printf() 與 scanf()

互動….

C語言人與電腦的互動過程，通常是從主控台，也就是文字模式下開始，為了與程式互動，在主控台下輸出程式執行結果，或是從主控台取得使用者的輸入資料是基本需求。

在C語言中標準輸入輸出是由 stdio 提供，這也就是為何您要在程式的一開頭終是加上：

#include <stdio.h>

將訊息輸出至主控台，稱之為標準輸出（Stand output），C 藉由 printf() 將訊息輸出至主控台，換句話說，舉凡要將指定的文字、數值等輸出至螢幕上，並且執行過後會傳回在螢幕上，就是用printf()，例如：

#include <stdio.h>

**int** main(**void**) {

 //**int** count = printf("This is a test!\n");

 //printf("%d\n", count);

 printf("This is a test!\n");

 **return** 0;

}

"This is a test!\n" 當中包括換行字元\n，共有 16 個字元，因此 count 的值會是 16，顯示結果如下：

This **is** a test!

16

如果在使用 printf() 時要指定整數、浮點數、字元等進行顯示，則要配合格式指定字（format specifier），以下列出幾個可用的格式指定碼：

|  |  |
| --- | --- |
| %c | 以字元方式輸出 |
| %d | 以 10 進位整數輸出 |
| %o | 以 8 進位整數方式輸出 |
| %u | 無號整數輸出 |
| %x, %X | 將整數以 16 進位方式輸出 |
| %f | 浮點數輸出 |
| %e, %E | 使用科學記號顯示浮點數 |
| %g, %G | 浮點數輸出，取 %f 或 %e（%f 或 %E），看哪個表示精簡 |
| %% | 顯示 % |
| %s | 字串輸出 |
| %lu | long unsigned 型態的整數 |
| %p | 指標型態 |

常見的跳脫字元有：

* **\n**：用來換行
* **\t**：用來縮排(等同於輸出一個 tab)
* **\’**：用來輸出一個單引號
* **\”**：用來輸出一個雙引號
* **\**：用來輸出一個反斜線

基本上，要顯示的是什麼資料型態，就必須搭配對應資料型態的格式指定字，但 %d 若用來輸出某個字元，將顯示其整數編碼值，若 %c 用來顯示某個整數，將顯示該整數對應編碼的字元，一個使用的範例如下所示：

#include <stdio.h>

**int** main(**void**) {

 printf("顯示字元 %c\n", 'A');

 printf("顯示字元編碼 %d\n", 'A');

 printf("顯示字元編碼 %c\n", 65);

 printf("顯示十進位整數 %d\n", 15);

 printf("顯示八進位整數 %o\n", 15);

 printf("顯示十六進位整數 %X\n", 15);

 printf("顯示十六進位整數 %x\n", 15);

 printf("顯示科學記號 %E\n", 0.001234);

 printf("顯示科學記號 %e\n", 0.001234);

 **return** 0;

}

顯示結果如下所示：

顯示字元 A

顯示字元編碼 65

顯示字元編碼 A

顯示十進位整數 15

顯示八進位整數 17

顯示十六進位整數 F

顯示十六進位整數 f

顯示科學記號 1.234000E-03

顯示科學記號 1.234000e-03

您可以在輸出浮點數時指定精度，例如若為浮點數：

printf("example:%.2f\n", 19.234);

.2 指定小數點後取兩位，執行結果會輸出：

example:19.23

也可以指定輸出時，至少要預留的字元寬度，無論是數值或字串，例如：

printf("example:%6.2f\n", 19.234);

整數 6 表示預留 6 個字元寬度，由於預留了 6 個字元寬度，不足的部份要由空白字元補上，所以執行結果會輸出如下（19.23只佔五個字元，所以補上一個空白在前端）：

example: 19.23

若在 % 之後指定負號，例如 %-6.2f，則表示靠左對齊，沒有指定則靠右對齊，例如：

#include <stdio.h>

**int** main(**void**) {

 printf("example:%6.2f\n", 19.234);

 printf("example:%-6.2f\n", 19.234);

 **return** 0;

}

顯示結果如下：

example: 19.23

example:19.23

若事先無法決定字元寬度，則可以使用 \*，例如：

#include <stdio.h>

**int** main(**void**) {

 printf("%\*d\n", 1, 1);

 printf("%\*d\n", 2, 1);

 printf("%\*d\n", 3, 1);

 **return** 0;

}

printf() 的 \* 將被之後的第一個引數所取代，所以第一個 printf() 將預留一個字元寬度，第二個預留兩個字元寬度，第三個預留三個，顯示結果如下：

1

 1

 1

scanf函数称为格式输入函数，即按用户指定的格式从键盘上把数据输入到指定的变量之中。

**scanf函数的一般形式**

scanf函数是一个标准库函数，它的函数原型在头文件“stdio.h”中。scanf函数的一般形式为：scanf(“格式控制字符串”, 地址表列);
其中，格式控制字符串的作用与printf函数相同，但不能显示非格式字符串，也就是不能显示提示字符串。地址表列中给出各变量的地址。地址是由地址运算符“&”后跟变量名组成的。

例如：&a、&b分别表示变量a和变量b的地址。

这个地址就是编译系统在内存中给a、b变量分配的地址。在C语言中，使用了地址这个概念，这是与其它语言不同的。 应该把变量的值和变量的地址这两个不同的概念区别开来。变量的地址是C编译系统分配的，用户不必关心具体的地址是多少。

**变量的地址和变量值的关系**
在赋值表达式中给变量赋值，如：
 a=567;
则，a为变量名，567是变量的值，&a是变量a的地址。

但在赋值号左边是变量名，不能写地址，而scanf函数在本质上也是给变量赋值，但要求写变量的地址，如&a。这两者在形式上是不同的。&是一个取地址运算符，&a是一个表达式，其功能是求变量的地址。

【例4-7】

1. #include <stdio.h>
2. int main(void){
3. int a,b,c;
4. printf("input a,b,c\n");
5. scanf("%d%d%d",&a,&b,&c);
6. printf("a=%d,b=%d,c=%d",a,b,c);
7. **return** 0;
8. }

在本例中，由于scanf函数本身不能显示提示串，故先用printf语句在屏幕上输出提示，请用户输入a、b、c的值。执行scanf语句，等待用户输入。在scanf语句的格式串中由于没有非格式字符在“%d%d%d”之间作输入时的间隔，因此在输入时要用一个以上的空格或回车键作为每两个输入数之间的间隔。如：
7 8 9
或
7
8
9

**格式字符串**

格式字符串的一般形式为：
 %[\*][输入数据宽度][长度]类型
其中有方括号[]的项为任选项。各项的意义如下。

**1) 类型**
表示输入数据的类型，其格式符和意义如下表所示。

|  |  |
| --- | --- |
| **格式** | **字符意义** |
| d | 输入十进制整数 |
| o | 输入八进制整数 |
| x | 输入十六进制整数 |
| u | 输入无符号十进制整数 |
| f或e | 输入实型数(用小数形式或指数形式) |
| c | 输入单个字符 |
| s | 输入字符串 |

**2) “\*”符**
用以表示该输入项，读入后不赋予相应的变量，即跳过该输入值。如：

1. scanf("%d %\*d %d",&a,&b);

当输入为：1 2 3时，把1赋予a，2被跳过，3赋予b。

**3) 宽度**
用十进制整数指定输入的宽度（即字符数）。例如：

1. scanf("%5d",&a);

输入12345678只把12345赋予变量a，其余部分被截去。又如：

1. scanf("%4d%4d",&a,&b);

输入12345678将把1234赋予a，而把5678赋予b。

**4) 长度**
长度格式符为l和h，l表示输入长整型数据（如%ld）和双精度浮点数（如%lf）。h表示输入短整型数据。

使用scanf函数还必须注意以下几点：

1. scanf函数中没有精度控制，如：scanf("%5.2f",&a);是非法的。不能企图用此语句输入小数为2位的实数。
2. scanf中要求给出变量地址，如给出变量名则会出错。如 scanf("%d",a);是非法的，应改为scnaf("%d",&a);才是合法的。
3. 在输入多个数值数据时，若格式控制串中没有非格式字符作输入数据之间的间隔则可用空格，TAB或回车作间隔。C编译在碰到空格，TAB，回车或非法数据(如对“%d”输入“12A”时，A即为非法数据)时即认为该数据结束。
4. 在输入字符数据时，若格式控制串中无非格式字符，则认为所有输入的字符均为有效字符。

例如：

1. scanf("%c%c%c",&a,&b,&c);

输入 d、e、f 则把'd'赋予a，' ' 赋予b，'e'赋予c。只有当输入为 def 时，才能把'd'赋于a，'e'赋予b，'f'赋予c。

如果在格式控制中加入空格作为间隔，如：

1. scanf ("%c %c %c",&a,&b,&c);

则输入时各数据之间可加空格。

【例4-8】

1. #include <stdio.h>
2. int main(void){
3. char a,b;
4. printf("input character a,b\n");
5. scanf("%c%c",&a,&b);
6. printf("%c%c\n",a,b);
7. **return** 0;
8. }

由于scanf函数"%c%c"中没有空格，输入M N，结果输出只有M。而输入改为MN时则可输出MN两字符。

【例4-9】

1. #include <stdio.h>
2. int main(void){
3. char a,b;
4. printf("input character a,b\n");
5. scanf("%c %c",&a,&b);
6. printf("\n%c%c\n",a,b);
7. **return** 0;
8. }

本例表示scanf格式控制串"%c %c"之间有空格时，输入的数据之间可以有空格间隔。

**5) 如果格式控制串中有非格式字符则输入时也要输入该非格式字符。**
例如：

1. scanf("%d,%d,%d",&a,&b,&c);

其中用非格式符“ , ”作间隔符，故输入时应为：5,6,7。又如：

1. scanf("a=%d,b=%d,c=%d",&a,&b,&c);

则输入应为：a=5,b=6,c=7。

**6) 如输入的数据与输出的类型不一致时，虽然编译能够通过，但结果将不正确。**

【例4-10】

1. #include <stdio.h>
2. int main(void){
3. int a;
4. printf("input a number\n");
5. scanf("%d",&a);
6. printf("%ld",a);
7. **return** 0;
8. }

由于输入数据类型为整型，而输出语句的格式串中说明为长整型，因此输出结果和输入数据不符。如改动程序如下（【例4-11】）：

1. #include <stdio.h>
2. int main(void){
3. long a;
4. printf("input a long integer\n");
5. scanf("%ld",&a);
6. printf("%ld",a);
7. **return** 0;
8. }

运行结果为：
input a long integer
1234567890
1234567890

当输入数据改为长整型后，输入输出数据相等。

【例4-12】

1. #include <stdio.h>
2. int main(void){
3. char a,b,c;
4. printf("input character a,b,c\n");
5. scanf("%c %c %c",&a,&b,&c);
6. printf("%d,%d,%d\n%c,%c,%c\n",a,b,c,a-32,b-32,c-32);
7. **return** 0;
8. }

输入三个小写字母，输出其ASCII码和对应的大写字母。

【例4-13】

1. #include <stdio.h>
2. int main(void){
3. int a;
4. long b;
5. float f;
6. double d;
7. char c;
8. printf("\nint:%d\nlong:%d\nfloat:%d\ndouble:%d\nchar:%d\n",**sizeof**(a),**sizeof**(b),**sizeof**(f),**sizeof**(d),**sizeof**(c));
9. **return** 0;
10. }

输出各种数据类型的字节长度。

如果打算取得使用者的輸入，則可以使用標準輸入的 scanf() 函式，並搭配格式指定字與 & 取址運算子指定給變數(變數：資料要存放的記憶體空間代號)，例如：

#include <stdio.h>

**int** main(**void**) {

 **int** input;

 printf("請輸入數字：");

 scanf("%d", &input);

 printf("您輸入的數字：%d\n", input);

 **return** 0;

}

在程式中先宣告了一個整數變數 input，使用 scanf() 函式時，若輸入的數值為整數，則使用格式指定字 %d，若輸入的是其他資料型態，則必須使用對應的格式指定字，如果是 double，特別注意要使用 %lf 來指定。

您必須告知程式儲存資料的變數位址，為此，必須使用 & 取址運算子，這會將變數的記憶體位址取出，則輸入的數值就知道變數的記憶體位址並儲存之（但字元陣列名稱本身就有位址資訊，故不用 & 來取址，之後說明陣列時會再看到）。

執行結果：

請輸入數字：10

您輸入的數字：10

scanf() 在接受輸入時，可以接受多個值，也可以指定輸入的格式，例如：

#include <stdio.h>

**int** main(**void**) {

 **int** number1, number2;

 printf("請輸入兩個數字，中間使用空白區隔）：");

 scanf("%d %d", &number1, &number2);

 printf("您輸入的數字：%d %d\n", number1, number2);

 printf("請再輸入兩個數字，中間使用-號區隔）：");

 scanf("%d-%d", &number1, &number2);

 printf("您輸入的數字：%d-%d\n", number1, number2);

 **return** 0;

}

在第一個 scanf() 中，指定了使用空白來區隔兩個輸入，而第二個 scanf() 中，指定了使用 - 來區隔兩個輸入，一個執行與輸入的結果如下所示：

請輸入兩個數字，中間使用空白區隔）：10 20

您輸入的數字：10 20

請再輸入兩個數字，中間使用-號區隔）：30-40

您輸入的數字：30-40

scanf() 還可以指定可接受的字元集合，例如若只想接受 1 到 5 的字元，則可以如下：

#include <stdio.h>

**int** main(**void**) {

 **char** str[50];

 printf("請輸入 1 到 5 的字元：");

 scanf("%[1-5]", str);

 printf("輸入的字元為 %s\n", str);

 fflush(stdin); // 清除輸入緩衝區

 printf("請輸入 XYZ 任一字元：");

 scanf("%[XYZ]", str);

 printf("輸入的字元為 %s\n", str);

 **return** 0;

}

上面的 str 宣告，為 C 語言中的 [字串（字元陣列）](https://openhome.cc/Gossip/CGossip/String.html)，scanf() 函式連續讀入符合集合的字元並放到字元陣列中，直到讀到不符合的字元為止，剩下的字元仍會存在輸入緩衝區中，可以直接使用 fflush(stdin) 清除輸入緩衝區，以利進行下一次重新輸入，scanf() 函式若成功，則傳回成功填寫的格式指定字數目，否則直接略過輸入而傳回 0，在學會流程控制語法之後，可以據以判斷該作什麼處理（像是使用 if 來針對不接受輸入時的處理）。

執行結果如下：

請輸入 1 到 5 的字元：146731

輸入的字元為 14

請輸入 XYZ 任一字元：XYZXDX

輸入的字元為 XYZX

您可以使用 %[0-9] 指定取得 0 至 9 的字元，使用 %[A-z] 指定取得 ASCII 表中的 A 到 z 的字元，如果要排除的話，則使用 ^，例如 %[^ABC] 可取得 ABC 字元以外的所有字元。

★

★★

★★★

★★★★

★★★★★

　　　　★

　　　★★

　　★★★

　★★★★

★★★★★

　　　　★

　　　★★★

　　★★★★★

　★★★★★★★

★★★★★★★★★

　　　　★

　　　★　★

　　★　　　★

　★　　　　　★

　　★　　　★

　　　★　★

　　　　★

1\*1=1 1\*2=2 1\*3=3 1\*4=4 1\*5=5 1\*6=6 1\*7=7 1\*8=8 1\*9=9

2\*1=2 2\*2=4 2\*3=6 2\*4=8 2\*5=10 2\*6=12 2\*7=14 2\*8=16 2\*9=18

3\*1=3 3\*2=6 3\*3=9 3\*4=12 3\*5=15 3\*6=18 3\*7=21 3\*8=24 3\*9=27

…