

程式設計第八章



真理大學資工系 洪麗玲

llhung@mail.au.edu.tw

8.2 字串和字元的基本知識



- 程式裡可以含有**字元常數 (character constant)**。字元常數是一個**int**值，以單引號括起來的字元來表示。字元常數的值便是此字元在機器的**字元集 (character set)** 中的整數值。例如 '**z**' 代表了 **z** 的整數值， '**\n**' 代表了換行字元的整數值(在ASCII中分別是122和10)。
- 測試一



- 字串 (string) 是視為單一個體的一連串字元。字串可以含有字母、數字、以及數種**特殊的字元 (special character)**，如 **+**、**-**、*****、**/**、**\$** 等)。C 裡的**字串常數 (string literal 或 string constant)** 會寫在雙引號裡，如下：

| | |
|--------------------------|---------|
| "John Q. Doe" | (人名) |
| "99999 Main Street" | (街名住址) |
| "Waltham, Massachusetts" | (城市和州名) |
| "(201) 555-1212" | (電話號碼) |

- 以空字元 **'\0'** 做為結束的字元陣列

WATSE



- 字串可在宣告時指定給一個字元陣列，或一個型別為 **char *** 的變數。以下的每個定義

```
char color[] = "blue";
const char *colorPtr = "blue";
```

- 會將變數初始化為字串 **"blue"**。其中第一個宣告會建立一個**5**個元素的陣列 **color**，它的元素分別為 **'b'**、**'l'**、**'u'**、**'e'** 和 **'\0'**。第二個宣告則建立了一個指標變數 **colorPtr**，它指向存放在記憶體中某個位置的字串 **"blue"**。
- 字串=指到字串的第一個字元的**指標**

WATSE



- 我們可用**scanf**將字串儲存到陣列中。例如以下的敘述式將會指定一個字串給字元陣列**word[20]**。

```
scanf( "%19s", word );
```

- 由使用者輸入的字串將會存放在**word**當中。注意到**word**是個陣列，亦即是一個指標，所以引數**word**之前不需加 **&**。
- 測試二

WATSE

8.3 字元處理函式庫



- **字元處理函式庫 (character-handling library)** (**<ctype h>**) 包括了數個執行字元資料測試和操作的函式。每個函式都接收了一個無號字元 (表示為整數) 或**EOF**做為引數。

WATSE

| 原型 | 函式的描述 |
|-------------------------------------|---|
| <code>int isblank(int c);</code> | 如果 <code>c</code> 為一空格，用以在文件的行中分隔文字，則傳回真，否則傳回 0 (偽)。 [注意：此函數在 Microsoft Visual C++ 中不能使用。] |
| <code>int isdigit(int c);</code> | 如果 <code>c</code> 為一數字則傳回真，否則傳回 0 (偽)。 |
| <code>int isalpha(int c);</code> | 如果 <code>c</code> 為一字母則傳回真，否則傳回 0。 |
| <code>int isalnum(int c);</code> | 如果 <code>c</code> 為一數字或字母則會傳回真，否則傳回 0。 |
| <code>int isxdigit(int c);</code> | 如果 <code>c</code> 為一 16 進位的數字則會傳回真，否則傳回 0。(請參考附錄 C “數字系統”對二進制，八進制，十進制和十六進制數字的進一步描述) |
| <code>int islower(int c);</code> | 如果 <code>c</code> 為一小寫字母則傳回真，否則傳回 0。 |
| <code>int isupper(int c);</code> | 如果 <code>c</code> 為一大寫字母則傳回真，否則傳回 0。 |
| <code>int tolower(int c);</code> | 如果 <code>c</code> 是一個大寫字母， <code>tolower</code> 函式就傳回小寫的 <code>c</code> 。 如果不是， <code>tolower</code> 函式就傳回原來的引數。 |

圖8.1 字元處理函式庫 <ctype.h> 的函式(1/2)



| 原型 | 函式的描述 |
|------------------------------------|---|
| <code>int isspace(int c);</code> | 如果 <code>c</code> 為一空白字元則傳回真。空白字元包括：換行 (' <code>\n</code> ')，空白 (' '), 跳頁 (' <code>\f</code> ')，回車 (' <code>\r</code> ')，水平跳格 (' <code>\t</code> ') 及垂直跳格 (' <code>\v</code> ')。否則傳回 0。 |
| <code>int iscntrl(int c);</code> | 如果 <code>c</code> 為一控制字元則傳回真，否則傳回 0。 |
| <code>int ispunct(int c);</code> | 如果 <code>c</code> 是空格、數字以及字母以外的可列印字元，函式就會傳回真；不然傳回就是 0。 |
| <code>int isprint(int c);</code> | 如果 <code>c</code> 是包含空格 (' ') 的可列印字元，函式就會傳回真；不然就傳回零。 |
| <code>int isgraph(int c);</code> | 如果 <code>c</code> 是空格 (' ') 以外的可列印字元，函式就會傳回真；不然就傳回零。 |

圖8.1 字元處理函式庫 <ctype.h> 的函式(2/2)





❖ 8.3.1 函式 `isdigit`、`isalpha`、`isalnum` 以及 `isxdigit`

- 圖8.2的程式示範 `isdigit`、`isalpha`、`isalnum` 及 `isxdigit` 等函式的使用方式。函式 `isdigit` 會判斷它的引數是否為0到9之間的數字字元。函式 `isalpha` 會判斷它的引數是否為一個大寫字母 (A-Z) 或小寫字母 (a-z)。
- 函式 `isalnum` 會判斷它的引數是否為一個大寫字母、小寫字母或數字字元。函式 `isxdigit` 則會判斷它的引數是否為一個十六進制數的數字 (hexadecimal digit, A-F、a-f、0-9)。

WATSE

```

1 // Fig. 8.2: fig08_02.c
2 // Using functions isdigit, isalpha, isalnum, and isxdigit
3 #include <stdio.h>
4 #include <ctype.h>
5
6 int main( void )
7 {
8     printf( "%s\n%s\n\n", "According to isdigit: ",
9             isdigit( '8' ) ? "8 is a " : "8 is not a ", "digit",
10            isdigit( '#' ) ? "# is a " : "# is not a ", "digit" );
11
12     printf( "%s\n%s\n\n", "According to isalpha:",
13            isalpha( 'A' ) ? "A is a " : "A is not a ", "letter",
14            isalpha( 'b' ) ? "b is a " : "b is not a ", "letter",
15            isalpha( '&' ) ? "& is a " : "& is not a ", "letter",
16            isalpha( '4' ) ? "4 is a " : "4 is not a ", "letter" );
17
18 }
```

圖8.2 函式 `isdigit`、`isalpha`、`isalnum` 和 `isxdigit` 的用法(1/3)

WATSE

```

19     printf( "%s\n%s%s\n%s%s\n%s%s\n\n",
20             "According to isalnum:",
21             isalnum( 'A' ) ? "A is a " : "A is not a ",
22             "digit or a letter",
23             isalnum( '8' ) ? "8 is a " : "8 is not a ",
24             "digit or a letter",
25             isalnum( '#' ) ? "# is a " : "# is not a ",
26             "digit or a letter" );
27
28     printf( "%s\n%s%s\n%s%s\n%s%s\n%s%s\n%s%s\n",
29             "According to isxdigit:",
30             isxdigit( 'F' ) ? "F is a " : "F is not a ",
31             "hexadecimal digit",
32             isxdigit( 'J' ) ? "J is a " : "J is not a ",
33             "hexadecimal digit",
34             isxdigit( '7' ) ? "7 is a " : "7 is not a ",
35             "hexadecimal digit",
36             isxdigit( '$' ) ? "$ is a " : "$ is not a ",
37             "hexadecimal digit",
38             isxdigit( 'f' ) ? "f is a " : "f is not a ",
39             "hexadecimal digit" );
40 } // end main

```



圖8.2 函式isdigit、isalpha、isalnum和isxdigit的用法(2/3)

WATSE

```

According to isdigit:
8 is a digit
# is not a digit

According to isalpha:
A is a letter
b is a letter
& is not a letter
4 is not a letter

According to isalnum:
A is a digit or a letter
8 is a digit or a letter
# is not a digit or a letter

According to isxdigit:
F is a hexadecimal digit
J is not a hexadecimal digit
7 is a hexadecimal digit
$ is not a hexadecimal digit
f is a hexadecimal digit

```

圖8.2 函式isdigit、isalpha、isalnum和isxdigit的用法(3/3)

WATSE

本日練習一



- ❖ 改Fig8.2程式片段,
- ❖ 輸入:使用者自己輸入各種字元
- ❖ 輸出:顯示剛才輸入的字元型別為何
- ❖ 範例:

13

WATSE



❖ 8.3.2 函式islower、isupper、tolower以及toupper

- 圖8.3的程式示範islower、isupper、tolower和toupper函式的使用方式。函式islower會判斷它的引數是否為一個小寫字母(a-z)。函式isupper會判斷它的引數是否為一個大寫字母(A-Z)。
- 函式tolower會將大寫字母轉換成小寫字母，並傳回轉換過的小寫字母。函式toupper會將小寫字母轉換成大寫字母，並傳回轉換過的大寫字母。

WATSE

```

1 // Fig. 8.3: fig08_03.c
2 // Using functions islower, isupper, tolower, toupper
3 #include <stdio.h>
4 #include <ctype.h>
5
6 int main( void )
7 {
8     printf( "%s\n%s%s\n%s%s\n%s%s\n%s%s\n\n",
9         "According to islower:",
10         islower( 'p' ) ? "p is a " : "p is not a ",
11         "lowercase letter",
12         islower( 'P' ) ? "P is a " : "P is not a ",
13         "lowercase letter",
14         islower( '5' ) ? "5 is a " : "5 is not a ",
15         "lowercase letter",
16         islower( '!' ) ? "! is a " : "! is not a ",
17         "lowercase letter" );
18 }

```

圖8.3 函式islower、isupper、tolower和toupper的使用(1/3)



```

19     printf( "%s\n%s%s\n%s%s\n%s%s\n%s%s\n\n",
20         "According to isupper:",
21         isupper( 'D' ) ? "D is an " : "D is not an ",
22         "uppercase letter",
23         isupper( 'd' ) ? "d is an " : "d is not an ",
24         "uppercase letter",
25         isupper( '8' ) ? "8 is an " : "8 is not an ",
26         "uppercase letter",
27         isupper( '$' ) ? "$ is an " : "$ is not an ",
28         "uppercase letter" );
29
30     printf( "%s%c\n%s%c\n%s%c\n%s%c\n",
31         "u converted to uppercase is ", toupper( 'u' ),
32         "7 converted to uppercase is ", toupper( '7' ),
33         "$ converted to uppercase is ", toupper( '$' ),
34         "L converted to lowercase is ", tolower( 'L' ) );
35 } // end main

```

圖8.3 函式islower、isupper、tolower和toupper的使用(2/3)




```
According to islower:  
p is a lowercase letter  
P is not a lowercase letter  
5 is not a lowercase letter  
! is not a lowercase letter  
  
According to isupper:  
D is an uppercase letter  
d is not an uppercase letter  
8 is not an uppercase letter  
$ is not an uppercase letter  
  
u converted to uppercase is U  
7 converted to uppercase is 7  
$ converted to uppercase is $  
L converted to lowercase is l
```

圖8.3 函式islower、isupper、tolower和toupper的使用(3/3)



本日練習二



- ❖ 改Fig8.3程式片段
- ❖ 輸入:使用者自己輸入各種字元
- ❖ 輸出:顯示剛才輸入的字元是否為大寫或小寫，並做大小寫的轉換





❖ 8.3.3 函式 `isspace`、`isctrl`、`ispunct`、`isprint` 以及 `isgraph`

- 圖8.4的程式示範 `isspace`、`isctrl`、`ispunct`、`isprint` 和 `isgraph` 函式的使用方式。函式 `isspace` 會判斷某個字元是否為空白字元之一。函式 `isctrl` 會判斷它的引數是否為 **控制字元 (control characters)** 之一。函式 `ispunct` 會判斷某個字元是否為空白、數字字元或字母之外的 **可列印字元 (printing character)**。



- 函式 `isprint` 會判斷某個字元是否為一個可顯示在螢幕上的字元。函式 `isgraph` 跟 `isprint` 一樣，不過所檢查的字元不包含空白。



```

1 // Fig. 8.4: fig08_04.c
2 // Using functions isspace, iscntrl, ispunct, isprint, isgraph
3 #include <stdio.h>
4 #include <ctype.h>
5
6 int main( void )
7 {
8     printf( "%s\n%s%s\n%s%s\n%s\n\n",
9         "According to isspace:",
10         "Newline", isspace( '\n' ) ? " is a " : " is not a ",
11         "whitespace character", "Horizontal tab",
12         isspace( '\t' ) ? " is a " : " is not a ",
13         "whitespace character",
14         isspace( '%' ) ? "% is a " : "% is not a ",
15         "whitespace character" );
16
17     printf( "%s\n%s%s\n%s\n\n", "According to iscntrl:",
18         "Newline", iscntrl( '\n' ) ? " is a " : " is not a ",
19         "control character", iscntrl( '$' ) ? "$ is a " :
20         "$ is not a ", "control character" );
21

```

圖8.4 函式isspace、iscntrl、ispunct、isprint和isgraph的使用方式 (1/3)



```

22     printf( "%s\n%s%s\n%s%s\n%s\n\n",
23         "According to ispunct:",
24         ispunct( ';' ) ? "; is a " : "; is not a ",
25         "punctuation character",
26         ispunct( 'Y' ) ? "Y is a " : "Y is not a ",
27         "punctuation character",
28         ispunct( '#' ) ? "# is a " : "# is not a ",
29         "punctuation character" );
30
31     printf( "%s\n%s%s\n%s%s\n\n", "According to isprint:",
32         isprint( '$' ) ? "$ is a " : "$ is not a ",
33         "printing character",
34         "Alert", isprint( '\a' ) ? " is a " : " is not a ",
35         "printing character" );
36
37     printf( "%s\n%s%s\n%s%s\n\n", "According to isgraph:",
38         isgraph( 'Q' ) ? "Q is a " : "Q is not a ",
39         "printing character other than a space",
40         "Space", isgraph( ' ' ) ? " is a " : " is not a ",
41         "printing character other than a space" );
42 } // end main

```

圖8.4 函式isspace、iscntrl、ispunct、isprint和isgraph的使用方式 (2/3)



```

According to isspace:
Newline is a whitespace character
Horizontal tab is a whitespace character
% is not a whitespace character

According to iscntrl:
Newline is a control character
$ is not a control character

According to ispunct:
; is a punctuation character
Y is not a punctuation character
# is a punctuation character

According to isprint:
$ is a printing character
Alert is not a printing character

According to isgraph:
Q is a printing character other than a space
Space is not a printing character other than a space

```

圖8.4 函式isspace、iscntrl、ispunct、isprint和isgraph的使用方式 (3/3)



本周課堂作業



❖ 題目: 修改回文程式(文字與數字都能處理，以陣列處理)