



西北工业大学  
NORTHWESTERN POLYTECHNICAL UNIVERSITY

---

# C程序设计 Programming in C



**1011014**

---

主讲：姜学锋，计算机学院

## 编程实现简单数据的输出（1）

- ◆ 1、printf函数
- ◆ 2、输出格式控制

### 3.2.2 格式化输出

---

- ▶ 1. printf函数
- ▶ printf函数的作用是向标准输出设备（显示终端）输出格式化的数据，一般形式为：

```
printf(格式控制,输出项列表.....);
```

- ▶ 示例

```
printf("a=%d,b=%d\n",a,b);
```

### 3.2.2 格式化输出

---

- ▶ (1) 格式控制
- ▶ 格式控制为字符串形式，称为格式控制串。

```
printf(格式控制, 输出项列表.....);
```

### 3.2.2 格式化输出

---

- ▶ 格式控制串主要有两种内容：
- ▶ ①格式说明。格式说明总是以百分号（%）字符开始，后跟格式控制字符，例如%d、%f等。它的作用是将输出项转换为指定格式输出。
- ▶ ②一般字符。除格式说明之外的其他字符，包含转义字符。一般字符根据从左向右的出现顺序直接输出到显示终端上，ASCII控制字符的输出有特殊的含义。

### 3.2.2 格式化输出

---

- ▶ (2) 输出项列表
- ▶ 输出项列表为将要输出的数据，可以是常量、变量或表达式。输出项可以是零个或多个，但必须与格式说明一一对应，即一个格式说明决定一个输出项。

```
printf(格式控制,输出项列表.....);
```

### 3.2.2 格式化输出

---

- ▶ 可以没有输出项且无格式说明:
- ▶ 示例

```
printf("hello,world\n"); //没有输出项, 且无格式说明
```

### 3.2.2 格式化输出

---

- ▶ 2. 格式控制
- ▶ 格式控制串按照从左向右的顺序，第1个格式说明对应第1个输出项，第2个格式说明对应第2个输出项，以此类推。如果输出项多于格式说明，则多出的输出项被忽略。如果没有足够的输出项对应所有的格式说明，则输出结果无法预料。

```
printf("%d,%d,%d\n",a,b,c);  
//输出项与格式说明个数相等，正常情况  
printf("%d,%d\n",a,b,c);  
//输出项多于格式说明个数，多余输出项（c）忽略  
printf("%d,%d,%d\n",a,b);  
//输出项少于格式说明个数，第3个输出不可预料
```



### 3.2.2 格式化输出

---

举例：

```
1 int a=123,b=-1,c=12345;  
2  
3 printf("%d,%d\n",a,b,c); //格式数目小于输出项数，忽略多余输出项  
4 //输出结果：123,-1  
5 printf("%d,%d,%d\n",a,b); //格式数目大于输出项数，输出结果不确定  
6 //输出结果：123,-1,2367460
```

### 3.2.2 格式化输出

---

- ▶ (1) 格式说明域
- ▶ 格式说明由可选（方括号）及必需的域组成，其形式如下：

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

### 3.2.2 格式化输出

---

- ▶ 格式说明域是个表明具体格式选项的单个字符或数字。最简单的格式说明只有百分号和type字符（如%s）。可选域出现在type字符前，控制格式的其他特征。表3-1解释了每个域的含义，如果百分号后的字符作为格式说明域没有意义，则该字符直接输出。

### 3.2.2 格式化输出

表3-1 printf格式说明域含义

域	域选	描述	含义
type	必需	类型字符	决定输出项转换为字符、字符串还是数值
flags	可选	标志字符	控制输出的对齐、符号、空格及八进制和十六进制前缀。可以出现多个标志
width	可选	宽度说明	指定输出项的最小显示宽度
.prec	可选	精度说明	指定输出项的最大输出字符数或浮点数小数精度
h/l/L/F/N	可选	大小修饰	指明输出项类型大小或指针的远近

### 3.2.2 格式化输出

---

- ▶ (2) type类型字符
- ▶ 类型字符是printf函数唯一必需的格式说明域。它出现在任何可选域之后，用来确定输出项的类型。表3-2列出了常用类型字符的含义。

### 3.2.2 格式化输出

表3-2 printf类型字符含义

字符	类型	输出格式
d	int	带符号的十进制整数
u	int	无符号十进制整数
o	int	无符号八进制整数
x或X	int	无符号十六进制整数（若输出为字母，x用abcdef，X用ABCDEF）

### 3.2.2 格式化输出

---

举例：

```
1 int a=123,b=-1;
2
3 //①输出整型数据
4 printf("%d,%u,%x,%X,%o\n",a,a,a,a,a);
5 //十进制、无符号、十六进制和八进制
6 //输出结果：123,123,7b,7B,173
7 printf("%d,%u,%x,%X,%o\n",b,b,b,b,b);
8 //十进制、无符号、十六进制和八进制，负数为补码
9 //输出结果：-1,4294967295,ffffffff,FFFFFFFF,37777777777
```

### 3.2.2 格式化输出

表3-2 printf类型字符含义

字符	类型	输出格式
f	double	具有[－]dddd.dddd格式的带符号数值，dddd为一位或多位十进制数字。小数点前数字个数取决于数的量级；小数点后面数字个数取决于所要求的精度
e或E	double	具有[－]d.dddde[+/-]ddd格式的带符号数值，其中d为单个十进制数字，dddd为一位或多位十进制数字，ddd为三位十进制数。用e或E表示指数
g或G	double	以f或e格式输出的带符号数值，对给出的值及其精度，f和e哪个简洁就用哪个。只有当值的指数小于－4或大于、等于精度说明时才使用e格式。尾部的0被截断，只有小数点后跟1位或多位数字时才出现小数点。用e或E表示指数



### 3.2.2 格式化输出

---

举例：

```
1 double x=12.3456,y=12,z=12.123456789123;  
2  
3 //⑤输出浮点型数据  
4 printf("%lf,%e,%g\n",x,x,x); //小数格式、指数格式、最简格式  
5 //输出结果: 12.345600,1.234560e+001,12.3456  
6 printf("%lf,%e,%g\n",y,y,y); //小数格式、指数格式、最简格式  
7 //输出结果: 12.000000,1.200000e+001,12
```

### 3.2.2 格式化输出

---

举例：

```
1 int a=123;  
2 double x=12.3456;  
3  
4 printf("%d,%lf\n",x,a); //类型不对应，输出结果不确定  
5 //输出结果：2075328197,0.000000
```

### 3.2.2 格式化输出

续表3-2 printf类型字符含义

字符	类型	输出格式
c	char	单个字符
s	字符串指针	直到第一个非空字符（'\0'）或满足精度的字符串
%		输出百分号'%'

### 3.2.2 格式化输出

---

举例：

```
1 char c1=97;
2
3 //③输出字符型数据
4 printf("%d,%c\n",c1,c1); //字符型数值、ASCII码
5 //输出结果：97,a
6
7 //⑨特殊输出
8 printf("%%\n",c1); //两个%%表示输出一个%，输出项
9 //输出结果：%
```

**CP 程序设计**