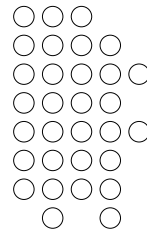


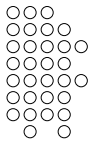
第四章 格式化的輸出與輸入

printf() 函數的使用方法
scanf() 函數的使用方法
各種列印格式碼與輸入格式碼
字元的輸入與輸出函數



4.1 輸出函數 printf()

輸出函數 printf() (1/3)



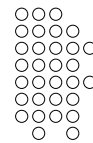
- printf 是由 print (列印) 與 format (格式) 二字組成
格式化輸出 (formatted output)
- printf() 函數的使用格式：

printf() 函數的使用格式

```
printf("格式字串", 項目1, 項目2, ...);
```

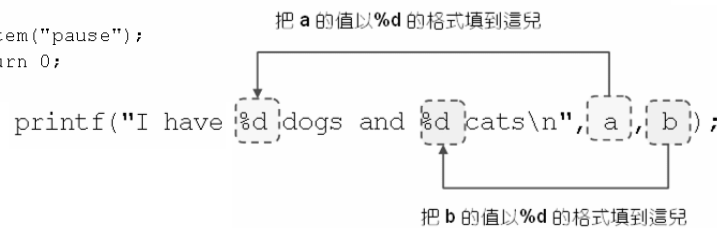
4.1 輸出函數 printf()

輸出函數 printf() (2/3)



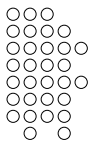
- 下面的程式為使用 printf() 函數的範例：

```
01 /* prog4_1, printf() 函數的使用 */
02 #include <stdio.h>
03 #include <stdlib.h>          /* prog4_1 OUTPUT-----
04 int main(void)              I have 2 dogs and 4 cats
05 {                            -----*/
06     int a=2;
07     int b=4;
08     printf("I have %d dogs and %d cats\n", a,b); /* 呼叫 printf() 函數 */
09
10     system("pause");
11     return 0;
12 }
```



4.1 輸出函數 printf()

輸出函數 printf() (3/3)



- 下面的範例示範了如何印出字串：

```
01 /* prog4_2, 印出字串 */
02 #include <stdio.h>
03 #include <stdlib.h>
04 int main(void)
05 {
06     printf("Have a nice day!!\n"); /* 印出字串內容 */
07
08     system("pause");
09     return 0;
10 }
```

```
/* prog4_2 OUTPUT--
Have a nice day!!
-----*/
```

用於 printf() 的格式碼

- 下表列出了 printf() 函數常用的格式碼：

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	%%	印出百分比號
%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

5

跳脫序列

- 下表列出常用的跳脫序列：

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		
\t	跳格		
\'	印出單引號		

6

跳脫序列與格式碼的應用

- 下面的程式碼是利用格式碼印出字串：

```
01 /* prog4_3, 使用 printf() 函數 */
02 #include <stdio.h>
03 #include <stdlib.h>
04 int main(void)
05 {
06     int num=25;
07     printf("\"%d%%的學生來自小康家庭\"\n", num); /* 印出字串 */
08
09     system("pause");
10     return 0;
11 }
```

格式碼，用來印出整數直 跳脫序列，用來印出雙引號

printf(" \" %d %%" 的學生來自小康家庭 \" \n ", num);

格式碼，用來印出百分比符號 跳脫序列，用來進行換行

跳脫序列，用來印出雙引號

7

控制輸出欄位的寬度

- 設定欄位的寬度：

```
01 /* prog4_4, 印出特定格式 */
02 #include <stdio.h>
03 #include <stdlib.h>
04 int main(void)
05 {
06     int num1=32, num2=1024;
07     float num3=12.3478f;
08
09     printf("num1=%6d 公里\n", num1); /* 以「%6d」格式印出 num1 */
10     printf("num2=%-6d 公里\n", num2); /* 以「%-6d」格式印出 num2 */
11     printf("num3=%6.2f 英哩\n", num3); /* 以「%6.2f」格式印出 num3 */
12
13     system("pause");
14     return 0;
15 }
```

num1 = 32 公里
%6d, 佔 6 格, 靠右對齊

num2 = 1024 公里
%-6d, 佔 6 格, 靠左對齊

num3 = 12.35 英哩
%6.2f, 佔 6 格, 靠右對齊

/* prog4_4 OUTPUT ---
num1= 32 公里
num2=1024 公里
num3= 12.35 英哩
-----*/

8

printf() 函數的修飾子(1/2)

表 4.1.3 printf() 函數的修飾子

修飾子	功能	舉例
-	靠左對齊	%-3d
+	將數值的正負號顯示出來	%+5d
空白	數值為正值時，留一格空白；為負值時，顯示負號	% 6f
0	將固定欄位長度的數值前空白處填上 0（與負號「-」同時使用時，此功能無效）	%07.2f

資料內容	格式	執行結果
12345	%10d	1 2 3 4 5
12345	%+d	+ 1 2 3 4 5
12345	%09d	0 0 0 0 1 2 3 4 5
12345	%-10d	1 2 3 4 5
12345	% d	1 2 3 4 5
123.456	%7.2f	1 2 3 . 4 6
123.456	%10.3f	0 0 0 1 2 3 . 4 5 6
123.456	%+10.4f	+ 1 2 3 . 4 5 6 0

9

printf() 函數的修飾子(2/2)

- printf() 函數修飾子的使用範例：

```

01 /* prog4_5, 使用 printf() 函數的修飾子+ */
02 #include <stdio.h>
03 #include <stdlib.h>
04 int main(void)
05 {
06     int i=1234;
07     printf("i=%+08d\n",i); /* 呼叫 printf() 函數 */
08
09     system("pause");
10     return 0;
11 }

/* prog4_5 OUTPUT--
i+=0001234
-----*/

```

10

以不同進位的型式輸出

- 下面的程式將整數以八進位與十六進位輸出：

```

01 /* prog4_6, 將 10 進位整數以不同的進位系統做輸出 */
02 #include <stdio.h>
03 #include <stdlib.h>
04 int main(void)
05 {
06     printf("42的八進位是 %o\n",42); /* 印出 42 的八進位 */
07     printf("42的十六進位是 %x\n",42); /* 印出 42 的十六進位 */
08
09     system("pause");
10     return 0;
11 }

/* prog4_6 OUTPUT--
42的八進位是 52
42的十六進位是 2a
-----*/

```

11

控制碼必須符合輸出的型態(1/2)

- 錯誤的範例：整數資料以其它型態輸出

```

01 /* prog4_7, 整數資料以其它型態輸出, 錯誤的範例 */
02 #include <stdio.h>
03 #include <stdlib.h>
04 int main(void)
05 {
06     int a=15; /* 宣告整數變數 a, 並設值為 15 */
07
08     printf("a=%d\n",a); /* 印出 a 的值 */
09     printf("以浮點數型態印出: %f\n",a); /* 以%f 格式碼印出 a 的值 */
10     printf("以指數型態印出: %e\n",a); /* 以%e 格式碼印出 a 的值 */
11
12     system("pause"); /* prog4_7 OUTPUT-----
13     return 0; a=15
14 } 以浮點數型態印出: 0.000000
以指數型態印出: 1.910519e-297
-----*/

```

http://www.binaryconvert.com/convert_double.html

12

4.1 輸出函數 printf()

控制碼必須符合輸出的型態 (2/2)

- 修正 prog4_7 的錯誤：

```

01 /* prog4_8, 修正 prog4_7 的錯誤 */
02 #include <stdio.h>
03 #include <stdlib.h>
04 int main(void)
05 {
06     int a=15;          /* 宣告整數變數 a, 並設置為 15 */
07
08     printf("a=%d\n", a);          /* 印出 a 的值 */
09     printf("以浮點數型態印出: %f\n", (float) a); /* 以浮點數型態印出 a */
10     printf("以指數型態印出: %e\n", (double) a); /* 以指數型態印出 a */
11
12     system("pause");          /* prog4_8 OUTPUT-----
13     return 0;                a=15
14 }                             以浮點數型態印出: 15.000000
                                以指數型態印出: 1.500000e+001
                                -----*/

```

%e 或是 %f 都預期對應的資料是 double 型態, 如果用 float 時會自動轉為 double

記憶體模型



Memory	
Address	Contents
0000	-27.2
0004	354
0008	0.005
000C	-26
0010	H
⋮	⋮
⋮	⋮
FFF8	X
FFFC	75.62

conceptual model

Address: 8 / 16 / 32 / 64 bit

Memory: store data and program

4.2 輸入函數 scanf()

輸入函數 scanf() (1/6)

- scanf() 函數可用來輸入字元、數字或字串
- scanf() 函數的使用格式如下：

```
scanf("格式字串", &變數1, &變數2, ...);
```

& is the "address-of" operator

scanf("格式字串", 記憶體位址1, 記憶體位址2, ...);

4.2 輸入函數 scanf()

輸入函數 scanf() (2/6)

- 由鍵盤輸入一個整數的範例：

```

01 /* prog4_9, 使用 scanf() 函數 */ scanf("%d", &num);
02 #include <stdio.h>
03 #include <stdlib.h>
04 int main(void)
05 {
06     int num;
07
08     printf("請輸入一個整數:");
09     scanf("%d", &num);          /* 由鍵盤輸入整數, 並指定給 num 存放 */
10     printf("num=%d\n", num);    /* 印出 num 的內容 */
11
12     system("pause");          /* prog4_9 OUTPUT---
13     return 0;                請輸入一個整數: 78
14 }                             num=78
                                -----*/

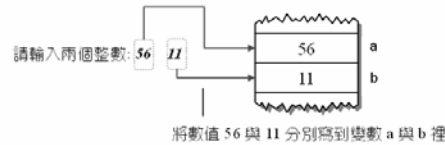
```



輸入函數 scanf() (3/6)

- 由鍵盤上輸入兩個整數的範例：

```
01 /* prog4_10, 使用 scanf() 函數 scanf("%d %d",&a,&b);
02 #include <stdio.h>
03 #include <stdlib.h>
04 int main(void)
05 {
06     int a,b;
07
08     printf("請輸入兩個整數: ");
09     scanf("%d %d",&a,&b); /* 由鍵盤輸入二個數並設定給變數 a、b */
10     printf("%d+%d=%d\n",a,b,a+b); /* 計算總和並印出內容 */
11
12     system("pause"); /* prog4_10 OUTPUT--
13     return 0;        請輸入兩個整數: 56 11
14 }                    56+11=67
                      -----*/
```



17

輸入函數 scanf() (4/6)

- 使用逗號區隔輸入：

```
01 /* prog4_11, 使用逗號區隔輸入格式 */
02 #include <stdio.h>
03 #include <stdlib.h>
04 int main(void)
05 {
06     int a,b;
07
08     printf("請輸入兩個整數, 請用逗號隔開數值: ");
09     scanf("%d,%d",&a,&b); /* 以「,」隔開兩個輸入格式碼 */
10     printf("%d+%d=%d\n",a,b,a+b); /* 計算總和並印出內容 */
11
12     system("pause"); /* prog4_11 OUTPUT-----
13     return 0;        請輸入兩個整數, 請用逗號隔開數值: 14,36
14 }                    14+36=50
                      -----*/
```

14,36 ... ok
14,36 ... ok
14,36 ...not good

18

輸入函數 scanf() (5/6)

- scanf() 函數常用的輸入格式碼：

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)		

19

輸入函數 scanf() (6/6)

- 下面的範例可輸入一個十六進位的數值：

```
01 /* prog4_12, 輸入十六進位數值, 再印出它的十進位 */
02 #include <stdio.h>
03 #include <stdlib.h>
04 int main(void)
05 {
06     int num;
07
08     printf("請輸入十六進位的整數: ");
09     scanf("%x",&num); /* 輸入十六進位數值, 並指定給變數 num */
10     printf("%x的十進位為%d\n",num,num); /* 將十六進位數值以十進位印出 */
11
12     system("pause"); /* prog4_12 OUTPUT----
13     return 0;        請輸入十六進位的整數: 12ab
14 }                    12ab 的十進位為 4779
                      -----*/
```

20

輸入字元應注意的事項 (1/2)

- 輸入空白字元會造成 scanf() 讀取時的錯誤：

```

01 /* prog4_13, 輸入字元 */
02 #include <stdio.h>
03 #include <stdlib.h>
04 int main(void)
05 {
06     char ch;
07
08     printf("Input a character:");
09     scanf("%c",&ch);      /* 由鍵盤輸入字元並指定給變數 ch */
10     printf("ch=%c, ascii code is %d\n",ch,ch);
11     system("pause");
12     return 0;
13 }

```

/* prog4_13 OUTPUT----

Input a character: [] R] → 先輸入一個空白鍵再輸入 R
 ch= , ascii code is 32
 -----*/

1

輸入字元應注意的事項 (2/2)

- 下面的程式碼會讀取第一個不是空白的字元：

```

01 /* prog4_14, 讀取第一個不是空白的字元 */
02 #include <stdio.h>
03 #include <stdlib.h>
04 int main(void)
05 {
06     char ch;
07
08     printf("Input a character:");
09     scanf("%c",&ch);      /* 由鍵盤輸入字元並指定給變數 ch */
10     printf("ch=%c, ascii code is %d\n",ch,ch);
11
12     system("pause");      /* prog4_14 OUTPUT---
13     return 0;
14 }

```

Input a character: [] R] → 先輸入一個空白鍵再輸入 R
 ch=R, ascii code is 82
 -----*/

22

字串的輸入 (1/2)

- C 語言以字元陣列來儲存字串：

字元陣列的宣告

```
char 字串變數[字串長度];
```

```
char str[10];
```

str[0], str[1],, str[9]

str

可容納 10 個字元的陣列

scanf("%s", str); —— 從鍵盤讀取字串，並把它寫到 str 字元陣列裡

scanf("%9s", str); 才是安全的

23

字串的輸入 (2/2)

- 字串輸入的範例：

What's your name: Alice

name [A l i c e \0] [] [] [] [] [] [] [] []

字串結束符號

未使用到的空間

```

01 /* prog4_15, 輸入字串 */
02 #include <stdio.h>
03 #include <stdlib.h>
04 int main(void)
05 {
06     char name[10];      /* 宣告字元陣列 */
07
08     printf("What's your name: ");
09     scanf("%s",name);   /* 輸入字串，並由字元陣列 name 所接收 */
10     printf("Hi, %s, How are you?\n",name); /* 印出字串的內容 */
11     system("pause");
12     return 0;
13 }

```

/* prog4_15 OUTPUT----

What's your name: Alice
 Hi, Alice, How are you?
 -----*/

24

使用 scanf() 常見的問題 (1/2)

- 鍵盤 ⇒ 輸入緩衝區(buffer) ⇒ scanf() 函數

```

01  /* prog4_16, 利用 scanf
02  #include <stdio.h>
03  #include <stdlib.h>
04  int main(void)
05  {
06      int num;
07
08      printf("請輸入一個整數: ");
09      scanf("%d",&num); /* 輸入整數並設定給變數 num 存放 */
10      printf("num=%d\n",num); /* 印出 num 的值 */
11
12      system("pause");
13      return 0;
14  }

```

請輸入一個整數: 1250dollars

scanf("%d",&num) 只讀取到這個部份

空白字元不讀取

非數值的字元也不讀取

這個部分的字元會留在輸入緩衝區內

/* prog4_16 OUTPUT-----
請輸入一個整數: { 1250dollars } → 先輸入三個空白, 再輸入 1250dollars
num=1250
-----*/

25

使用 scanf() 常見的問題 (2/2)

- 讀取輸入緩衝區內殘留的資料:

```

01  /* prog4_17, 讀取輸入緩衝區內殘留的資料 */
02  #include <stdio.h>
03  #include <stdlib.h>
04  int main(void)
05  {
06      int num;
07      char str[10];
08      printf("請輸入一個整數: ");
09      scanf("%d",&num);
10      printf("num=%d\n",num);
11      printf("請輸入一個字串: ");
12      scanf("%s",str); /* 輸入字串 */
13      printf("str=%s\n",str); /* 印出字串的內容 */
14      system("pause");
15      return 0;
16  }

```

請輸入一個整數: 1250dollars

scanf("%d",&num) 只讀取到這個部份

空白字元不讀取

scanf("%s",str) 會從緩衝區內讀取到這個部分

/* prog4_17 OUTPUT-----
請輸入一個整數: { 1250dollars } → 先輸入三個空白, 再輸入 1250dollars
num=1250
請輸入一個字串: str=dollars
-----*/

26

讀取字元時常見的錯誤 (1/2)

- scanf() 如何處理 Enter 鍵?

```

01  /* prog4_18, 讀取到錯誤的字元 */
02  #include <stdio.h>
03  #include <stdlib.h>
04  int main(void)
05  {
06      int num;
07      char ch;
08      printf("請輸入一個整數: ");
09      scanf("%d",&num); /* 由鍵盤輸入整數, 並指定給變數 num */
10      printf("請輸入一個字元: ");
11      scanf("%c",&ch); /* 由鍵盤輸入字元, 並指定給變數 ch */
12      printf("num=%d, ascii of ch=%d\n",num,ch); /* 印出 num 與 ch 的 ascii 碼 */
13      system("pause");
14      return 0;
15  }

```

請輸入一個整數: 22 Enter

按下 Enter 鍵

Enter 鍵被解讀為歸位 (CR) 與換行 (LF) 兩個動作

scanf() 函數讀取到字元 2 時知道第 9 列所需要讀入的整數已經輸入完畢, 第 11 列就會讀接下來的字元

scanf() 函數讀取到字元 2 時知道第 9 列所需要讀入的整數已經輸入完畢, 第 11 列就會讀接下來的字元

/* prog4_18 OUTPUT-----
請輸入一個整數: 22
請輸入一個字元: num=22, ascii of ch=10!
-----*/

沒有輸入字元

27

讀取字元時常見的錯誤 (2/2)

- 下面的程式碼修正 prog4_18 的錯誤:

```

01  /* prog4_19, 修正 prog4_18 的錯誤 */
02  #include <stdio.h>
03  #include <stdlib.h>
04  int main(void)
05  {
06      int num;
07      char ch;
08
09      printf("請輸入一個整數: ");
10      scanf("%d",&num); /* 由鍵盤輸入整數, 並指定給變數 num */
11      printf("請輸入一個字元: ");
12      scanf(" %c",&ch); /* 由鍵盤輸入字元, 並指定給變數 ch */
13      printf("num=%d, ascii of ch=%d\n",num,ch); /* 印出 num 與 ch 的 ascii 碼 */
14      system("pause");
15      return 0;
16  }

```

/* prog4_19 OUTPUT----
請輸入一個整數: 22
請輸入一個字元: k
-----*/

skip all white spaces ' ', '\t', '\n'

28



清除緩衝區的資料 (1/2)

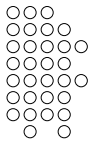
- fflush可用來清除緩衝區的資料

fflush() 函數的用法

```
fflush(stdin); /* 清除緩衝區內的資料 */
```

fflush(輸出串流) 有明確定義
但是 fflush(輸入串流) 沒有明確定義
請避免使用

29



清除緩衝區的資料 (2/2)

- 利用fflush()修正prog4_18的錯誤

```
01 /* prog4_20, 修正 prog4_18 的錯誤 (二) */
02 #include <stdio.h>
03 #include <stdlib.h> /* prog4_20 OUTPUT-----
04 int main(void)      請輸入一個整數: 2332
05 {                  請輸入一個字元: k
06     int num;        num=2332, ascii of ch=107
07     char ch;        -----*/
08
09     printf("請輸入一個整數: ");
10     scanf("%d", &num);
11     fflush(stdin); /* 清空緩衝區內的資料 */
12     printf("請輸入一個字元: ");
13     scanf("%c", &ch);
14     printf("num=%d, ascii of ch=%d\n", num, ch);
15     system("pause");
16     return 0;
17 }
```

30



輸出、輸入字元的函數 (1/2)

- 讀取字元，可用 getchar()

getchar() 函數的用法

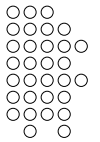
```
字元變數=getchar(); /* 讀取字元，再將它設給字元變數 */
```

- 列印字元，可用 putchar()

putchar() 函數的用法

```
putchar(字元變數); /* 將字元變數的內容列印在螢幕上 */
```

31



輸出、輸入字元的函數 (1/2)

- 下面的例子說明了 getchar() 及 putchar() 函數的使用：

```
01 /* prog4_21, 使用 getchar() 與 putchar() 函數 */
02 #include <stdio.h>
03 #include <stdlib.h>
04 int main(void)
05 {
06     char ch;
07     printf("請輸入一個字元: ");
08     ch=getchar(); /* 輸入一個字元，並指定給變數 ch */
09     printf("您輸入的字元是: ");
10     putchar(ch); /* 將字元 ch 輸出到螢幕上 */
11     putchar('\n'); /* 將換行字元 ch 輸出到螢幕上 */
12
13     system("pause"); /* prog4_21 OUTPUT---
14     return 0;        請輸入一個字元: h
15 }                    您輸入的字元是: h
                      -----*/
```

32



getche() 與 getch() 函數 (1/2)

- 所鍵入的字是否會回應在螢幕上？
 - getche() — yes (有echo)
 - getch() — no (沒有echo)

getche() 與 getch() 函數的用法

```
字元變數=getche(); /* 讀取一個字元，並顯示在螢幕上 */
```

```
字元變數=getch(); /* 讀取一個字元，但不顯示在螢幕上 */
```



getche() 與 getch() 函數 (2/2)

- getche() 與 getch() 函數的使用範例：

```
01 /* prog4_22, 使用 getche() 與 getch() 函數 */
02 #include <stdio.h>
03 #include <conio.h> /* 載入 conio.h 標頭檔 */
04 #include <stdlib.h>
05 int main(void)
06 {
07     char ch;
08     printf("請輸入一個字元: ");
09     ch=getche(); /* 利用 getche() 輸入字元 */
10     printf(" 您輸入的字元是: %c\n",ch);
11
12     printf("請輸入一個字元: ");
13     ch=getch(); /* 利用 getch() 輸入一個字元 */
14     printf(" 您輸入的字元是: %c\n",ch);
15
16     system("pause"); /* prog4_22 OUTPUT-----
17     return 0;        請輸入一個字元: 8 您輸入的字元是: 8
18 }                  請輸入一個字元: 您輸入的字元是: h
                   -----*/
```