

序言

這本書是我規劃 C 語言學習地圖的第三本。看完第一本的「樂在 C 語言」後，希望您能在「戰勝 C 語言」，這也是本書取名的考量因素。同時在內容的深度與廣度上也加以擴大，俾能對「C」有更進一步的了解。第三本是「精通 C/C++ 指標」期使您對 C 語言的核心主題——指標，能更上一層樓。

有人常問我學好程式設計的秘方是什麼，其實沒有秘方，只要在碰到有錯誤時（bugs）時，務必親自除錯（debug）。常發現同學在上機實作時，遇到 bugs 常常會請別的同学幫他 debug，久而久之，幫人 debug 的同学的功力向上提昇，而另一方卻向下沈淪。因為親自除錯，不僅可以知道撰寫程式時常易犯錯的地方外，也能讓人充實實力，下次再戰時一定可以戰勝它，那心中一定相當的喜悅，有時還會因 debug 成功而睡不著覺。

基於上述除錯的重要性，因此在每章的每一小節後，皆有「除錯題」。在程式包含有錯誤的方式加以呈現，讓您親自來 debug，以了解此程式常犯錯的地方，下次再撰寫這類程式時就難不倒您了。而「練習題」部份，測試您對這小節一的了解程度。若沒有百分百的了解，請將此章再看一遍，同時幾乎每一章皆有應用範例，得以讓您了解本章主題內容的應用之處。

本書的內容共有十三章，希望您能依照順序閱讀。各章除了上述提到的除錯題與練習題外，在章末皆附有「問題演練」與「程式實作」，以測試您對這一章的了解程度。筆者雖然在這領域的研究將近 30 年，但難免仍有疏漏不足之處，因此期盼您的批評與指正。



mjtsai168@gmail.com

標準的輸出 與輸入

2

CHAPTER

在程式的撰寫中，最基本也是最常用的就是標準的輸入與輸出（Standard input/output）功能。標準的輸出/輸入表示，結果從螢幕輸出/從鍵盤輸入資料，其庫函數分別是 `printf()` 與 `scanf()`。就讓我們先從 `printf()` 開始介紹起！

2-1 printf() 函數

`printf()` 函數的語法如下：

```
printf("欲輸出於螢幕的內容", 變數或常數);
```

簡單的使用方式，請參閱以下的範例及其剖析。



範例 2-1a

```
01 /* ex2-1a.c */
02 #include <stdio.h>
03 #include <stdlib.h>
04 int main()
05 {
06     printf("標準的輸出與輸入函數計有:\n");
07     printf("printf() 和 scanf()\n");
08     system("PAUSE");
09     return 0;
10 }
```

輸出結果

標準的輸出與輸入函數計有：
`printf()` 和 `scanf()`

【程式剖析】

第 6-7 行的 `printf()` 函數將雙引號內的資料加以輸出。在雙引號內的 `\n` 是控制字元，具有跳行的功能，若沒有 `\n`，則輸出結果如下所示：

標準的輸出與輸入函數計有：`printf()` 和 `scanf()`

有關控制字元往後會再詳述。



範例 2-1b

```
01 /* ex2-1b.c */
02 #include <stdio.h>
03 #include <stdlib.h>
04 int main()
05 {
06     int number = 8;
07     char letter = 'z';
08     /* %d、%c 為格式特定字 */
09     printf("1 加 1 是 %d\n", 2);
10     printf("1 加 7 是 %d\n", number);
11     printf("%c 是第一個英文字母\n", 'a');
12     printf("%c 是最後一個英文字母\n", letter);
13     system("PAUSE");
14     return 0;
15 }
```

輸出結果

1 加 1 是 2
1 加 7 是 8
a 是第一個英文字母
z 是最後一個英文字母

【程式剖析】

在雙引號內的格式特定字，是由 % 加上一個英文字母所組成的，如程式中第 9-12 行的 %d、%c 就是。逗號後之變數或常數，乃是一對一的與格式特定字相對應之。%d 對應的是整數的變數或常數，而 %c 對應的是字元的變數或常數。

2-1-1 printf() 函數的格式特定字

除了 %d 與 %c 之外，還有一些 printf() 函數常用的格式特定字（format specific character），如表 2-1 所示。

表 2-1 printf() 函數常用的格式特定字

| 格式特定字 | 功能 | 對應的變數或常數 |
|-------|----------------|--------------|
| %c | 印出單一字元 | char |
| %d | 印出十進位整數（有正負號） | int |
| %x | 印出十六進位整數（無正負號） | int |
| %o | 印出八進位整數（無正負號） | int |
| %f | 印出浮點數（使用小數表示） | float、double |
| %% | 印出百分比的符號（%） | |

我們以下列幾個範例加以解說之。



範例 2-1c

```
01 /* ex2-1c.c */
02 #include <stdio.h>
03 #include <conio.h>
04 int main()
05 {
06     char letter1 = 'a';
07     char letter2 = 'b';
08     char letter3 = 'c';
09     char letter4 = 't';
```

```
10     printf("%c %c %c are known by everyone\n", letter1, letter2,  
11         letter3);  
12     printf("I have a %c%c%c\n", letter3, letter1, letter4);  
13     getch();  
14     return 0;  
15 }
```

輸出結果

```
a b c are known by everyone  
I have a cat
```

【程式剖析】

此範例使用多個格式特定字，以印出其對應的變數值或常數。注意，每一格式特定字與後面的變數或常數，是一對一的對應，若有三個變數，則會有三個對應的格式特定字元。



範例 2-1d

```
01  /* ex2-1d.c */  
02  #include <stdio.h>  
03  #include <stdlib.h>  
04  int main()  
05  {  
06     int i_num1= 100;  
07     printf("100%% orange juice\n");  
08  
09     /* 使用有關整數的格式特定字輸出 */  
10     printf("輸出 100 使用 %d: %d\n", i_num1);  
11     printf("輸出 100 使用 %%x: %x\n", i_num1);  
12     printf("輸出 100 使用 %%o: %o\n", i_num1);  
13     system("PAUSE");  
14     return 0;  
15 }
```

輸出結果

```
100% orange juice  
輸出 100 使用 %d: 100  
輸出 100 使用 %%x: 64  
輸出 100 使用 %%o: 144
```

【程式剖析】

在程式中輸出時，其中第 10-12 行使用 %%，即代表輸出 %，而 %d、%x、%o 格式特定字對應的是整數變數。

範例 2-1e

```
01 /* ex2-1e.c */
02 #include <stdio.h>
03 #include <stdlib.h>
04 int main()
05 {
06     float f_number = 123.456;
07     double d_number = 123.456;
08     printf("f_number = %f\n", f_number);
09     printf("d_number = %f\n", d_number);
10     system("PAUSE");
11     return 0;
12 }
```

輸出結果

```
f_number = 123.456001
d_number = 123.456000
```

【程式剖析】

此範例說明，浮點數變數對應的格式特定字元是 %f（第 8-9 行）。float 資料型態的數值，輸出時可能會有誤差，但 double 資料型態的數值不會有誤差。一般系統將浮點數常數預設為 double。

2-1-2 控制字元

控制字元是以反斜線為首，如 '\n'，表示跳行。表 2-2 列出 printf() 函數常用的控制字元（control character）。

表 2-2 printf() 函數常用的控制字元

| 控制字元 | 功能 | 對應鍵盤按鍵 |
|------|----------------------|---------------|
| \n | 跳行 | Enter, return |
| \t | 跳格 | tab |
| \b | 倒退一格 | ← |
| \r | 回到此列最前端 | Home |
| \f | 跳頁 | |
| \' | 印出單引號 (') | |
| \" | 印出雙引號 (") | |
| \\ | 印出反斜線 (\) | |
| \xhh | 印出 ASCII 十六進位碼所對應的字元 | |
| \0oo | 印出 ASCII 八進位碼所對應的字元 | |

有關控制字元的用法，我們以範例 2-1f 來說明。

範例 2-1f

```

01 /* ex2-1f.c */
02 #include <stdio.h>
03 #include <stdlib.h>
04 int main()
05 {
06     printf("We don\t have enough time\n");
07     printf("Everybody say \" Later is better than never\"\n");
08     printf("Computer is powerfuk\b1\n");
09     printf("    can make it\rYou\n");
10     printf("Failure \151\163 the mother \x6f\x66 success\n");
11     system("PAUSE");
12     return 0;
13 }

```

輸出結果

```

We don't have enough time
Everybody say "Later is better than never"
Computer is powerful
You can make it
Failure is the mother of success

```

【程式剖析】

在上例中第 10 行，八進位 ASCII 的 151、163 所對應的字元是 i 和 s，所以 \151 會印出 i，而 \163 則印出 s。十六進位 ASCII 的 6f、66 所對應的字元是 o 和 f，所以 \x6f 會印出 o，\x66 則印出 f。有關 ASCII，請參閱附錄 A。

2-1-3 輸出欄位寬

回頭看看範例 2-1e，使用 %f 輸出的浮點數，小數點後面是否太多位數了？若想少一點的位數是否可以呢？此問題可以使用欄位寬 (width) 加以控制。請參閱表 2-3：

表 2-3 欄位寬的功能

| 格式 | 功能 | 適用格式 |
|--------|----------------------------|------|
| %nc | 設定欄位寬為 n。 | %c |
| %-nc | 設定欄位寬為 n，而且靠左對齊。 | %c |
| %nd | 設定欄位寬為 n。 | %d |
| %-nd | 設定欄位寬為 n，而且靠左對齊。 | %d |
| %n.mf | 設定欄位寬為 n，小數點後欄位寬為 m。 | %f |
| %-n.mf | 設定欄位寬為 n，小數點後欄位寬為 m，並向左靠齊。 | %f |
| %.mf | 設定小數點後欄位寬為 m。 | %f |

欄位寬的用法，請參閱以下的範例及其剖析。

範例 2-1g

```

01  * ex2-1g.c */
02  #include <stdio.h>
03  #include <stdlib.h>
04  int main()
05  {
06      char letter = 'a';
07
08      printf("|%c| using %c\n", letter);
09      printf("|%10c| using %%10c\n", letter);
10      printf("|%-10c| using %%-10c\n", letter);

```



```

11     system("PAUSE");
12     return 0;
13 }

```

輸出結果

```

|a| using %c
|      a| using %10c
|a      | using %-10c

```

【程式剖析】

從輸出結果可明顯的看出，使用 %10c 與 %-10c 的不同。第 9 行使用 %10c，輸出的資料會向右對齊，而第 10 行 %-10c 則是向左對齊。



範例 2-1h

```

01 /* ex2-1h.c */
02 #include <stdio.h>
03 #include <stdlib.h>
04 int main()
05 {
06     int number1 = 123;
07     double number2 = 123.456;
08     printf("使用 %d-----|%d|\n", number1);
09     printf("使用 %10d-----|%10d|\n", number1);
10     printf("使用 %-10d-----|%-10d|\n", number1);
11     printf("使用 %2d-----|%2d|\n", number1);
12     printf("使用 %10.2f----|%10.2f|\n", number2);
13     printf("使用 %%.2f-----|%.2f| \n", number2);
14     system("PAUSE");
15     return 0;
16 }

```

輸出結果

```

使用 %d-----|123|
使用 %10d-----|          123|
使用 %-10d-----|123      |
使用 %2d-----|123|
使用 %10.2f----|    123.46|
使用 %%.2f-----|123.46|

```

【程式剖析】

第 12 行使用 %10.2f，表示欄位寬為 10（包含整數、小數與小數點），小數後輸出二位，結果為 | 123.46|（四捨五入）；在範例中第 11 行的 %2d 會變為 %d，因為 number1 的位數大於 2，故 2 會被忽略掉。第 10 行 %-10d 則將輸出結果向左靠齊。

我們為變數設定欄位寬，可使輸出結果整齊美觀。請參閱範例 2-1i。

範例 2-1i

```
01 /* ex2-1i.c */
02 #include <stdio.h>
03 #include <stdlib.h>
04 int main()
05 {
06
07     double num1=123.456, num2=12.34, num3=12.345 ;
08     double num4=4.456, num5=45.67, num6=456.789 ;
09     printf("%f %f %f\n", num1, num2, num3);
10     printf("%f %f %f\n", num4, num5, num6);
11
12     printf("\n\n 使用欄位寬...\n");
13     printf("%8.3f %8.3f %8.3f\n", num1, num2, num3);
14     printf("%8.3f %8.3f %8.3f\n", num4, num5, num6);
15
16     system("PAUSE");
17     return 0;
18 }
```

輸出結果

```
123.456000 12.340000 12.345000
4.456000 45.670000 456.789000
```

使用欄位寬...

```
123.456 12.340 12.345
4.456 45.670 456.789
```

【程式剖析】

從輸出結果得知，設定欄位寬可使輸出結果更加美觀。

printf() 函數暫時介紹到這，接下來介紹 scanf() 輸入函數。



除錯題

以下是小王學完這一節後所寫的程式，請您訂正之。

- ```
1. /* bugs2-1-1.c */
#include <stdio.h>
#include <stdlib.h>
int main()
{
 char letter = 'a';

 printf("%c\n", letter);
 printf("|10%c\n", letter);
 printf("|-10c\n", letter);
 system("PAUSE");
 return 0;
}
```
- ```
2. /* bugs2-1-2.c */
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int number1 = 123;
    double number2 = 123.456;

    printf("10|d\n", number1);
    printf("-|10d\n", number1);

    printf("|10|f\n", number2);
    printf("-|10.2f\n", number2);
    system("PAUSE");
    return 0;
}
```
- ```
3. /* bugs2-1-3.c */
#include <stdio.h>
#include <stdlib.h>
int main()
```

```
{
 int i_number = 100;
 double d_number = 123.456;
 printf("i_number = %f\n", i_number);
 printf("d_number = %d\n", d_number);
 system("PAUSE");
 return 0;
}
```



1. 試撰寫一程式，以 printf 函數配合適當的格式特定字，輸出下列變數的值。
  - (1) var1 (資料型態: char, 值: 'a')
  - (2) var2 (資料型態: short int, 值: 123)
  - (3) var3 (資料型態: double, 值: 123.45)
  - (4) var4 (資料型態: int, 值: 1234567)
  - (5) var5 (資料型態: unsigned short int, 值: 60000)
2. 試以 printf 函數，輸出 1、10、100、1000，以及 10000，其格式為 |%-4d|。
3. 試以 printf 函數，輸出 9.9、99.99、999.999，以及 9999.9999，其格式為 |%10.4f| 和 |%-10.4f|。
4. 試問 printf("abcde\tfghij\rklmno\tpqrst\buvwxyz\n") 的輸出結果為何？
5. 試利用 printf 函數與控制字元，將 "1/5 = 20%" 輸出於螢幕上。

## 2-2 scanf() 函數

scanf() 函數是從鍵盤輸入資料，其語法與 printf() 相似，我們以範例 2-2a 說明之。

### 範例 2-2a

```
01 /* ex2-2a.c */
02 #include <stdio.h>
03 #include <stdlib.h>
04 int main()
05 {
06 int x;
07 printf("請輸入一整數值: ");
08 scanf("%d", &x); /* &為位址運算子 */
09 printf("您輸入的整數是 %d\n", x);
10 system("PAUSE");
11 return 0;
12 }
```

### 輸出結果

```
請輸入一整數值: 300
您輸入的整數是 300
```


### 【程式剖析】

您是否發現 scanf() 與 printf() 函數語法的不同之處。在第 8 行 scanf() 函數中，變數前面多了一個 & 的符號，以取得此變數在記憶體中的位址。& 為位址運算子 (address operator)。

scanf() 的完整語法為：

```
scanf("格式特定字", 變數的位址);
```

由於 scanf() 將輸入的資料，根據變數的位址加以存放之，因此必須藉由 & 來取得此變數在記憶體的位址。這好比郵差必須依據收件人的地址投遞信件，否則，信件將無法送達。

 範例 2-2b

```
01 /* ex2-2b.c */
02 #include <stdio.h>
03 #include <stdlib.h>
04 int main()
05 {
06 int x;
07 printf("請輸入一整數值: ");
08 scanf("%d", x);
09 printf("您輸入的整數是 %d\n", x);
10 system("PAUSE");
11 return 0;
12 }
```

## 輸出結果

```
請輸入一整數值: 100
(此會產生錯誤訊息)
```

## 【程式剖析】

在上例第 7 行 `scanf` 函數中，`a` 變數前沒有加上 `&`，雖然使用者輸入的值為 100，但會產生錯誤訊息。此乃因為編譯程式不知道要將輸入值置於何處。

## 2-2-1 `scanf()` 函數的格式特定字

`scanf()` 除了上述與 `printf()` 不同外，在格式特定字也有一些相異的地方，請參閱表 2-4：

表 2-4 `scanf()` 的格式特定字及其功能

| <code>scanf()</code> 的格式特定字 | 功 能           | 對應的變數型態             |
|-----------------------------|---------------|---------------------|
| <code>%c</code>             | 輸入字元          | <code>char</code>   |
| <code>%d</code>             | 輸入十進位整數（有正負號） | <code>int</code>    |
| <code>%f</code>             | 輸入單準確度的浮點數    | <code>float</code>  |
| <code>%lf</code>            | 輸入倍準確度的浮點數    | <code>double</code> |

值得注意的是，使用 `scanf()` 於輸入 `double` 浮點數的變數時，則需用 `%lf`，而輸入 `float` 浮點數的變數時，只需用 `%f` 即可。在輸出方面，不論是 `float` 或 `double` 的變數，皆使用 `%f`。請參閱 2-2c。

### 範例 2-2c

```
01 /* ex2-2c.c */
02 #include <stdio.h>
03 #include <stdlib.h>
04
05 int main()
06 {
07 char ch;
08 float f_num;
09 double d_num;
10 /* 要求輸入各種格式的值 */
11
12 printf("請輸入一字元: ");
13 scanf("%c", &ch);
14
15 printf("請輸入 float 浮點數: ");
16 scanf("%f", &f_num);
17
18 printf("請輸入 double 浮點數: ");
19 scanf("%lf", &d_num);
20
21 /* 將輸入結果輸出至螢幕上 */
22 printf("-----\n");
23 printf("您輸入的字元是 %c\n", ch);
24 printf("您輸入的 float 浮點數是 %f\n", f_num);
25 printf("您輸入的 double 浮點數是 %f\n", d_num);
26 system("PAUSE");
27 return 0;
28 }
```

### 輸出結果

```
請輸入一字元: p
請輸入 float 浮點數: 123.456
請輸入 double 浮點數: 123.456

您輸入的字元是 p
您輸入的 float 浮點數是 123.456001
您輸入的 double 浮點數是 123.456000
```

## 【程式剖析】

輸出 float 的變數值時會有誤差，而 double 則不會。

### 2-2-2 特殊輸入的處理

scanf() 在輸入時可以加入一些特殊符號當作分隔字元，如冒號 (:) 或分號 (;)，而這些特殊符號在輸入資料時，也必需加以輸入這些特殊符號，否則 scanf 函數無法成功讀取輸入的資料。請參閱範例 2-2d 及其說明。



#### 範例 2-2d

```
01 /* ex2-2d.c */
02 #include <stdio.h>
03 #include <stdlib.h>
04 int main()
05 {
06 int hour, min, sec;
07 int year, month, days;
08 char letter1, letter2;
09 printf("請輸入兩個字元，中間以空白隔開: ");
10 scanf("%c %c", &letter1, &letter2); /* 輸入兩個字元 */
11 printf("請輸入現在的時間? (hour:min:sec): ");
12 scanf("%d:%d:%d", &hour, &min, &sec); /* 輸入時間 */
13 printf("請輸入現在的日期? (month-day-year): ");
14 scanf("%d-%d-%d", &month, &days, &year); /* 輸入日期 */
15 printf("\n");
16 printf("兩個字元分別為: %c 和 %c\n", letter1, letter2);
17 printf("現在時間是: %d 點 %d 分 %d 秒\n", hour, min, sec);
18 printf("現在的日期是: 西元 %d %d 月 %d 日\n", year, month, days);
19 system("PAUSE");
20 return 0;
21 }
```

#### 輸出結果

```
請輸入兩個字元，中間以空白隔開: a b
請輸入現在的時間? (hour:min:sec): 11:15:30
請輸入現在的日期? (month-day-year): 11-30-2018
```



```
兩個字元分別為： a 和 b
現在時間是： 11 點 15 分 30 秒
現在的日期是： 西元 2018 11 月 30 日
```

## 【程式剖析】

此範例第 10 行輸入資料的分隔字元為空白，但第 12 行與第 14 行則分別為：  
和 -。若輸入時不含這些分隔字元，將會產生錯誤的結果。



### 除錯題

以下是阿三學完這一節後所寫的程式，請您訂正之。

1. 此程式試以 123.45 浮點數輸入。

```
/* bugs2-2-1.c */
#include <stdio.h>
#include <stdlib.h>
int main()
{
 double x;
 printf("請輸入一浮點數: ");
 scanf("%d", x);
 printf("Number is %f\n", x);
 system("PAUSE");
 return 0;
}
```

2. 請以 123.12 和 123.12 資料輸入。

```
/* bugs2-2-2.c */
#include <stdio.h>
#include <stdlib.h>

int main()
{
 float f_num;
 double d_num;

 printf("輸入單準確度浮點數: ");
 scanf("%f", &f_num);

 printf("請輸入倍準確度浮點數: ");
 scanf("%f", &d_num);
}
```

```
printf("f_number = %f\n", f_num);
printf("d_number = %f\n", d_num);
system("PAUSE");
return 0;
}
```

3. 請以輸入的提示訊息之格式輸入資料，查看是否有問題。

```
/* bugs2-2-3.c */
#include <stdio.h>
#include <stdlib.h>
int main()
{
 int hour, min, sec;
 int year, month, days;

 printf("請輸入現在的時間? (hour:min:sec): ");
 scanf("%d:%d:%d", &hour, &min, &sec); /* 輸入時間 */
 printf("請輸入現在的日期? (month/day/year): ");
 scanf("%d-%d-%d", &month, &days, &year); /* 輸入日期 */
 printf("\n");

 printf("現在時間是: %d 點 %d 分 %d 秒\n", hour, min, sec);
 printf("現在的日期是: 西元 %d %d 月 %d 日\n", year, month, days);
 system("PAUSE");
 return 0;
}
```



1. 試先定義變數 var1、var2 與 var3 為字元、整數與浮點數之資料型態，然後以 scanf 讀取使用者輸入資料後，利用 printf 輸出每一變數的內容。
2. 試利用 scanf 函數，要求使用者依序輸入五個浮點數，並將輸入資料顯示至螢幕上。
3. 試以 scanf 函數，要求使用者輸入一個日期（包括年、月、日），並以斜線 / 分隔輸入的值。

## 2-3 問題演練

1. 試問下一個程式的輸出結果為何？

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
 char letter = 'a';
 printf("|%c| using %%c\n", letter);
 printf("|%10c| using %%10c\n", letter);
 printf("|%-10c| using %%-10c\n", letter);
 system("PAUSE");
 return 0;
}
```

2. 試問下一個程式的輸出結果為何？

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
 int number1 = 12345;
 double number2 = 123.456;
 printf("|%d| using %%d\n", number1);
 printf("|%10d| using %%10d\n", number1);
 printf("|%2d| using %%2d\n", number1);
 printf("|%10.3f| using %%10.3f\n", number2);
 system("PAUSE");
 return 0;
}
```

3. 試問下一程式之輸出結果為何？

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
 double f = 678.90;
 printf("The original floating point is %f\n", f);
 printf("|%f|\n", string);
 printf("|%3.2f|\n", f);
 printf("|%7.2f|\n", f);
 printf("|%-7.2f|\n", f);
 printf("|%7.0f|\n", f);
 system("PAUSE");
 return 0;
}
```

#### 4. 試問下一程式之輸出結果為何？

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
 printf("Hello, how are you?");
 printf("\r");
 printf("\r Hello, how are you?\n");
 printf("\t Hello, how are you?\n");
 printf("\bHello, how are you?\n");
 system("PAUSE");
 return 0;
}
```

## 2-4 程式實作

### 1. 試修正此程式錯誤之處。

```
#include <stdio.h>
#include <stdlib.h>
int main()
(
 int a, b;
 printf("Please enter the date today: ")
 scanf('%d %d %d', a, b, c);
 print("Today is %d:%d:%d", a, b);
 System("PASUE");
)
```

試以 `scanf` 函數，要求使用者輸入六位同學的 C 語言分數（`double` 的浮點數），每三位同學印成一列。請以 `printf` 函數配合欄位寬的設定，美化其輸出結果。