

# 格式的輸出與輸入

# 輸出函數

1

➡ printf (print format) – 格式化列印

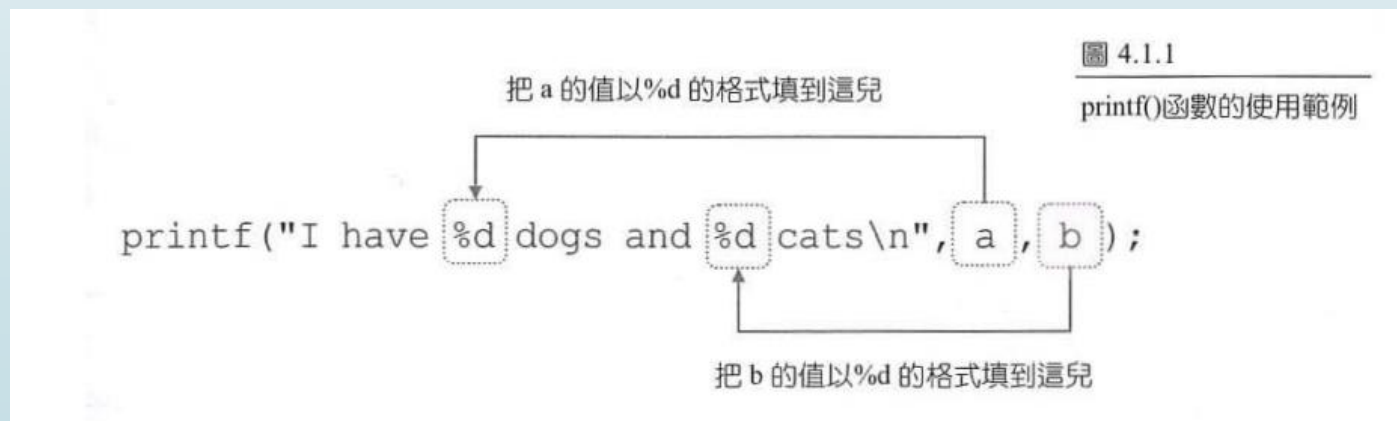
➡ 語法

➡ printf("格式字串",項目1,項目2,...)

➡ 雙引號

➡ 項目可以是常數、變數或是運算式

➡ 格式碼(如%d)依照順序讀取參數



# printf常用格式碼

表 4.1.1 printf() 函數常用的格式碼

| 格式碼 | 說 明         | 格式碼 | 說 明      |
|-----|-------------|-----|----------|
| %c  | 字元          | %%  | 印出百分號    |
| %d  | 十進位整數       | %o  | 無號八進位整數  |
| %ld | 長整數         | %s  | 字串       |
| %e  | 浮點數，指數 e 型式 | %u  | 無號十進位整數  |
| %f  | 浮點數，小數點型式   | %x  | 無號十六進位整數 |

# printf常用的跳脫序列


表 4.1.2 使用於 printf() 函數的跳脫序列

| 跳脫序列 | 功能    | 跳脫序列 | 功能            |
|------|-------|------|---------------|
| \a   | 警告音   | \"   | 印出雙引號         |
| \b   | 倒退    | \\   | 印出反斜線         |
| \n   | 換行    | \/   | 印出斜線          |
| \r   | 歸位    | \d   | ASCII 碼（八進位）  |
| \t   | 跳格    | \x   | ASCII 碼（十六進位） |
| \'   | 印出單引號 |      |               |

# 練習

4

➡ 輸出以下結果

 Z:\C2講義\C語言\C語言教學手冊\課程範例\04\_04\_練習-百分比符號及雙引號印出.exe

"25%"的學生來自小康家庭  
請按任意鍵繼續 . . .


# 控制輸出欄位的寬度

- ➡ **%**修飾符號欄位寬度.小數點寬度格式碼
  - ➡ **%**格式碼
  - ➡ 修飾符號
    - ➡ - ( 靠左對齊 )
    - ➡ + ( 將數值正負號表示出來 )
    - ➡ 0 ( 補0 )
    - ➡ 空白 ( 數值為正值時，留一格空白；為負值時，顯示負號 )
  - ➡ 欄位寬度
  - ➡ .小數點寬度

# 練習

6

➡ 輸出以下結果，注意寬度為6

 Z:\C2講義\C語言\C語言教學手冊\課程範例\04\_06\_練習-輸出寬度.exe

```
num1=    25公里  
num2=1024  公里  
num3= 12.35公里  
請按任意鍵繼續 . . .
```

# Printf修飾子範例

| 資料內容    | 格式      | 執行結果   |   |   |   |   |   |   |   |   |   |   |
|---------|---------|--|---|---|---|---|---|---|---|---|---|---|
| 12345   | %10d    | <table><tr><td></td><td></td><td></td><td></td><td></td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr></table>      |   |   |   |   |   | 1 | 2 | 3 | 4 | 5 |
|         |         |  |   |   | 1 | 2 | 3 | 4 | 5 |   |   |   |
| 12345   | %+d     | <table><tr><td>+</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr></table>   | + | 1 | 2 | 3 | 4 | 5 |   |   |   |   |
| +       | 1       | 2  | 3 | 4 | 5 |   |   |   |   |   |   |   |
| 12345   | %09d    | <table><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr></table>           | 0 | 0 | 0 | 0 | 1 | 2 | 3 | 4 | 5 |   |
| 0       | 0       | 0  | 0 | 1 | 2 | 3 | 4 | 5 |   |   |   |   |
| 12345   | %-10d   | <table><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td></td><td></td><td></td><td></td><td></td></tr></table>      | 1 | 2 | 3 | 4 | 5 |   |   |   |   |   |
| 1       | 2       | 3  | 4 | 5 |   |   |   |   |   |   |   |   |
| 12345   | % d.    | <table><tr><td></td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr></table>  |   | 1 | 2 | 3 | 4 | 5 |   |   |   |   |
|         | 1       | 2  | 3 | 4 | 5 |   |   |   |   |   |   |   |
| 123.456 | %7.2f   | <table><tr><td></td><td>1</td><td>2</td><td>3</td><td>.</td><td>4</td><td>6</td></tr></table>                                |   | 1 | 2 | 3 | . | 4 | 6 |   |   |   |
|         | 1       | 2  | 3 | . | 4 | 6 |   |   |   |   |   |   |
| 123.456 | %010.3f | <table><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>2</td><td>3</td><td>.</td><td>4</td><td>5</td><td>6</td></tr></table> | 0 | 0 | 0 | 1 | 2 | 3 | . | 4 | 5 | 6 |
| 0       | 0       | 0  | 1 | 2 | 3 | . | 4 | 5 | 6 |   |   |   |
| 123.456 | %+10.4f | <table><tr><td></td><td>+</td><td>1</td><td>2</td><td>3</td><td>.</td><td>4</td><td>5</td><td>6</td><td>0</td></tr></table>  |   | + | 1 | 2 | 3 | . | 4 | 5 | 6 | 0 |
|         | +       | 1  | 2 | 3 | . | 4 | 5 | 6 | 0 |   |   |   |

圖 4.1.4

printf() 函數的修飾子  
之應用範例



# 以不同進位的型式輸出

## ■ 八進制

■ %o (octal)

## ■ 十六進制


■ %x (hexadecimal)

```
#include <stdio.h>

int main(void)
{
    printf("42的八進位是%o\n",42);
    printf("42的十六進位是%x\n",42);
    system("pause");
    return 0;
}
```

# 練習-輸出浮點數格式

➡ 輸出以下結果

 Z:\C2講義\C語言\C語言教學手冊\課程範例\04\_09\_練習-輸出浮點數格式.exe

```
a=15  
以浮點數型態印出: 15.000000  
以指數型態印出: 1.500000e+001  
請按任意鍵繼續 . . .
```

# 輸入函數

## ➡ scanf

➡ scanf("格式文字",&變數1,&變數2,...)

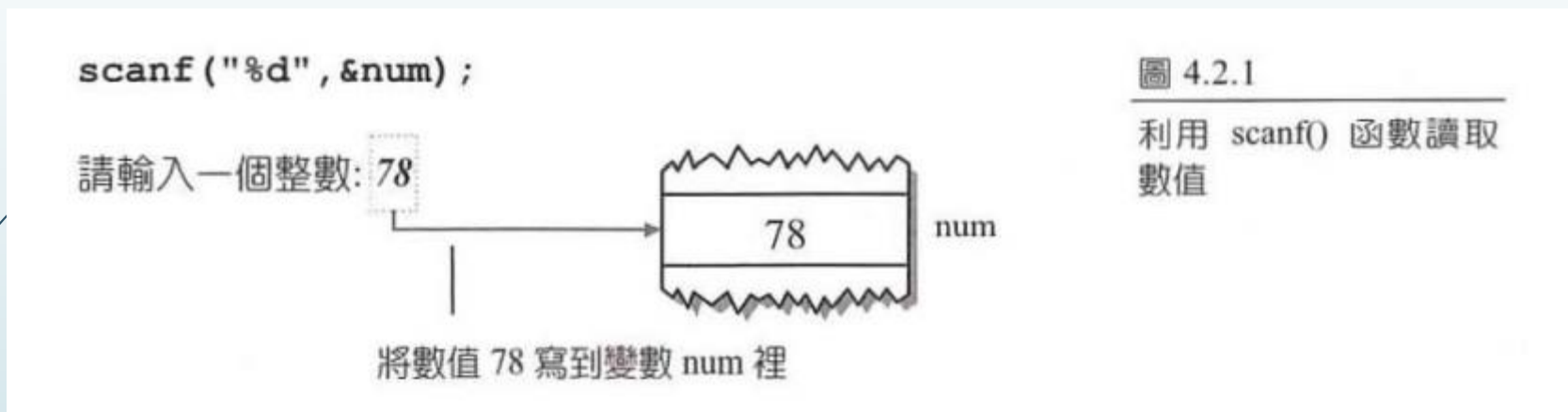
➡ &將資料的值設給某個變數時，實際上是將值存在這個變數所在的位址

```
#include <stdio.h>

int main(void)
{
    int num;
    printf("請輸入一個整數:");
    scanf("%d",&num);
    printf("num = %d\n",num);
    system("pause");
    return 0;
}
```

## 範例解說

- 輸入整數78，按下Enter鍵之後，scanf函數會根據變數num的位址把整數78寫到num裡



- 若忘記加上「&」符號，則會發生不可預期的錯誤

# 連續輸入兩個數值

```
#include <stdio.h>

int main(void)
{
    int a,b;
    printf("請輸入兩個整數:");
    scanf("%d %d",&a,&b);
    printf("%d+%d=%d\n",a,b,a+b);
    system("pause");
    return 0;
}
```

# 輸入格式說明

- 格式字串若是以空白區隔格式碼，則在輸入資料時可以用
  - 換行(按Enter鍵)
  - 跳格(按Tab鍵)
  - 空白
- 若是以逗號「,」區隔格式碼時，則輸入資料時就必須要輸入都逗號做區隔內容

# scanf所使用的輸入格式碼

表 4.2.1 scanf() 函數常用的輸入格式

| 輸入格式 | 輸入敘述                         | 輸入格式 | 輸入敘述   |
|------|------------------------------|------|--------|
| %c   | 字元                           | %s   | 字串     |
| %d   | 十進位整數                        | %o   | 八進位整數  |
| %f   | 浮點數                          | %x   | 十六進位整數 |
| %lf  | 倍精度浮點數（注意%lf 裡的 l 是英文小寫字母 l） |      |        |

# 練習-輸入十六進制轉成十進制輸出

```
#include <stdio.h>

int main(void)
{
    int num;
    printf("請輸入十六進制整數:");
    scanf("%x",&num);
    printf("%x的十進位為%d\n",num,num);
    system("pause");
    return 0;
}
```



# 輸入字元

```
#include <stdio.h>

int main(void)
{
    char ch;
    printf("請輸入一個字元:");
    scanf("%c",&ch);
    printf("ch=%c,ASCII code is %d\n",ch,ch);
    system("pause");
    return 0;
}
```

- ▶ 若在輸入字元前加一個空白，則在scanf前也需加上空白才可以正常擷取字元，且代表意思為接收第一個非空白字元

# 輸入字串

17

- 字串(string)是由兩個以上的字元所組成，因此我們宣告字元陣列來存放字串，在C語言中字串可看成是字元所組成的陣列。
- 宣告語法
  - char 字串變數[字串長度]
- C語言的字串必須以「0」作為結尾，因此實際可儲存的字元數會比字串長度少1
- scanf
  - %s讀取字串，且變數不需要加「&」，因陣列名稱即為陣列的位址
  - 讀取字串時，會找到第一個非空白字元後，再一個字元一個字元讀取，直到下一個空白為止
  - 不會檢查輸入的字串長度是否小於所宣告的長度大小，若超過時可能會發生不可預期的錯誤

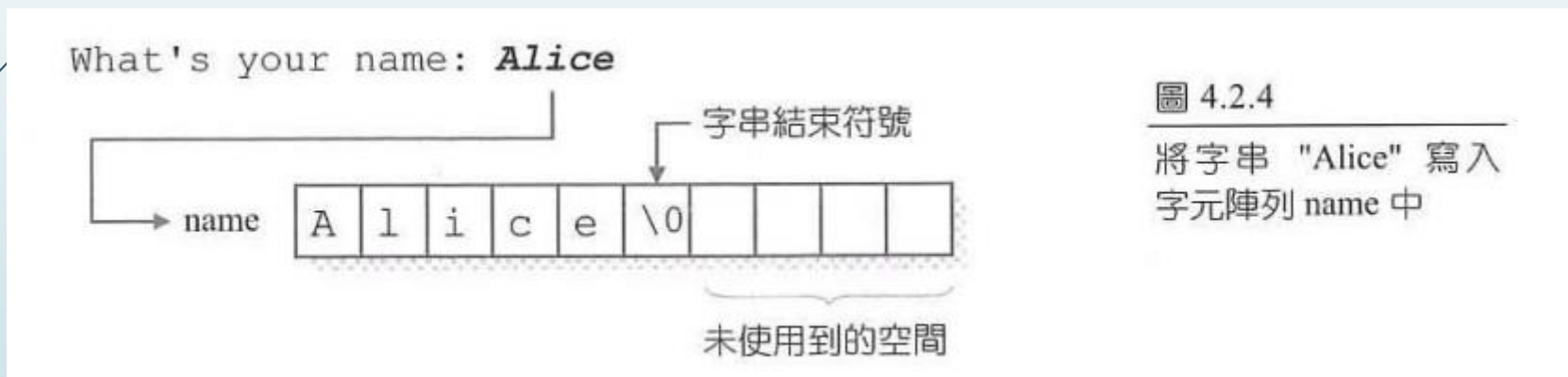
# 輸入字串

```
#include <stdio.h>

int main(void)
{
    char name[10];
    printf("請輸入你的名字:");
    scanf(" %s",name);
    printf("Hi,%s,How are you?\n",name);
    system("pause");
    return 0;
}
```

# 程式解說

- C語言的字串必須以跳脫序列「\0」做結尾，用來識別字串到此結束，
- printf函數填上字元陣列時，便會逐字印出字串中的字元，直到「\0」為止




# scanf讀取數值的方式

- ➡ 找到第一個非空白的字元後，在讀入數值，讀取的動作會持續到讀進的內容為數字為止
- ➡ 如果讀入的數值之後還有其他字元，當作下一個輸入所要讀取的資料

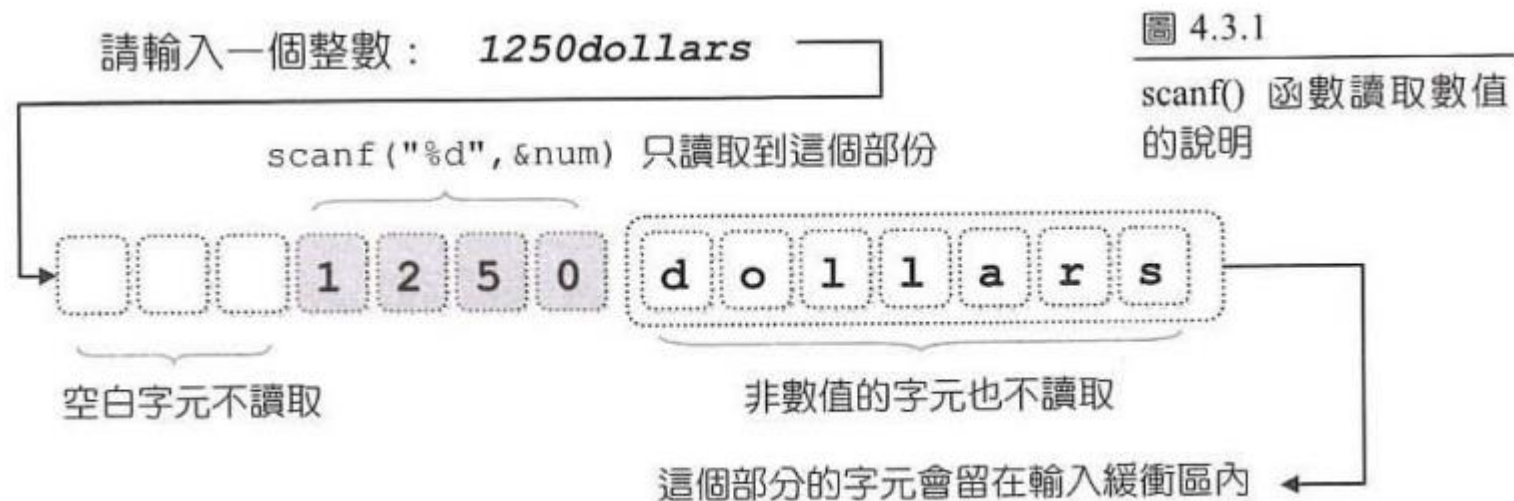
```
#include <stdio.h>

int main(void)
{
    int num;
    printf("請輸入一個整數:");
    scanf("%d",&num);
    printf("num = %d\n",num);
    system("pause");
    return 0;
}
```

# 程式解說

 Z:\C2講義\C語言\C語言教學手冊\課程範例\04\_20\_scanf讀取數值的方式.exe

```
請輸入一個整數: 1250dollars  
num = 1250  
請按任意鍵繼續 . . .
```



# 讀取輸入緩衝區內殘留的資料

```
#include <stdio.h>

int main(void)
{
    int num;
    char str[10];
    printf("請輸入一個整數:");
    scanf("%d",&num);
    printf("num = %d\n",num);
    printf("請輸入一個字串:");
    scanf("%s",str);
    printf("str = %s\n",str);
    system("pause");
    return 0;
}
```



# 程式解說

🖥️ Z:\C2講義\C語言\C語言教學手冊\課程範例\04\_22\_讀取輸入緩衝區內殘留的資料.exe

```
請輸入一個整數: 1250dollars
num = 1250
請輸入一個字串: str = dollars
請按任意鍵繼續 . . .
```

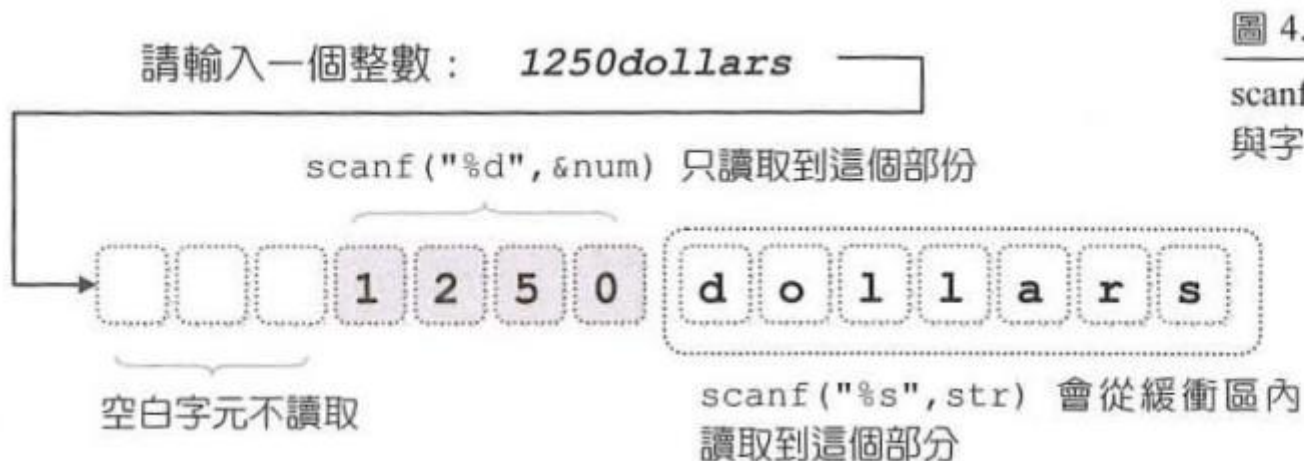


圖 4.3.2


scanf() 函數讀取數值  
與字串的說明



# 讀取字元時常見的錯誤

```
#include <stdio.h>

int main(void)
{
    int num;
    char ch;
    printf("請輸入一個整數:");
    scanf("%d",&num);
    printf("請輸入一個字元:");
    scanf("%c",&ch);
    printf("num = %d, ASCII of ch =%d\n",num,ch);
    system("pause");
    return 0;
}
```

 Z:\C2講義\C語言\C語言教學手冊\課程範例\04\_24\_讀取字元時常見的錯誤.exe

```
請輸入一個整數:22
請輸入一個字元:num = 22, ASCII of ch =10
請按任意鍵繼續 . . .
```

# 程式解說

- 輸入整數後按下的「Enter」，在dos或windows的環境內會被解譯為carriage return 與 line feed，意思違歸位且換行，而歸位的ASCII碼為13，另一個則是換行的ASCII碼為10
- 當scanf()接收到「歸位」字元時，便會判定資料已經輸入完畢，就把輸入的整數22寫入到num變數中，但此時的「換行」字元還在緩衝區內，導致scanf()函數等不及輸入，便先讀取此換行字元

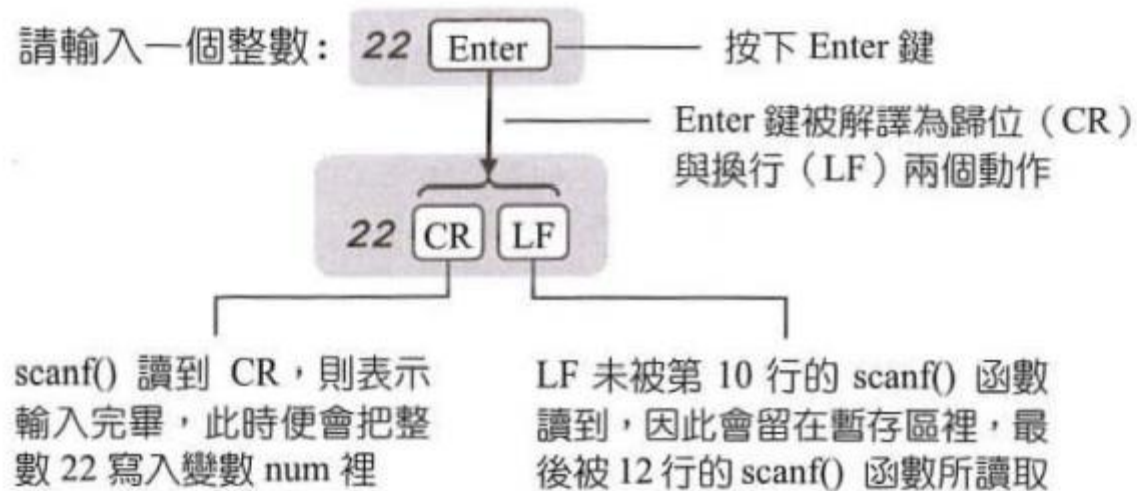


圖 4.3.3

scanf() 函數處理  
Enter 鍵的說明

## 解決錯誤的方法

- 讓scanf()在讀取字元時，可以跳過換行字元不讀取，因換行字元屬於「不可列印字元」，而在scanf()的格式字串中，只要在「%c」之前留下一個空白，便可跳過「不可列印字元」，直到讀取一個可列印字元

# 清除緩衝區的資料

- 殘留在緩衝區內的資料會導致下一個scanf()讀取時的錯誤，因此需要適時的清除緩衝區內的資料
- fflush(stdin)
  - stdin(standard input)

```
#include <stdio.h>

int main(void)
{
    int num;
    char ch;
    printf("請輸入一個整數:");
    scanf("%d",&num);
    fflush(stdin);
    printf("請輸入一個字元:");
    scanf("%c",&ch);
    printf("num = %d, ASCII of ch =%d\n",num,ch);
    system("pause");
    return 0;
}
```

# 輸出、輸入的字元函數

## ► getchar與putchar

```
#include <stdio.h>


int main(void)
{
    char ch;
    printf("輸入一個字元:");
    ch = getchar();
    fflush(stdin);
    printf("您輸入的字元:");
    putchar(ch);
    putchar('\n');
    system("pause");
    return 0;
}
```

# getche與getch

- `getche()`與`getch()`
  - 按下鍵盤一個字元，則馬上被接收
  - `getchar()`則是要按下Enter來結束字元輸入的動作
  - `getche(echo)`表示是否回應到螢幕上
  - 需要引用`conin.h`(console input/output)

# 程式範例

```
#include <stdio.h>
#include <conio.h>
int main(void)
{
    char ch;
    printf("輸入一個字元:");
    ch = getche();
    printf(" 您輸入的字元:%c\n", ch);
    printf("輸入一個字元:");
    ch = getch();
    printf(" 您輸入的字元:%c\n", ch);
    system("pause");
    return 0;
}
```

 Z:\C2講義\C語言\C語言教學手冊\課程範例\04\_30\_getche與getch.exe

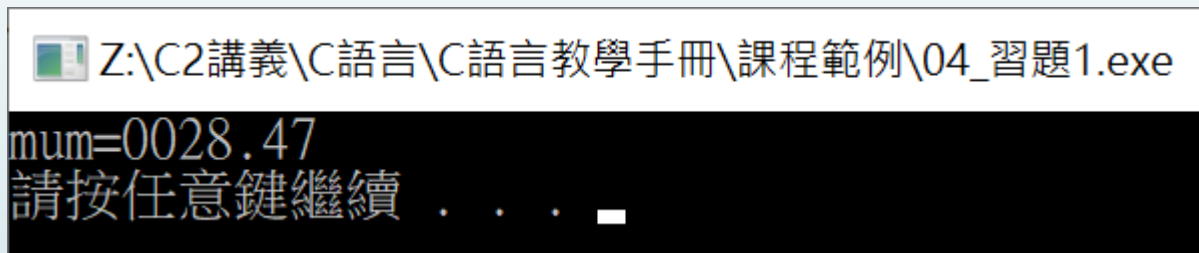
輸入一個字元:8 您輸入的字元:8  
輸入一個字元: 您輸入的字元:0  
請按任意鍵繼續 . . .



# 習題

31

- 是撰寫一程式，將浮點數變數 num=28.47f 如下圖輸入(小數點前有4位，小數點後有2位，不滿欄位長度時補0)




```
Z:\C2講義\C語言\C語言教學手冊\課程範例\04_習題1.exe  
num=0028.47  
請按任意鍵繼續 . . .
```



# 習題

32

- 是撰寫一程式，利用scanf函數輸入兩個整數，然後以printf函數印出兩個整數相乘的結果

 Z:\C2講義\C語言\C語言教學手冊\課程範例\04\_習題2.exe

請輸入兩個整數:20 40

num1 = 20

num2 = 40

num1 \* num2 = 800

請按任意鍵繼續 . . .