

C 语言

一、Printf

`%[flags][width][.perc][F|N|h|l]type`

1、printf 的格式部分定义如下：`%[flags][width][.perc][F|N|h|l]type`。因此格式字符数以%开始的，然后是 flags(0 个或 1 个)，width(0 个或 1 个)，perc (以小数点开始的一个数字或没有)，F/N/h/l/ll 其中之一，最后是 type (d, x, f, s, p 等)。

2、flags 有 5 种：一是空格，表示用空格填充。二是 0，表示用 0 填充。三是减号，表示使用左对齐（默认是右对齐的）。四是加号，表示在数字前显示正负号。五是#，表示如果后面的 type 是 x 时打印的十六进制数前加 0x。（-与+或#可共用）

3、width 表示打印长度，不足的部分用空格或 0 填充。但是要注意一些默认处理，譬如如果同时设置了左对齐和零填充，那么 0 填充将自动转为空格填充。（因为如果在整数的后面填充 0 那打印值就错了）。

4、perc 部分只在浮点数时有效，表示浮点数小数点后精确到的最多位数。

5、F/N/h/l/ll 等，整数转换说明符，即对后面打印的整数进行强制转换后输出。譬如 F 表示远指针、N 表示近指针、h 表示将 int 强制截断为 short 输出，l 表示将整数强制按 long 型输出，ll 表示将整数强制按照 long long 输出。

- 1、
 - (1) %d (%i) 输出十进制 int 型
 - (2) %o 输出八进制 int 型
 - (3) %x (X) 输出十六进制 int 型
 - (4) %u 输出 unsigned int 型
 - (5) %f 输出 float 和 double 型（小数形式）
 - (6) %e (E) 输出 float 和 double 型（指数形式）
 - (7) %g 自动选择小数和指数形式输出 float 和 double 型
 - (8) %c 输入单个字符
 - (9) %s 字符串型
- 2、在 l 的前面添加
 - (1) h 输出 short 型
 - (2) l 输出 long 型
 - (3) ll 输出 long long 型

二、scanf

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。

- (1) d 输入十进制整数
- (2) o 输入八进制整数
- (3) x 输入十六进制整数
- (4) u 输入无符号十进制整数
- (5) f 或 e 输入实型数(用小数形式或指数形式)
- (6) c 输入单个字符
- (7) 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：

```
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：

```
scanf("%5d", &a);
```

输入 12345678 只把 12345 赋予变量 a，其余部分被截去。又如：

```
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。

注意：

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 scanf("%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：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' 赋予 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：scanf("%c %c %c", &a, &b, &c); 则输入时各数据之间可加空格。

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：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：scanf("a=%d, b=%d, c=%d", &a, &b, &c); 则输入应为：a=5, b=6, c=7。

C++

一、cout

1、使用流操作算子

C++ 中常用的输出流操纵算子如表 1 所示，它们都是在头文件 `iomanip` 中定义的；要使用这些流操纵算子，必须包含该头文件。

注意：“流操纵算子”一栏中的星号*不是算子的一部分，星号表示在没有使用任何算子的情况下，就等效于使用了该算子。例如，在默认情况下，整数是用十进制形式输出的，等效于使用了 `dec` 算子。

(1)

<code>*dec</code>	以十进制形式输出整数
<code>hex</code>	以十六进制形式输出整数
<code>oct</code>	以八进制形式输出整数
<code>setbase(b)</code>	设置输出整数时的进制，b=8、10 或 16
<code>uppercase</code>	十六进制数中使用 A~E。若输出前缀，则前缀输出 0X，科学计数法中输出 E
<code>*nouppercase</code>	十六进制数中使用 a~e。若输出前缀，则前缀输出 0x，科学计数法中输出 e。
<code>showbase</code>	输出表示数值的进制的前缀
<code>*noshowbase</code>	不输出表示数值的进制. 的前缀
<code>showpos</code>	在非负数值中显示 +
<code>*noshowpos</code>	在非负数值中不显示 +

(2)

<code>left</code>	左对齐，即在宽度不足时将填充字符添加到右边
<code>*right</code>	右对齐，即在宽度不足时将填充字符添加到左边
<code>setw(w)</code>	指定输出宽度为 w 个字符，或输入字符串时读入 w 个字符
<code>setfill(c)</code>	在指定输出宽度的情况下，输出的宽度不足时用字符 c 填充（默认情况是用空格填充）注意： <code>setfill(' ')</code>
<code>internal</code>	数值的符号（正负号）在指定宽度内左对齐，数值右对 齐，中间由填充字符填充。

(3)

fixed	以普通小数形式输出浮点数
scientific	以科学计数法形式输出浮点数
setprecision(n)	设置输出浮点数的精度为 n。 在使用非 fixed 且非 scientific 方式输出的情况下,n 即为有效数字位数,如果有效数字位数超过 n,则小数部分四舍五人,或自动变为科学计数法输出并保留一共 n 位有效数字。 在使用 fixed 方式和 scientific 方式输出的情况下, n 是小数点后面应保留的位数。
showpoint	不必要的时候也显示 10 进制的小数点以及其后的 0
*noshowpoint	只有当小数部分存在时才显示小数点

(4)

boolalpha	把 true 和 false 输出为字符串
*noboolalpha	把 true 和 false 输出为 0、1

2、setiosflags() 算子

setiosflags() 算子实际上是一个库函数,它以一些标志作为参数,这些标志可以是在 `iostream` 头文件中定义的以下几种取值,它们的含义和同名算子一样。

<code>ios::left</code>	输出数据在本域宽范围内向左对齐
<code>ios::right</code>	输出数据在本域宽范围内向右对齐
<code>ios::internal</code>	数值的符号位在域宽内左对齐,数值右对齐,中间由填充字符填充
<code>ios::dec</code>	设置整数的基数为 10
<code>ios::oct</code>	设置整数的基数为 8
<code>ios::hex</code>	设置整数的基数为 16
<code>ios::showbase</code>	强制输出整数的基数(八进制数以 0 开头,十六进制数以 0x 打头)
<code>ios::showpoint</code>	强制输出浮点数的小点和尾数 0
<code>ios::uppercase</code>	在以科学记数法格式 E 和以十六进制输出字母时以大写表示
<code>ios::showpos</code>	对正数显示 “+” 号
<code>ios::scientific</code>	浮点数以科学记数法格式输出
<code>ios::fixed</code>	浮点数以定点格式(小数形式)输出
<code>ios::unitbuf</code>	每次输出之后刷新所有的流
<code>ios::stdio</code>	每次输出之后清除 <code>stdout</code> , <code>stderr</code>

注意：

1、多个标志可以用|运算符连接，表示同时设置。例如：`cout << setiosflags(ios::scientific|ios::showpos) << 12.34;`输出结果是：`+1.234000e+01`

2、如果两个相互矛盾的标志同时被设置，如先设置 `setiosflags(ios::fixed)`，然后又设置 `setiosflags(ios::scientific)`，那么结果可能就是两个标志都不起作用。因此，在设置了某标志，又要设置其他与之矛盾的标志时，就应该用 `resetiosflags()` 清除原先的标志。

二、cin

*dec	以十进制形式输出整数
hex	以十六进制形式输出整数
oct	以八进制形式输出整数
setbase(b)	设置输出整数时的进制，b=8、10 或 16