C語言scanf函式用法詳細解釋！

其他 · 發表 2018-12-23

函式名: scanf
功 能: 執行格式化輸入
用 法: int scanf(char *format[,argument,...]);
scanf()函式是通用終端格式化輸入函式，它從標準輸入裝置(鍵盤) 讀取輸入的資訊。可以讀入任何固有型別的資料並自動把數值變換成適當的機內格式。
其呼叫格式為: scanf("<格式化字串>"，<地址表>);
scanf()函式返回成功賦值的資料項數，出錯時則返回EOF。
其控制串由三類字元構成:
1。格式化說明符;
2。空白符;
3。非空白符;

printf ,scanf 的應用和區別

C語言中的printf與scanf函式

（A）　　格式化說明符

格式字元	說明
%a	讀入一個浮點值(僅C99有效)
%A	同上
%c	讀入一個字元
%d	讀入十進位制整數
%i	讀入十進位制，八進位制，十六進位制整數
%o	讀入八進位制整數
%x	讀入十六進位制整數
%X	同上
%c	讀入一個字元
%s	讀入一個字串
%f	讀入一個浮點數
%F	同上
%e	同上
%E	同上
%g	同上
%G	同上
%p	讀入一個指標
%u	讀入一個無符號十進位制整數
%n	至此已讀入值的等價字元數
%[]	掃描字元集合
%%	讀%符號

附加格式說明字元表

scanf 和 printf的一般用法

51微控制器實現scanf和printf函式

gcc中printf/scanf、printf_P/scanf_P的區別與用法

scanf與printf轉換說明符區別

%d,%c,%s,%x各代表什麼

C語言經典一百例

scanf("%*s");

在ListView控制元件中實現修改功能

修飾符　　說明
L/l 長度修飾符　　輸入"長"資料
h 長度修飾符　　輸入"短"資料
W 整型常數　　指定輸入資料所佔寬度
* 星號　　空讀一個數據
hh,ll同上h,l但僅對C99有效。

（B）　　空白字元
空白字元會使scanf()函式在讀操作中略去輸入中的一個或多個空白字元，空白符可以是space,tab,newline等等，直到第一個非空白符出現為止。
（C）　　非空白字元
一個非空白字元會使scanf()函式在讀入時剔除掉與這個非空白字元相同的字元。

注：scanf()控制串知識就介紹到這裡（應該比較齊全了^_^），如有遺漏下次補上。下面將結合實際例程，一一闡述.
三、　scanf()函式的控制串的使用
例1.

```
#include "stdio.h"
int main(void)
{
int a,b,c;

scanf("%d%d%d",&a,&b,&c);
printf("%d,%d,%d/n",a,b,c);
return 0;
}
```

執行時按如下方式輸入三個值：
3□4□5↙（輸入a,b,c的值）
3，4，5 （printf輸出的a，b，c的值）
（1） &a、&b、&c中的&是地址運算子，分別獲得這三個變數的記憶體地址。
（2） "%d%d%d"是按十進值格式輸入三個數值。輸入時，在兩個資料之間可以用一個或多個空格、tab鍵、回車鍵分隔。
以下是合法輸入方式：
① 3□□4□□□□5↙
② 3↙
4□5↙
③ 3（tab鍵）4↙
5↙

例2.

```
#include "stdio.h"
int main(void)
{
int a,b,c;
scanf("%d,%d,%d",&a,&b,&c);
printf("%d,%d,%d/n",a,b,c);
return 0;
}
```

執行時按如下方式輸入三個值：
3,4,5↙（輸入a,b,c的值）
或者
3,□4,□5↙（輸入a,b,c的值）
3,□□□4,□5↙（輸入a,b,c的值）
.....

都是合法的，但是","一定要跟在數字後面，如：
3□，4,□5↙就非法了，程式出錯。（解決方法與原因後面講）
再如：
1、sacnf()中的變數必須使用地址。

```
    int a, b;
scanf("%d%d",a,b); //錯誤
scanf("%d%d",&a,&b);
```

2、scanf()的格式控制串可以使用其它非空白字元，但在輸入時必須輸入這些字元。
例：

```
scanf("%d,%d",&a,&b);
```

輸入： 3，4↙（逗號與"%d,%d"中的逗號對應）

```
scanf("a=%d,b=%d",&a,&b);
```

輸入： a=3，b=4↙（"a=","b=",逗號與"%d,%d"中的"a=","b="及逗號對應）

3、在用"%c"輸入時，空格和“轉義字元”均作為有效字元。
例：

```
scanf("%c%c%c",&c1,&c2,&c3);
```

輸入：a□b□c↙
結果：a→c1，□→c2，b→c3 (其餘被丟棄)

scanf()函式接收輸入資料時，遇以下情況結束一個數據的輸入：（不是結束該scanf函式，scanf函式僅在每一個數據域均有資料，並按回車後結束）。

① 遇空格、“回車”、“跳格”鍵。
② 遇寬度結束。
③ 遇非法輸入。

問題二：scanf()函式不能正確接受有空格的字串？如: I love you!

```
#include <stdio.h>
int main()
{
    char str[80];

    scanf("%s",str);
    printf("%s",str);
    return 0;
}
```

輸入：I live you!

輸出：I

scanf()函式接收輸入資料時，遇以下情況結束一個數據的輸入：（不是結束該scanf函式，scanf函式僅在每一個數據域均有資料，並按回車後結束）。

① 遇空格、“回車”、“跳格”鍵。

② 遇寬度結束。

③ 遇非法輸入。

所以，上述程式並不能達到預期目的，scanf()掃描到"I"後面的空格就認為對str的賦值結束，並忽略後面的"love you!".這裡要注意是"love you!"還在鍵盤緩衝區（關於這個問題，網上我所見的說法都是如此，但是，我經過除錯發現，其實這時緩衝區字串首尾指標已經相等了，也就是說緩衝區清空了，scanf()函式應該只是掃描stdin流，這個殘存資訊是在stdin中)。我們改動一下上面的程式來驗證一下：

```
#include <stdio.h>
int main()
{
    char str[80];
    char str1[80];
    char str2[80];
```

```
    scanf("%s",str);/*此處輸入:I love you! */
    printf("%s",str);
    sleep(5);/*這裡等待5秒,告訴你程式執行到什麼地方*/
    scanf("%s",str1);/*這兩句無需你再輸入,是對鍵盤盤緩衝區再掃描   */
    scanf("%s",str2);/*這兩句無需你再輸入,是對鍵盤盤緩衝區再掃描    */
    printf("/n%s",str1);
    printf("/n%s",str2);
    return 0;
}
```

輸入：I love you!

輸出：I

love

you!

好了，原因知道了，那麼scanf()函式能不能完成這個任務？回答是：能！別忘了scanf()函式還有一個 %[] 格式控制符（如果對%[]不瞭解的請檢視本文的上篇），請看下面的程式：

```
#include "stdio.h"
int main()
{
    char string[50];

     /*scanf("%s",string);不能接收空格符*/
     scanf("%[^/n]",string);
     printf("%s/n",string);
     return 0;
}
```

問題三：鍵盤緩衝區殘餘資訊問題

```
#include <stdio.h>
int main()
{
    int a;
    char c;
    do
    {
        scanf("%d",&a);
        scanf("%c",&c);
        printf("a=%d     c=%c/n",a,c);
        /*printf("c=%d/n",c);*/
    }while(c!='N');
}
```

scanf("%c",&c);這句不能正常接收字元,什麼原因呢？我們用printf("c=%d/n",c);將C用int表示出來，啓用printf("c=%d/n",c);這一句，看看scanf()函式賦給C到底是什麼，結果是 c=10 ,ASCII值為10是什麼？換行即/n.對了，我們每擊打一下"Enter"鍵，向鍵盤緩衝區發去一個“回車”(/r),一個“換行"(/n),在這裡/r被scanf()函式處理掉了（姑且這麼認為吧^_^），而/n被scanf()函式“錯誤”地賦給了c.

解決辦法：可以在兩個scanf()函式之後加個fflush(stdin);，還有加getch(); getchar(); 也可以，但是要視具體scanf()語句加那個，這裡就不分析了，讀者自己去摸索吧。但是加fflush(stdin);不管什麼情況都可行。

函式名: fflush

功 能: 清除一個流

用 法: int fflush(FILE *stream);

```
#include <stdio.h>
int main()
{
    int a;
    char c;
```

```
    do
    {
        scanf("%d",&a);
        fflush(stdin);
        scanf("%c",&c);
        fflush(stdin);
        printf("a=%d     c=%c/n",a,c);
    }while(c!='N');
}
```

這裡再給一個用“空格符”來處理緩衝區殘餘資訊的示例：

執行出錯的程式：

```
#include <stdio.h>
int main()
{
    int i;
    char j;
    for(i = 0;i < 10;i++)
    {
        scanf("%c",&j);/*這裡%前沒有空格*/
    }
}
```

使用了空格控制符後：

```
#include <stdio.h>
int main()
{
    int i;
    char j;
    for(i = 0;i < 10;i++)
    {
```

```
        scanf(" %c",&j);/*注意這裡%前有個空格*/
    }
}
```

可以執行看看兩個程式有什麼不同。

問題四　如何處理scanf()函式誤輸入造成程式死鎖或出錯?

```
#include <stdio.h>
int main()
{
int a,b,c; /*計算a+b*/
scanf("%d,%d",&a,&b);
c=a+b;
printf("%d+%d=%d",a,b,c);
}
```

如上程式，如果正確輸入a,b的值，那麼沒什麼問題，但是，你不能保證使用者每一次都能正確輸入，一旦輸入了錯誤的型別，你的程式不是死鎖，就是得到一個錯誤的結果,呵呵，這可能所有人都遇到過的問題吧?

解決方法：scanf()函式執行成功時的返回值是成功讀取的變數數,也就是說，你這個scanf()函式有幾個變數，如果scanf()函式全部正常讀取，它就返回幾。但這裡還要注意另一個問題，如果輸入了非法資料，鍵盤緩衝區就可能還個有殘餘資訊問題。

正確的例程:

```
#include <stdio.h>
int main()
{
int a,b,c; /*計算a+b*/
while(scanf("%d,%d",&a,&b)!=2)fflush(stdin);
c=a+b;
printf("%d+%d=%d",a,b,c);
}
```

scanf函式探討

1.空白符問題

```
#include <stdio.h>
main()
{
int a;
printf( "input the data/n ");
scanf( "%d/n ",&a);//這裡多了一個回車符/n
printf( "%d ",a);
return 0;
}
```

結果要輸入兩個數程式才結束，而不是預期的一個。why?
原因：用空白符結尾時，scanf會跳過空白符去讀下一個字元，所以你必須再輸入一個數。這裡的空白符包括

空格，製表符，換行符，回車符和換頁符。所以如果你用scanf("%d ",&a)也會出現

同樣的問題。
解決方法：這種錯誤大多是輸入的時候不小心，多注意一點就好了。這種問題也不好檢查，編譯沒有問題，

一個空格也不容易看出來。當你的程式出現上面的問題時，自己對照檢查一下就可以了。

2.緩衝區問題
這是一個非常容易錯的地方，我就錯過多次。

```
#include <stdio.h>
main()
{
int n = 5;
char c[n];
for(int i = 0; i < n; i++)
c[i] = scanf( "%c ",&c[i]);
printf(c);
return 0;
}
```

如果輸入:

```
a
b
c
```

那麼迴圈就會“提前”結束了.
原因：輸入a和第一個回車後，a和這個回車符都留在緩衝區中。第一個scanf讀取了

a，但是輸入緩衝區裡面

還留有一個/n，第二個scanf讀取這個/n。然後輸入b和第二個回車，同樣的，第三個scanf讀取了b，第四個

scanf讀取了第二個回車符。第五個讀取了c。所以五個scanf都執行了，並沒有提前結束。只不過有的scanf

讀取到了回車符而已。
解決方法：把程式改成這樣就可以了：

```
for( i = 0; i < n; i++){
scanf( "%c ",&c[i]);
fflush(stdin);//重新整理緩衝區
}
```

或者不用scanf，而用gets（）函式，如：

```
#include <stdio.h>
main()
{
char c[5];
gets(c);
printf(c);
return 0;
}
```

但要注意：這個函式自動把你最後敲的回車轉換為字元 '/0 '。如果你的輸入超過了陣列的大小，那麼就會產

生錯誤。

3.scanf()函式的引數輸入型別不匹配問題
這是我在csdn論壇上見到的問題，這個錯誤有時候會讓人莫名其妙。

```
#include <stdio.h>
main()
{
int a=123;
char c= 't ';
printf( "input/n ");
scanf( "%d%c ",&a,&c);
scanf( "%d%c ",&a,&c);
scanf( "%d%c ",&a,&c);
printf( "%d/n%c/n ",a,c);
return 0;
}
```

當輸入a 回車 後，會直接跳過下面 2 個scanf語句，直接輸出為
123
t
原因：對於scanf("%d%c ",&a,&c)，scanf語句執行時，首先試圖從緩衝區中讀入一個%d型別的資料，如果和

第一個引數匹配，則繼續從緩衝區中讀取資料和第二個引數進行匹配，依次進行下去，直到匹配完所有的參

數；如果其中有一個引數不匹配，那就從這個地方跳出，忽略這個scanf後面所有的引數，而去執行下一條語

句。
可以用下面的程式驗證一下：

```
#include <stdio.h>
int main()
{
int a=123,b=1;
char c= 't ';
scanf( "%d%d ",&a,&b);
scanf( "%c ",&c);
printf( "%d/n%d/n%c/n ",a,b,c);
return 0;
}
```

輸入：2 回車a 回車
結果是：

```
2
1
a
```

解決方法：scanf()函式執行成功時的返回值是成功讀取的變數數,也就是說，你這個scanf()函式有幾個變數

，如果scanf()函式全部正常讀取，它就返回幾。但這裡還要注意另一個問題，如果輸入了非法資料，鍵盤緩

衝區就可能還個有殘餘資訊問題。
比如：

```
#include <stdio.h>
main()
{
int a=123,b;
while(scanf( "%d%d ",&a,&b)!=2)
fflush(stdin);
printf( "%d/n%d/n ",a,b);
return 0;
}
```

你可以試一下，如果輸入不是數字時，會有什麼反應。

補充：scanf中一種很少見但很有用的轉換字元：[...]和[^...]。

```
#include <stdio.h>
main()
{
char strings[100];
scanf( "%[1234567890] ",strings);
printf( "%s ",strings);
return 0;
}
```

執行，輸入：1234werew後，結果是：1234。
通過執行可以發現它的作用是：如果輸入的字元屬於方括號內字串中某個字元，那麼就提取該字元；如果

一經發現不屬於就結束提取。該方法會自動加上一個字串結束符到已經提取的字元後面。
scanf("%[^1234567890] ",strings); 它的作用是：如果一經發現輸入的字元屬於方括號內字串中某個字元

，那麼就結束提取；如果不屬於就提取該字元。該方法會自動加上一個字串結束符到已經提取的字元後面

。
注意：方括號兩邊不能空格，如：scanf("%[1234567890] ",strings); scanf("%[^1234567890

] ",strings); 不讓空格也會算在裡面的。
用這種方法還可以解決scanf的輸入中不能有空格的問題。只要用
scanf("%[^/n] ",strings); 就可以了。很神奇吧。

scanf原型:參見《C語言大全》和K&C
# include <stdio.h> ;
int scanf(const char *format, ...);
函式 scanf() 是從標準輸入流 stdin 中讀內容的通用子程式，可以讀入全部固有型別的資料並自動轉換成機內形式。

在 C99 中，format 用 restrict 修飾。
format 指向的控制串由以下三類字元組成:
● 格式說明符
● 空白符
● 非空白符

轉換字元(就是%後跟的部分）
a 讀浮點值(僅適用於 C99)
A 讀浮點值(僅適用於 C99)
c 讀單字元
d 讀十進位制整數
i 讀十進位制、八進位制、十六進位制整數
e 讀浮點數
E 讀浮點數
f 讀浮點數
F 讀浮點數(僅適用於 C99)
g 讀浮點數
G 讀浮點數
o 讀八進位制數
s 讀字串
x 讀十六進位制數
X 讀十六進位制數
p 讀指標值
n 至此已讀入值的等價字元數
u 讀無符號十進位制整數
[ ] 掃描字元集合
% 讀 % 符號(百分號)

例如： %s 表示讀串而 %d 表示讀整數。格式串的處理順序為從左到右，格式說明符逐一與變元表中的

變元匹配。為了讀取長整數，可以將 l(ell) 放在格式說明符的前面；為了讀取短整數，可以將 h 放在格式

說明符的前面。這些修飾符可以與 d、i、o、u 和 x 格式程式碼一起使用。

預設情況下，a、f、e 和 g 告訴 scanf() 為 float 分配資料。 如果將 l(ell) 放在這些修飾符的前

面，則 scanf() 為 double 分配資料。使用 L 就是告訴 scanf()，接收資料的變數是 long double 型變數。

如果使用的現代編譯器程式支援 1995 年增加的寬字元特性， 則可以與 c 格式程式碼一起，用 l 修飾符

說明型別 wchar_t 的寬字元指標；也可以與 s 格式程式碼一起，用 l 修飾符說明寬字串的指標。l 修飾符

也可以用於修飾掃描集，以說明寬字元。

控制串中的空白符使 scanf() 在輸入流中跳過一個或多個空白行。空白符可以是空格(space)、製表符

(tab)和新行符(newline)。 本質上，控制串中的空白符使 scanf() 在輸入流中讀，但不儲存結果，直到發

現非空白字元為止。

非空白符使 scanf() 在流中讀一個匹配的字元並忽略之。例如， "%d,%d " 使 scanf() 先讀入一個整數

，讀入中放棄逗號，然後讀另一個整數。如未發現匹配，scanf() 返回。

scanf() 中用於儲存讀入值的變元必須都是變數指標，即相應變數的地址。

在輸入流中，資料項必須由空格、製表符和新行符分割。逗號和分號等不是分隔符，比如以下程式碼:
scanf("%d %d ", &r, &c);
將接受輸入 10 20，但遇到 10,20 則失敗。

百分號(%)與格式符之間的星號(*)表示讀指定型別的資料但不儲存。因此，
scanf("%d %*c %d ", &x, &y);
對 10/20 的讀入操作中，10 放入變數 x，20 放入 y。

格式命令可以說明最大域寬。 在百分號(%)與格式碼之間的整數用於限制從對應域讀入的最大字元數。

例如，希望向 address 讀入不多於 20 個字元時，可以書寫成如下形式：
scanf("%20s ", address);

如果輸入流的內容多於 20 個字元，則下次 scanf() 從此次停止處開始讀入。 若達到最大域寬前已遇

到空白符，則對該域的讀立即停止；此時，scanf() 跳到下一個域。

雖然空格、製表符和新行符都用做域分割符號，但讀單字元操作中卻按一般字元處理。例如，對輸入流

"x y " 呼叫：
scanf("%c%c%c ", &a, &b, &c);
返回後，x 在變數 a 中，空格在變數 b 中，y 在變數 c 中。

注意，控制串中的其它字元，包括空格、製表符和新行符，都用於從輸入流中匹配並放棄字元，被匹配

的字元都放棄。例如，給定輸入流 "10t20 "，呼叫：
scanf("%dt%d ", &x, &y);
將把 10 和 20 分別放到 x 和 y 中，t 被放棄，因為 t 在控制串中。

ANSI C 標準向 scanf() 增加了一種新特性，稱為掃描集(scanset)。 掃描集定義一個字元集合，可由

scanf() 讀入其中允許的字元並賦給對應字元陣列。 掃描集合由一對方括號中的一串字元定義，左方括號前

必須綴以百分號。 例如，以下的掃描集使 scanf() 讀入字元 A、B 和 C：
%[ABC]

使用掃描集時，scanf() 連續吃進集合中的字元並放入對應的字元陣列，直到發現不在集合中的字元為

止(即掃描集僅讀匹配的字元)。返回時，陣列中放置以 null 結尾、由讀入字元組成的字串。

用字元 ^ 可以説明補集。把 ^ 字元放為掃描集的第一字元時，構成其它字元組成的命令的補集合，指

示 scanf() 只接受未説明的其它字元。
對於許多實現來説，用連字元可以説明一個範圍。 例如，以下掃描集使 scanf() 接受字母 A 到 Z:
%[A-Z]
重要的是要注意掃描集是區分大小寫的。因此，希望掃描大、小寫字元時，應該分別説明大、小寫字母

。
scanf() 返回等於成功賦值的域數的值，但由於星號修飾符而讀入未賦值的域不計算在內。給第一個域

賦值前已出錯時，返回 EOF。

C99 為 scanf() 增加了幾個格式修飾符：hh、ll、j、z 和 t。hh 修飾符可用於 d、i、o、u、x、X 或

n。它説明相應的變元是 signed 或 unsigned char 值，或用於 n 時， 相應的變元是指向 long char 型變

量的指標。ll 修飾符也可用於 d、i、o、u、x、X 或 n。它説明相應的變元是 signed

或者 unsigned long

long int 值。
j 格式修飾符應用於 d、i、o、u、x、X 或 n，説明匹配的變元是型別 intmax_t 或 uintmax_t。這些

型別在 <stdint.h> ; 中宣告，並説明最大寬度的整數。
z 格式修飾符應用於 d、i、o、u、x、X 或 n，説明匹配的變元是指向 size_t 型別物件的指標。該類

型在 <stddef.h> ; 中宣告，並説明 sizeof 的結構。
t 格式修飾符應用於 d、i、o、u、x、X 或 n，説明匹配的變元是指向 ptrdiff_t 型別物件的指標。

該型別在 <stddef.h> ; 中宣告，並説明兩個指標之間的差別。

例子:

```
# include <stdio.h> ;
int main( void )
{
char str[80], str2[80];
int i;
```

```
/* read a string and a integer */
scanf( "%s%d ", str, &i );
/* read up to 79 chars into str */
scanf( "%79s ", str );
/* skip the integer between the two strings */
scanf( "%s%*d%s ", str, str2 );
return 0;
```
}你的問題在這個部分：

3.scanf()函式的引數輸入型別不匹配問題

這是我在csdn論壇上見到的問題，這個錯誤有時候會讓人莫名其妙。

```
#include <stdio.h>
main()
{
int a=123;
char c= 't ';
printf( "input/n ");
scanf( "%d%c ",&a,&c);
scanf( "%d%c ",&a,&c);
scanf( "%d%c ",&a,&c);
printf( "%d/n%c/n ",a,c);
return 0;
}
```

當輸入a 回車 後，會直接跳過下面 2 個scanf語句，直接輸出為

123

t

原因：對於scanf("%d%c ",&a,&c)，scanf語句執行時，首先試圖從緩衝區中讀入一個%d型別的資料，如果和

第一個引數匹配，則繼續從緩衝區中讀取資料和第二個引數進行匹配，依次進行下去，直到匹配完所有的參

數；如果其中有一個引數不匹配，那就從這個地方跳出，忽略這個scanf後面所有的引數，而去執行下一條語

句。
可以用下面的程式驗證一下：

```
#include <stdio.h>
int main()
{
int a=123,b=1;
char c= 't ';
scanf( "%d%d ",&a,&b);
scanf( "%c ",&c);
printf( "%d/n%d/n%c/n ",a,b,c);
return 0;
}
```

輸入：2 回車a 回車
結果是：

```
2
1
a
```

解決方法：scanf()函式執行成功時的返回值是成功讀取的變數數,也就是說，你這個scanf()函式有幾個變數

，如果scanf()函式全部正常讀取，它就返回幾。但這裡還要注意另一個問題，如果

輸入了非法資料，鍵盤緩

衝區就可能還個有殘餘資訊問題。
比如：

```
#include <stdio.h>
main()
{
int a=123,b;
while(scanf( "%d%d ",&a,&b)!=2)
fflush(stdin);
printf( "%d/n%d/n ",a,b);
return 0;
}
```


標籤：

您可能也會喜歡...

C語言scanf函式用法詳細解釋！

C語言sscanf函式用法總結（一） 正則表示式

C語言--氣泡排序法（詳細註釋）

C語言scanf()函式的詭異事件

關於c語言scanf函式中格式化輸入中加入空格的一個問題

C語言scanf函式與printf函式

C語言sprintf函式的深入理解

C語言qsort函式用法

C語言：assert()函式用法總結

C語言scanf函式輸入時鍵盤緩衝區\n的問題[經典問題]

iOS開發——c語言——scanf函式詳細講解

C語言scanf()函式返回值的問題（實驗一）

C語言（scanf函式工作原理）

C語言scanf函式詳細解釋

c++ STL List查詢遍歷及各成員函式用法詳細介紹