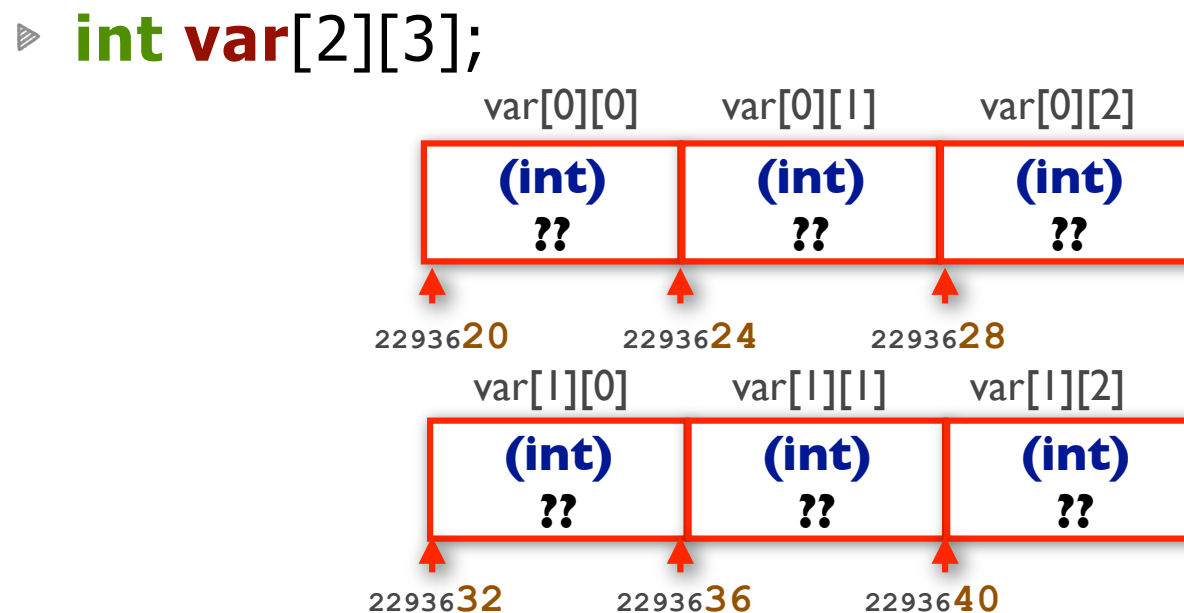
參考 `sort_1.cpp`

二維陣列

- 二維陣列就是一維陣列的陣列
- 二維陣列跟一維陣列的差別：



(int) var == 2293620



(int) var == 2293620

(int) var[0] == 2293620

(int) var[1] == 2293632

《範例》 洗牌與發牌

- 試寫一程式 (**card.cpp**)，將 **52** 張撲克牌洗牌後依序印出
 - ▶ 我們用 S, H, D, C 分別代表四個花色

D12

H1

D4

C3

...

習題 (1)

- **[E0601]** 請寫一個程式，讓使用者先依序照座號輸入十名學生的成績，接著再依序輸入一小一大的整數後印出所有成績在那兩整數之間的學生座號與成績
- **[E0602]** 請寫一個程式，讓使用者先依序照座號輸入十名學生的成績，接著再輸入一個整數，請找出所有成績最接近該數的學生座號與成績（超過一人時顯示座號最小的即可）
- **[E0603]** 試寫一程式，讓使用者先輸入三名學生的英文姓氏與成績。接著當使用者輸入任意學生的英文姓氏後，顯示該姓學生的成績。如果使用者輸入的英文姓氏不在名單中，則顯示『找不到』（**Not**

習題 (2)

- **[E0604]** 試用陣列撰寫日曆顯示程式，讓使用者輸入某個月份後，顯示西元 **2012** 該月份的日曆
 - ▶ 2012 年 1 月 1 日是星期日，2 月有 29 天

請輸入月份 (1~12) : 3

日	一	二	三	四	五	六
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

習題 (3)

- **[E0605]** 試寫一程式，讓兩個玩家輪流玩井字遊戲

```
123
A
B
C
Player #1:
Please enter your choice: 2A
```

```
123
A O
B
C
Player #2:
Please enter your choice: 2B
```

```
123
A O
B X
C
```

習題 (4)

- **[E0606]** 試寫一程式，亂數產生 5 個 0 ~ 9 之間不重複的整數
- **[E0607]** 試寫一程式，讓使用者輸入自己的英文名字後，將所有小寫英文字母轉成大寫後顯示

```
Please enter your name: Ken-Yi Lee
Hello, KEN-YI LEE !
```

- 提示：

- `'\0'` 字元表示字串結尾
- 只要比對字元的值是否在 `'a'` 與 `'z'` 之間即可知道是否是小寫字母
 - * 也可以用 `<ctype.h>` 內的 `islower()` 函式
- 要將一字元從小寫英文字母轉成大寫英文字母可以將 該字元 - `'a'` + `'A'`
 - * 也可以用 `<ctype.h>` 內的 `toupper()` 函式
