

程式設計 (**Programming**)

真理大學 資訊工程系 吳汶涓老師

CH09

C格式化輸入/輸出



本章綱要

9-1 簡介

9-2 資料流

9-3 使用printf的格式化輸出

9-4 顯示整數

9-5 顯示浮點數

9-6 顯示字串和字元

9-7 其他轉換指定詞

9-8 使用欄位寬度和精準度的顯示方式

9-9 在printf格式控制字串中使用旗標

9-10 字元常數和跳脫序列的顯示

9-11 使用scanf的格式化輸入

9.1 簡介

■ 標準輸入/輸出資料流

- 控制資料輸入、結果輸出
- printf()、scanf()
- gets()、puts()、getchar()、putchar() → 需 **#include <stdio.h>**

9.2 資料流

- 所有輸入輸出都是以資料流(**stream**)來完成
 - 資料流就是位元組(**bytes**)所形成的序列
 - 標準輸入: 鍵盤 (從裝置流向主記憶體)
 - 標準輸出: 螢幕 (從主記憶體流向裝置)
 - 標準錯誤: 螢幕

9.3 使用printf的格式化輸出

- `printf(format-control-string, other-arguments)`
 - 格式控制字串：描述輸出的格式 (“ ”)
 - 其他參數：對應到格式控制字串中的轉換指定 (%d, %c, %f, ...)
 - 可精確地控制輸出 ← 使用轉換指定 %
 - 可執行四捨五入、欄位對齊、靠右/左對齊、指數或浮點格式、精確度、十六進位格式等等

```
printf("姓名: %s\n", name);  
printf("%s %s\n", "姓名:", name);
```

9.4 顯示整數

■ 整數

- 不含小數的數字
- 正數、負數或零
- %d: 有號十進位整數
- %i: 有號十進位整數 (在scanf時, %d 和 %i 不同)
- %o: 無號八進位整數
- %u: 無號十進位整數
- %x 或是 %X: 無號十六進位整數
- %h 或是 %l: short 或long 整數 (放在轉換指定詞前面)

```

1  /* Fig 9.2: fig09_02.c */
3  #include <stdio.h>
4
5  int main( void )
6  {
7      printf( "%d\n", 455 );
8      printf( "%i\n", 455 );
9      printf( "%d\n", +455 );
10     printf( "%d\n", -455 );
11     printf( "%hd\n", 32000 );
12     printf( "%ld\n", 2000000000L );
13     printf( "%o\n", 455 );
14     printf( "%u\n", 455 );
15     printf( "%u\n", -455 );
16     printf( "%x\n", 455 );
17     printf( "%X\n", 455 );
19     return 0;
21 }

```

d 和 i 指定有號整數

h 指定 short 整數

o 指定八進位整數

u 指定無號整數

x 和 X 指定十六進位整數

```

455
455
455
-455
32000
2000000000
707
455
4294966841
1c7
1c7

```

9.5 顯示浮點數

■ 浮點數

- 具有小數點的數字
- 指數記號 (電腦所使用的科學記號表示法)
 - $150.3 \rightarrow$ 科學記號為 $1.503 * 10^2$
 - $150.3 \rightarrow$ 指數記號為 $1.503E+02$
- %e 或是 %E: (指數記號表示，預設小數後有6位數)
- %f: 使用固定點表示法
- %g 或是 %G: 以浮點f 或指數形式(e, E)表示數值

※注意: 指定精確度會產生捨去誤差。(其中%f不會四捨五入)


```
1 /* Fig 9.4: fig09_04.c */
```

```
5 #include <stdio.h>
```

```
7 int main( void )
```

```
8 {
```

```
9     printf( "%e\n", 1234567.89 );
```

```
10    printf( "%e\n", +1234567.89 );
```

```
11    printf( "%e\n", -1234567.89 );
```

```
12    printf( "%E\n", 1234567.89 );
```

```
13    printf( "%f\n", 1234567.89 );
```

```
14    printf( "%g\n", 1234567.89 );
```

```
15    printf( "%G\n", 1234567.89 );
```

```
16
```

```
17     return 0;
```

```
18
```

```
19 }
```

e 和 **E** 指定指數記號

f 指定固定點表示法

g 和 **G** 依數值大小決定要用指數記號或固定點表示法

1.234568e+006

1.234568e+006

-1.234568e+006

1.234568E+006

1234567.890000

1.23457e+006

1.23457E+006

9.6 顯示字串和字元

■ %c

- 顯示 char 的變數

■ %s

- 需使用一個指向char的指標或為一個字元陣列
- 顯示出字串中每個字元，直到空字元('\0')為止

■ 注意：

- 字元常數應用單引號(' ')括起來
- 雙引號(" ")用在字串上

```

1  /* Fig 9.5: fig09_05c */
3  #include <stdio.h>
5  int main( void )
6  {
7      char character = 'A';
8      char string[] = "This is a string";
9      const char *stringPtr = "This is also a string";
10
11     printf( "%c\n", character ); ← c 指定要顯示的字元
12     printf( "%s\n", "This is a string" );
13     printf( "%s\n", string ); ← s 指定要列印的字串
14     printf( "%s\n", stringPtr );
16     return 0;
18 }

```

```


A
This is a string
This is a string
This is also a string

```

9.7 其他轉換指定詞

- **%n**: 儲存目前printf輸出的字元個數

`printf("The num of characters %n", &y);`



- **%p**: 顯示某變數之記憶體位址 (通常使用十六進位表示)
- **%%**: 顯示出百分比符號

```
7   int *ptr;  
8   int x = 12345;  
9   int y;  
11  ptr = &x;  
12  printf( "The value of ptr is %p\n", ptr );  
13  printf( "The address of x is %p\n\n", &x );  
15  printf( "Total characters printed on this line:%n", &y );  
16  printf( " %d\n\n", y );  
17  
18  y = printf( "This line has 28 characters\n\n" );  
19  printf( "%d characters were printed\n\n", y );  
21  printf( "Printing a %% in a format control string\n" );
```

```
The value of ptr is 0012FF78  
The address of x is 0012FF78  
  
Total characters printed on this line: 38  
  
This line has 28 characters  
28 characters were printed  
  
Printing a % in a format control string
```

9.8 使用欄位寬度和精準度的顯示

- %數字: 欄位寬度控制 (數值會靠右對齊，負號占一個字元)
`printf("%6d", 1200);`
`printf("%6d", -1200);`
- %.數字: 精確度控制
`printf("%.6d", 1200);`
`printf("%.3f", 123.9453);`
`printf("%.5s", "Happy Birthday");`
- 欄位寬度與精確度混合使用
`printf("%9.3f", 123.45678);`

```

9  printf( "%4d\n", 123 );
10 printf( "%4d\n", 1234 );

13 printf( "%4d\n", -1 );
14 printf( "%4d\n", -12 );
15 printf( "%4d\n", -123 );

```

欄寬為4，C會將數字印在4個字元的位置中

C將負號視為一個字元的寬度

```

123
1234

```

```

-1
-12
-123

```

```

8  int i = 873;
9  double f = 123.94536;

11
12 printf( "Using precision for integers\n" );
13 printf( "\t%.4d\n\t%.9d\n\n", i, i );
14
15 printf( "Using precision for floating-point numbers\n" );
16 printf( "\t%.3f\n\t%.3e\n\t%.3g\n\n", f, f, f );

```

Using precision for integers

```

0873
000000873

```

Using precision for floating-point numbers

```

123.945
1.239e+002
124

```

9.9 在printf格式控制字串中使用旗標

■ 旗標

- 輔助的格式化功能
- 直接在百分記號右邊放置旗標
- 可同時使用數個旗標

- (負號)	在指定的欄位中將輸出靠左對齊。
+(正號)	在正值之前顯示正號，在負值之前顯示負號。
<i>space</i>	在正數之前顯示一空格，但不印+號。
#	和八進位轉換指定詞 o 一起使用的時候，輸出之前會加上 0。 和十六進位轉換指定詞 x 或 X 一起使用的時候，在輸出之前會加上 0x 或 0X。 沒有小數部份但是會以 e, E, f, g 或 G 顯示的浮點數來顯示小數點。 (一般只在含有小數部分時才會顯示小數點。)對於 g 和 G 指定詞而言，小數點後面多餘的零不會消除。
0 (零)	使用前導的零填補欄位。

```

1  /* Fig 9.11: fig09_11.c */
3  #include <stdio.h>
4
5  int main( void )
6  {
7      printf( "%10s%10d%10c%10f\n\n", "hello", 7, 'a', 1.23 );
8      printf( "%-10s%-10d%-10c%-10f\n", "hello", 7, 'a', 1.23 );
9
10     return 0;
11
12 }

```

← 旗標會使字元在欄位中向左對齊

hello	7	a	1.230000
hello	7	a	1.230000

9.10 跳脫符號

■ 跳脫符號用來顯示字元常數

跳脫序列	說明
\' (單引號)	輸出單引號 (') 字元。
\" (雙引號)	輸出雙引號 (") 字元。
\? (問號)	輸出問號 (?) 字元。
\\ (反斜線)	輸出反斜線 (\) 字元。
\a (警告音或鈴聲)	輸出可聽見的聲音 (鈴聲) 或聽的到的警告聲。
\b (退格)	將游標在目前所在的行當中往後移一位。
\f (跳頁)	將游標移到下一個邏輯頁的起始位置。
\n (新行)	將游標移到下一行的起始位置。

練習

- 撰寫程式將整數的華氏溫度(0~212)轉成3位精確度的浮點攝氏溫度，使用公式如下。輸出必須印成兩行，都要靠右對齊，每一欄位均為10個字元，溫度需標明正負號。

$$\text{攝氏 } C = (\text{華氏 } F - 32) * 5/9$$



Fahrenheit	Celsius
0	-17.778
1	-17.222
2	-16.667
3	-16.111
4	-15.556
5	-15.000
6	-14.444
7	-13.889
.	.
.	.
.	.
204	+95.556
205	+96.111
206	+96.667
207	+97.222
208	+97.778
209	+98.333
210	+98.889
211	+99.444
212	+100.000