程式設計
(Programming)

真理大學 資訊工程系 吳汶涓老師

CH08

字元與字串

本章綱要

8-1 簡介
8-2 字串和字元的基本知識
8-3 字元處理函式庫
8-4 字串轉換函式
8-5 標準輸入/輸出函式庫函式
8-6 字串處理函式庫 - 字串操作函式
8-7 字串處理函式庫 - 比較函式
8-8 字串處理函式庫 - 搜尋函式
8-9 字串處理函式庫 - 記憶體函式
8-10 字串處理函式庫 - 其他函式

8.1 簡介

- 介紹一些C標準函式庫中的函式 (function)
  - 簡單的字串和字元處理函式
  - 可以處理字元、字串、數行文字以及記憶體區塊
- 這些技術可應用在
  - 文書處理器
  - 頁面設計軟體
  - 排版程式
  - …

8.2 字串和字元的基本知識

- 字元
  - 構成原始程式的基本元件
  - 一個int值，以**單引號**括起來的字元來表示
  - **char ch = 'z';** → ch變數內儲存 z 的整數值

- 字串
  - 由一連串字元構成，常用**雙引號**括起來，如 "Hello, John"
  - 字串是字元陣列
    - 以 '\0' 字元作為字元陣列的結束
    - 字串相當於一個指標，指向第一個字元
  - 利用 scanf 輸入字串

```
char word[20];
scanf("%19s", word);
```

不需要 &

```
char color[ ] = {'b','l','u','e','\0'};
char color[ ] = "blue";
char *colorPtr = "blue";
```

常見的程式設計錯誤 8.1

字元陣列的大小不夠存放進字串的結束符號 (空字元) 時會造成錯誤

```
char color[5] = "blue"; (O)
char color[4] = "blue"; (X)
```

常見的程式設計錯誤 8.2

列印一個沒有結束空字元的「字串」會造成錯誤

```
char word[20];
scanf("%s", word);  //輸入太多會出錯
```

➔ **scanf("%19s", word);**

課本pp. 8-4

8.3 字元處理函式庫

- 字元處理函式庫
  - 包括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字元(int)或EOF作為參數
  - **#include <ctype.h>**

原型	函式的描述
`int isdigit( int c );`	如果 c 爲一數字則傳回眞，否則傳回 0（僞）。
`int isalpha( int c );`	如果 c 爲一字母則傳回眞，否則傳回 0。
`int isalnum( int c );`	如果 c 爲一數字或字母則會傳回眞，否則傳回 0。
`int isxdigit( int c );`	如果 c 爲一 16 進位的數字則會傳回眞，否則傳回 0。（請參考附錄 D “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）
`int islower( int c );`	如果 c 爲一小寫字母則傳回眞，否則傳回 0。
`int isupper( int c );`	如果 c 爲一大寫字母則傳回眞，否則傳回 0。
`int tolower( int c );`	如果 c 是一個大寫字母，tolower 函式就傳回小寫的 c。如果不是，tolower 函式就傳回原來的引數。

課本pp. 8-5

原型	函式的描述
`int toupper( int c );`	如果 c 是一個小寫字母，toupper 函式就會傳回大寫的 c。如果不是，toupper 函式就傳回原來的引數。
`int isspace( int c );`	如果 c 爲一空白字元則傳回眞。空白字元包括：換行 ('\n')，空白 (' ')，跳頁 ('\f')，回車 ('\r')，水平跳格 ('\t') 及垂直跳格 ('\v')。否則傳回 0。
`int iscntrl( int c );`	如果 c 爲一控制字元則傳回眞，否則傳回 0。
`int ispunct( int c );`	如果 c 是空格、數字以及字母以外的可列印字元，函式就會傳回眞；不然傳回就是 0。
`int isprint( int c );`	如果 c 是包含空格 (' ') 的可列印字元，函式就會傳回眞；不然就傳回零。
`int isgraph( int c );`	如果 c 是空格 (' ') 以外的可列印字元，函式就會傳回眞；不然就傳回零。

圖 8.1　字元處理函式庫 <ctype.h> 的函式

```c
/* Fig. 8.2: fig08_02.c */

#include <stdio.h>
#include <ctype.h>

int main( void )
{
   printf( "%s\n%s%s\n%s%s\n\n", "According to isdigit: ",
       isdigit( '8' ) ? "8 is a " : "8 is not a ", "digit",
       isdigit( '#' ) ? "# is a " : "# is not a ", "digit" );

   printf( "%s\n%s%s\n%s%s\n%s%s\n%s%s\n\n",
       "According to isalpha:",
       isalpha( 'A' ) ? "A is a " : "A is not a ", "letter",
       isalpha( 'b' ) ? "b is a " : "b is not a ", "letter",
       isalpha( '&' ) ? "& is a " : "& is not a ", "letter",
       isalpha( '4' ) ? "4 is a " : "4 is not a ", "letter" );

```

isdigit 測試某字元是否為數字

isalpha 測試某字元是否為字母

```
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 ",
```

isalnum 測試某字元是否為數字或字母

isxdigit 測試某字元是否為十六進位數字

```
37          "hexadecimal digit",
38          isxdigit( 'f' ) ? "f is a " : "f is not a ",
39          "hexadecimal digit" );
41      return 0;
43  }
```

```
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
```

```c
/* Fig. 8.3: fig08_03.c */
#include <stdio.h>
#include <ctype.h>
int main( void )
{
   printf( "%s\n%s%s\n%s%s\n%s%s\n%s%s\n\n",
           "According to islower:",
           islower( 'p' ) ? "p is a " : "p is not a ",
           "lowercase letter",
           islower( 'P' ) ? "P is a " : "P is not a ",
           "lowercase letter",
           islower( '5' ) ? "5 is a " : "5 is not a ",
           "lowercase letter",
           islower( '!' ) ? "! is a " : "! is not a ",
           "lowercase letter" );

   printf( "%s\n%s%s\n%s%s\n%s%s\n%s%s\n\n",
           "According to isupper:",
           isupper( 'D' ) ? "D is an " : "D is not an ",
           "uppercase letter",
           isupper( 'd' ) ? "d is an " : "d is not an ",
           "uppercase letter",
           isupper( '8' ) ? "8 is an " : "8 is not an ",
           "uppercase letter",
           isupper( '$' ) ? "$ is an " : "$ is not an ",
           "uppercase letter" );

```

islower 測試某字元是否為小寫字母

isupper 測試某字元是否為大寫字母

```
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
36    return 0; /* indicates successful termination */
37
38 } /* end main */
```

toupper 和 tolower 將字母轉換成大寫或小寫

```
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
```

練習

- 撰寫程式來判斷使用者輸入的字元有多少個字母、數字、大寫字母、小寫字母、空白。

```
#include <stdio.h>
#include <ctype.h>
int main(void)
{
      char ch;
      while((ch = getchar())!='\n')
       {
            …
       }
}
```

課本pp. 8-45, EX. 8.17

8.4 字串轉換函式

- 轉換函式
  - 需 #include <stdlib.h>標頭檔
- 將數字所組成的字串轉換成**整數**或**浮點數值**
  - 如:

```
3  #include <stdio.h>
4  #include <stdlib.h>
6  int main( void )
7  {
8     int i;
10    i = atoi( "2593" );
12    printf( "%s%d\n%s%d\n",
13            "The string \"2593\" converted to int is ", i,
14            "The converted value minus 593 is ", i - 593 );
15
16    return 0;
18 }
```

atoi將字串轉換成int

```
The string "2593" converted to int is 2593
The converted value minus 593 is 2000
```

課本pp. 8-10

函式原型	函式的描述
`double atof( const char *nPtr );`	將字串 nPtr 轉換成 double。
`int atoi( const char *nPtr );`	將字串 nPtr 轉換成 int。
`long atol( const char *nPtr );`	將字串 nPtr 轉換成 long int。
`double strtod( const char *nPtr, char **endPtr );`	將字串 nPtr 轉換成 double。
`long strtol( const char *nPtr, char **endPtr, int base );`	將字串 nPtr 轉換成 long。
`unsigned long strtoul( const char *nPtr, char **endPtr, int base );`	將字串 nPtr 轉換成 unsigned long。

圖 8.5　一般公用函式庫當中的字串轉換函式

```c
/* Fig. 8.6: fig08_06.c
#include <stdio.h>
#include <stdlib.h>
int main( void )
{
   double d;
   d = atof( "99.0" );

   printf( "%s%.3f\n%s%.3f\n",
           "The string \"99.0\" converted to double is ", d,
           "The converted value divided by 2 is ",
           d / 2.0 );

   return 0;
}
```

atof 將字串轉換成 double

```
The string "99.0" converted to double is 99.000
The converted value divided by 2 is 49.500
```

```c
/* Fig. 8.9: fig08_09.c

#include <stdio.h>
#include <stdlib.h>
int main( void )
{
   const char *string = "51.2% are admitted";

   double d;
   char *stringPtr;

   d = strtod( string, &stringPtr );

   printf( "The string \"%s\" is converted to the\n", string );
   printf( "double value %.2f and the string \"%s\"\n", d, stringPtr );

   return 0;
}
```

strtod將一段字串轉換成 double

```
The string "51.2% are admitted" is converted to the
double value 51.20 and the string "% are admitted"
```

8.5 標準輸入/輸出函式庫

- 處理字元和字串資料的函式庫，需 #include <stdio.h>

函式原型	函式的描述
`int getchar( void );`	從標準輸入讀進下一個字元，並以整數值傳回。
`char *gets( char *s );`	從標準輸入持續讀進字元到陣列 s 中，直到出現 newline 或 end-of-file 字元爲止。陣列的最後會附加上結束的 null 字元。請注意，假如 s 的大小不足會發生一個錯誤。
`int putchar( int c );`	印出存放在 c 裡的字元，並將此字元以整數傳回。
`int puts( const char *s );`	印出字串 s 並且後面跟著一個換行字元。假如成功，則傳回一個非零的整數，假如發生錯誤，則傳回 EOF。
`int sprintf( char *s, const char *format, ... );`	和 printf 相同，不過輸出是放到陣列 s 而不是印到螢幕上。
`int sscanf( char *s, const char *format, ... );`	和 scanf 相同，不過輸入是從陣列 s 讀進而不是鍵盤。

課本pp. 8-14

- 範例：讀取鍵盤輸入的字元、顯示文字在螢幕上

```
5  int main( void )
6  {
7     char c;
8     char sentence[ 80 ];
9     int i = 0;
12    puts( "Enter a line of text:" );
13
15    while ( ( c = getchar() ) != '\n') {
16       sentence[ i++ ] = c;
17    }
19    sentence[ i ] = '\0';
22    puts( "\nThe line entered was:" );
23    puts( sentence );
25    return 0;
27 }
```

puts 將一行文字顯示在螢幕上

getchar 讀入使用者輸入的單一字元

```
Enter a line of text:
This is a test.

The line entered was:
This is a test.
```

- 範例：讀入一串文字並反向遞迴輸出每個字元

```
3  #include <stdio.h>
5  void reverse( const char * const sPtr );
7  int main( void )
8  {
9     char sentence[ 80 ];
11    printf( "Enter a line of text:\n" );
14    gets( sentence );
15
16    printf( "\nThe line printed backward is:\n" );
17    reverse( sentence );
19    return 0;
21 }
```

gets 讀入使用者輸入的一行文字

```
Enter a line of text:
Characters and Strings

The line printed backward is:
sgnirtS dna sretcarahC
```

```
24 void reverse( const char * const sPtr )
25 {
27    if ( sPtr[ 0 ] == '\0' ) {
28       return;
29    }
30    else {
31       reverse( &sPtr[ 1 ] );
33       putchar( sPtr[ 0 ] );
34    }
36 }
```

putchar 將單一字元顯示在銀幕上

- 範例：將文字快速存入陣列之中

```
5  int main( void )
6  {
7     char s[ 80 ];
8     int x;
9     double y;
11    printf( "Enter an integer and a double:\n" );
12    scanf( "%d%lf", &x, &y );
14    sprintf( s, "integer:%6d\ndouble:%8.2f", x, y );
15
16    printf( "%s\n%s\n",
17           "The formatted output stored in array s is:", s );
19    return 0;
21 }
```

sprintf將一行文字放到陣列中

```
Enter an integer and a double:
298 87.375
integer:   298
double:   87.38
The formatted output stored in array s is:
```

練習

- 撰寫程式輸入4個代表浮點數的字串，將這些字串轉換成整數再計算總和、平均並印出數值。

```
Enter four floating point string: 12.34 23.45 21.23 45.63

The total of the values is 102.65
請按任意鍵繼續 . . .
```

- 撰寫程式來讀取鍵盤輸入的字串並達到下列功能:
  - 列出a~z字母各別出現的次數
  - 列出每個單字出現的次數

```
Enter three lines of text:
This program counts the occurrences of each
letter of the alphabet in the input text. Then,
it prints a summary of the occurrences.
```

```
Total letter counts:
a:  6    b:  1
c:  8    d:  0
e: 14    f:  3
g:  1    h:  8
i:  5    j:  0
k:  0    l:  2
m:  3    n:  7
o:  7    p:  4
q:  0    r:  9
s:  6    t: 15
u:  5    v:  0
w:  0    x:  1
y:  1    z:  0
```

```
"This" appeared 1 time
"program" appeared 1 time
"counts" appeared 1 time
"the" appeared 4 times
"occurrences" appeared 2 times
"of" appeared 3 times
"each" appeared 1 time
"letter" appeared 1 time
"alphabet" appeared 1 time
"in" appeared 1 time
"input" appeared 1 time
"text" appeared 1 time
"Then," appeared 1 time
"it" appeared 1 time
"prints" appeared 1 time
"a" appeared 1 time
"summary" appeared 1 time
請按任意鍵繼續 . . .
```