

# 程式設計 (**Programming**)

真理大學 資訊工程系 吳汶涓老師

CH07  
指標(pointer)



# 本章綱要

7-1 簡介

7-2 指標變數的定義及初始值設定

7-3 指標運算子

7-4 傳參考呼叫

7-5 `const`修飾詞在指標上的使用

7-6 使用傳參考呼叫的氣泡排序法

7-7 `sizeof`運算子

7-8 指標運算式和指標的算術運算

7-9 指標與陣列的關係

7-10 指標陣列

7-11 範例研究：洗牌與發牌

7-12 函式指標

## 7.9 指標與陣列的關係

- 指標和陣列有密切的關係
  - 陣列名稱可當成一個常數指標
  - 可用指標來進行任何的陣列下標動作
- 範例：陣列b[5], 指標 bPtr
  - 將指標指定到陣列第一個元素的位址  
→ `bPtr = &b[0];`
  - 陣列名稱其實就是第一個元素的位址  
→ `bPtr = b;`

課本pp. 7-26

- 使用指標來參考陣列元素  $b[3]$ 
  - $*(bPtr + 3)$ ，稱為指標／**位移(offset)**表示法
  - $bPtr[3]$ ，稱為指標下標表示法，亦代表  $b[3]$
  - $*(b + 3)$ ，陣列也可視為一個指標，但它會一直指向陣列開頭  
**不可改變**，所以  **$b+=3$  是錯誤的**



### 常見的程式設計錯誤 7.10

嘗試以指標算術運算來更改陣列名稱將會造成語法錯誤

## ■ 因此，參考陣列中元素的方法有四種

- 陣列下標表示法  $\rightarrow b[i]$
- 陣列名稱作指標位移法  $\rightarrow *(b + \text{offset})$
- 指標下標表示法  $\rightarrow bPtr[i]$
- 指標位移法  $\rightarrow *(bPtr + \text{offset})$

課本pp. 7-27

```

1  /* Fig. 7.20: fig07_20.cpp */
4  #include <stdio.h>
6  int main( void )
7  {
8      int b[] = { 10, 20, 30, 40 };
9      int *bPtr = b;
10     int i;
11     int offset;
12
13     printf( "Array b printed with:\nArray subscript notation\n" );
14     for ( i = 0; i < 4; i++ ) {
15         printf( "b[ %d ] = %d\n", i, b[ i ] );
16     }
17
18     printf( "\nPointer/offset notation where\n"
19             "the pointer is the array name\n" );
20     for ( offset = 0; offset < 4; offset++ ) {
21         printf( "*( b + %d ) = %d\n", offset, *( b + offset ) );
22     }
23
24     printf( "\nPointer subscript notation\n" );
25     for ( i = 0; i < 4; i++ ) {
26         printf( "bPtr[ %d ] = %d\n", i, bPtr[ i ] );
27     }
28
29     printf( "\nPointer/offset notation\n" );
30     for ( offset = 0; offset < 4; offset++ ) {
31         printf( "*( bPtr + %d ) = %d\n", offset, *( bPtr + offset ) );
32     }
33
34     return 0;
35 }

```

陣列下標表示法

Array b printed with:  
Array subscript notation  
b[ 0 ] = 10  
b[ 1 ] = 20  
b[ 2 ] = 30  
b[ 3 ] = 40

Pointer/offset notation where  
the pointer is the array name  
\*( b + 0 ) = 10  
\*( b + 1 ) = 20  
\*( b + 2 ) = 30  
\*( b + 3 ) = 40

Pointer subscript notation  
bPtr[ 0 ] = 10  
bPtr[ 1 ] = 20  
bPtr[ 2 ] = 30  
bPtr[ 3 ] = 40

Pointer/offset notation  
\*( bPtr + 0 ) = 10  
\*( bPtr + 1 ) = 20  
\*( bPtr + 2 ) = 30  
\*( bPtr + 3 ) = 40

指標下標表示法

指標位移法

圖 7.20 使用四種方法來參考陣列元素

## ■ 字串複製: 2個函式

string1 = Hello  
string3 = Good Bye

```
1  /* Fig. 7.21: fig07_21.c */
2  #include <stdio.h>
3
4  void copy1( char * const s1, const char * const s2 );
5  void copy2( char *s1, const char *s2 );
6
7  int main( void )
8  {
9      char string1[ 10 ];
10     char *string2 = "Hello";
11     char string3[ 10 ];
12     char string4[] = "Good Bye";
13
14     copy1( string1, string2 );
15     printf( "string1 = %s\n", string1 );
16
17     copy2( string3, string4 );
18     printf( "string3 = %s\n", string3 );
19     return 0;
20 }
21
```

```
24 void copy1( char * const s1, const char * const s2 )
25 {
26     int i;
29     for ( i = 0; ( s1[ i ] = s2[ i ] ) != '\0'; i++ ) {
30         ;
31     }
32 }
33
35 void copy2( char *s1, const char *s2 )
36 {
38     for ( ; ( *s1 = *s2 ) != '\0'; s1++, s2++ ) {
39         ;
40     }
41 }
```

**copy s2 to s1**  
使用指標下標表示法

**copy s2 to s1**  
使用指標位移法

# 練習

## ■ 請回答以下問題:

- ❑ 定義一個型別為**unsigned int** 的陣列 **values**, 共有**5**個元素，其初始值為 **2到10**的偶數, 假設**SIZE**常數已定義為 **5**

```
unsigned int values[SIZE] = {2, 4, 6, 8, 10};
```

- ❑ 定義一個指向**unsigned int** 的指標 **vPtr**

```
unsigned int *vPtr;
```

- ❑ 用陣列下標法印出**values**陣列內所有的元素，使用**for**迴圈
- ❑ 用兩種方式將陣列**values**的起始位址指定給指標 **vPtr**
- ❑ 用指標位移法印出**values**陣列內所有的元素
- ❑ 以陣列名稱為指標，用指標位移法印出**values**陣列內所有的元素
- ❑ 以指標下標法印出**values**陣列內所有的元素



課本pp. 7-50, EX. 7.9題



# 練習

- 試將下列程式改爲 陣列下標法、陣列位移法、指標下標法:

```
#include <stdio.h>
int mystery2(const char *s);
int main(void){
    char string[80];
    printf("Enter a string: ");
    scanf("%s", string);
    printf("%d\n", mystery2(string));
    return 0;
}

int mystery2(const char *s){
    int x;
    for( x=0; *s !='\0'; s++)
        x++;
    return x;
}
```

指標位移法



課本pp. 7-54, EX. 7.20題

## 7.10 指標陣列

- 陣列所存放的內容也可以是指標資料

- 例如：字串陣列

```
char *suit[4] = {"Hearts", "Diamonds", "Clubs", "Spades"};
```

- 字串是指向第一個字元的指標
- **char \*** 代表 **suit** 內每個元素都是一個指向 **char** 的指標
- 實際上，字串並不是儲存在 **suit** 陣列中，而是儲存指向字串的指標

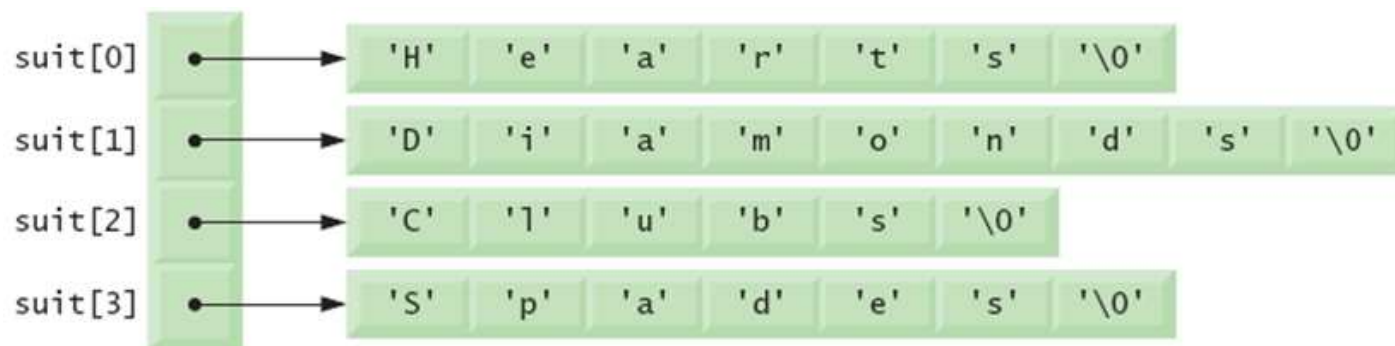


圖 7.22 suit 陣列的圖形表示

# 練習

- 寫一個查詢英文月份名稱之程式，由使用者輸入月份數值，利用指標字串陣列方式將其對應英文名稱輸出：
  - 使用者可不斷輸入數值，直到0為止
  - 若超出月份範圍，則輸出錯誤提示文字

```
input month<0 to end>:12
12 月==>December

input month:<0 to end>15
input error, please try again!!

input month:<0 to end>4
4 月==>April

input month:<0 to end>0
請按任意鍵繼續 . . .
```

※此方式可用於顧客姓名資料之查詢系統

## 7.11 範例研究：洗牌與發牌

### ■ 洗牌程式

- 使用字串陣列
- 使用雙重下標陣列 (二維陣列: 花色、點數)
- 將數字 1~ 52 放在陣列中 (代表發牌的次序)

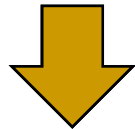
deck[2][12] 代表King of Clubs  
Clubs      King

圖 7.23 表示一副撲克牌的雙重下標陣列

		Ace	Two	Three	Four	Five	Six	Seven	Eight	Nine	Ten	Jack	Queer	King
		0	1	2	3	4	5	6	7	8	9	10	11	12
Hearts	0													
Diamonds	1													
Clubs	2													
Spades	3													

## ■ 虛擬碼 (課本pp. 7-33~34)

### □ Shuffle and deal 52 cards



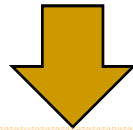
- 1) Initialize the suit array (字串陣列，表示四種花色名稱)  
`char *suit[4] = {"Hearts", "Diamonds", "Clubs", "Spades"};`
- 2) Initialize the face array (整數陣列，表示這張牌的點數)  
`const char *face[13] = {"ace", "two", "three", ..., "King"};`
- 3) Initialize the deck array (二維陣列，表示一副撲克牌)
- 4) **Shuffle the deck** (亂數打亂來洗牌)
- 5) **Deal 52 cards**  
(發牌，秀出牌的花色與點數，如King of Hearts)



## ■ Shuffle the deck

For each of the 52 cards

place card number in randomly selected unoccupied slot of deck



choose slot of deck randomly

While chosen slot of deck has been previous chosen

{

choose slot of deck randomly

place card number in chosen slot of deck

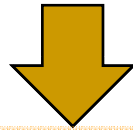
}

## ■ Deal 52 cards

For each of the 52 cards

find card number in deck array

print face and suit of card



```
for each slot of the deck array
{
    if slot contains card number
        print the face and suit of the card
}
```

```

1  /* Fig. 7.24: fig07_24.c */
2
3  #include <stdio.h>
4  #include <stdlib.h>
5  #include <time.h>
6
7
8  void shuffle( int wDeck[][ 13 ] );
9  void deal( const int wDeck[][13], const char *wFace[], const char *wSuit[] );
10
11
12 int main( void )
13 {
14     const char *suit[ 4 ] = { "Hearts", "Diamonds", "Clubs", "Spades" };
15     const char *face[ 13 ] =
16     { "Ace", "Deuce", "Three", "Four",
17       "Five", "Six", "Seven", "Eight",
18       "Nine", "Ten", "Jack", "Queen", "King" };
19
20     int deck[ 4 ][ 13 ] = { 0 };
21     srand( time( 0 ) );
22
23     shuffle( deck );
24     deal( deck, face, suit );
25     return 0;
26 }

```


**suit 和 face 皆為指標陣列**

圖 7.24 發牌程式

課本pp. 7-35



## ■ Shuffle the deck

```

34 void shuffle( int wDeck[][ 13 ] )
35 {
36     int row;
37     int column;
38     int card;
39
40     for ( card = 1; card <= 52; card++ ) {
41         do {
42             row = rand() % 4;
43             column = rand() % 13;
44         } while( wDeck[ row ][ column ] != 0 );
45         wDeck[ row ][ column ] = card;
46     }
47 }

```

**do ... while** 隨機地替每張牌  
選擇一個位置

	Ace	Two	Three	Four	Five	Six	Seven	Eight	Nine	Ten	Jack	Queen	King
	0	1	2	3	4	5	6	7	8	9	10	11	12
Hearts	0												
Diamonds	1												
Clubs	2												
Spades	3												

Nine of Hearts	Five of Clubs
Queen of Spades	Three of Spades
Queen of Hearts	Ace of Clubs
King of Hearts	Six of Spades
Jack of Diamonds	Five of Spades
Seven of Hearts	King of Clubs
Three of Clubs	Eight of Hearts
Three of Diamonds	Four of Diamonds
Queen of Diamonds	Five of Diamonds
Six of Diamonds	Five of Hearts
Ace of Spades	Six of Hearts
Nine of Diamonds	Queen of Clubs
Eight of Spades	Nine of Clubs
Deuce of Clubs	Six of Clubs
Deuce of Spades	Jack of Clubs
Four of Clubs	Eight of Clubs
Four of Spades	Seven of Spades
Seven of Diamonds	Seven of Clubs
King of Spades	Ten of Diamonds
Jack of Hearts	Ace of Hearts
Jack of Spades	Ten of Clubs
Eight of Diamonds	Deuce of Diamonds
Ace of Diamonds	Nine of Spades
Four of Hearts	Deuce of Hearts
King of Diamonds	Ten of Spades
Three of Hearts	Ten of Hearts

## ■ Deal 52 cards

```
55 void deal(const int wDeck[][13], const char *wFace[], const char *wSuit[])
56 {
57     int card;
58     int row;
59     int column;
60     for ( card = 1; card <= 52; card++ ) {
61         for ( row = 0; row <= 3; row++ ) {
62             for ( column = 0; column <= 12; column++ ) {
63                 if ( wDeck[ row ][ column ] == card ) {
64                     printf( "%5s of %-8s%c", wFace[ column ], wSuit[ row ],
65                         card % 2 == 0 ? '\n' : '\t' );
66                 }
67             }
68         }
69     }
70 }
```

課本pp. 7-36

# 練習

- 請修改課本圖7.24的洗牌發牌程式，使發牌函式能夠一次發出**5張牌**，並寫出以下的函式功能：
  - 判斷這**5張牌**是否有對子 (2張相同點數的牌)
  - 判斷這**5張牌**是否有雙對子
  - 判斷這**5張牌**是否有三條 (3張相同點數的牌)
  - 判斷這**5張牌**是否有四梅 (4張相同點數的牌)
  - 判斷這**5張牌**是否有同花 (5張相同花色的牌)
  - 判斷這**5張牌**是否有順子 (5張連續的牌)



課本pp. 7-51, EX. 7.12題

## 7.12 函式指標

- 函式指標 (pointer to a function)
  - 內含有函式在記憶體中的位址
  - 類似的例子：陣列名稱是第一個元素的位址
  - 函式名稱則是執行此函式程式碼的起始位置
- 函式指標可以
  - 傳遞給函式
  - 存放在陣列中
  - 指定給其他的函示指標

## ■ 範例：改寫氣泡排序法

### □ bubble 函式接收一個函式指標

- bubble 函式呼叫助手函式
- 用來決定是以遞增或遞減方式來排序

```
void bubble(int work[], const int size, int (*compare)(int a, int b));
```



→ 最後一個參數是函式指標，告知此函式將會傳入兩個int值並回傳int值

### □ 假如不使用小括號：int \*compare(int a, int b)

→ 則是宣告了一個函式，此函式將會傳入兩個int值並回傳一個整數的指標

課本pp. 7-39

```

1  /* Fig. 7.26: fig07_26.c */
3  #include <stdio.h>
4  #define SIZE 10
5
7  void bubble(int work[], const int size, int (*compare)(int a, int b));
8  int ascending( int a, int b );
9  int descending( int a, int b );
10
11 int main( void )
12 {
13     int order;
14     int counter;
17     int a[ SIZE ] = { 2, 6, 4, 8, 10, 12, 89, 68, 45, 37 };
18
19     printf( "Enter 1 to sort in ascending order,\n"
20            "Enter 2 to sort in descending order: " );
21     scanf( "%d", &order );
23     printf( "\nData items in original order\n" );
25
26     for ( counter = 0; counter < SIZE; counter++ ) {
27         printf( "%5d", a[ counter ] );
28     }
29
30     if ( order == 1 ) {
31         bubble( a, SIZE, ascending );
32         printf( "\nData items in ascending order\n" );
33     }
34     else {
35         bubble( a, SIZE, descending );
36         printf( "\nData items in descending order\n" );
37     }
38
39     for ( counter = 0; counter < SIZE; counter++ ) {
40         printf( "%5d", a[ counter ] );
41     }
42     printf( "\n" );
43     return 0;
44 }

```

**bubble 函式有個函式指標參數**

Enter 1 to sort in ascending order,  
Enter 2 to sort in descending order: 1

Data items in original order									
2	6	4	8	10	12	89	68	45	37
Data items in ascending order									
2	4	6	8	10	12	37	45	68	89

**根據選擇執行遞增或遞減**

Enter 1 to sort in ascending order,  
Enter 2 to sort in descending order: 2

Data items in original order									
2	6	4	8	10	12	89	68	45	37
Data items in descending order									
89	68	45	37	12	10	8	6	4	2

課本pp. 7-37



```

52 void bubble( int work[], const int size, int (*compare)( int a, int b ) )
53 {
54     int pass;
55     int count;
56     void swap( int *element1Ptr, int *element2Ptr );
57
58     for ( pass = 1; pass < size; pass++ ) {
59         for ( count = 0; count < size - 1; count++ ) {
60             if ( (*compare)( work[ count ], work[ count + 1 ] ) ) {
61                 swap( &work[ count ], &work[ count + 1 ] );
62             }
63         }
64     }
65 }
66
67
68
69
70
71
72

```

根據函式指標來判斷數值是否顛倒

```

75 void swap( int *element1Ptr, int *element2Ptr )
76 {
77     int hold;
78
79     hold = *element1Ptr;
80     *element1Ptr = *element2Ptr;
81     *element2Ptr = hold;
82 }

```

```

86 int ascending( int a, int b )
87 {
88     return b < a;
89 }

```

```

93 int descending( int a, int b )
94 {
95     return b > a;
96 }

```

Enter a number between 0 and 2, 3 to end: 0  
You entered 0 so function1 was called

Enter a number between 0 and 2, 3 to end: 1  
You entered 1 so function2 was called

Enter a number between 0 and 2, 3 to end: 2  
You entered 2 so function3 was called

## ■ 在選單驅動的系統上使用函式指標

```
3  #include <stdio.h>
6  void function1( int a );
7  void function2( int b );
8  void function3( int c );
9
10 int main( void )
11 {
14     void (*f[3])(int)={function1, function2, function3};
15
16     int choice;
18     printf( "Enter a number between 0 and 2, 3 to end: " );
19     scanf( "%d", &choice );
20
22     while ( choice >= 0 && choice < 3 ) {
26         (*f[ choice ])( choice );
28         printf( "Enter a number between 0 and 2, 3 to end: " );
29         scanf( "%d", &choice );
30     }
33     return 0;
34 }

36 void function1( int a )
37 {
38     printf( "You entered %d so function1 was called\n\n", a );
39 }
```

宣告一個函式指標陣列

根據選擇來決定呼叫哪一個函式



```
41 void function2( int b )
42 {
43     printf( "You entered %d so function2 was called\n\n", b );
44 }

46 void function3( int c )
47 {
48     printf( "You entered %d so function3 was called\n\n", c );
49 }
```

```
Enter a number between 0 and 2, 3 to end: 0
You entered 0 so function1 was called

Enter a number between 0 and 2, 3 to end: 1
You entered 1 so function2 was called

Enter a number between 0 and 2, 3 to end: 2
You entered 2 so function3 was called
```

# 練習

- 重新撰寫課本圖6.22的程式，改為選單驅動式的系統介面，程式提供四種選項給使用者：

Enter a choice:

- 0 Print the array of grades
- 1 Find the minimum grades
- 2 Find the maximum grades
- 3 Print the average on all tests for each student
- 4 End program

課本pp. 7-56, EX. 7.25題