

数据输入与输出

DATA TYPES AND EXPRESSIONS
2 数据类型与表达式



2.7 数据输入输出

数据是程序处理的对象；输入输出操作是程序中不可缺少的部分，C中输入输出操作都是通过函数调用实现的；C提供了一个“标准I/O库”。

2.7.1 格式输出函数——printf

基本形式是:

```
printf( "控制字符串" ,输出项列表);
```

功能：

按规定的输出格式,把输出项依次输出到系统指定的缺省输出设备,通常为显示器。

输出项可以是变量、常量、或表达式。其间用逗号分隔。



控制字符串包括格式控制说明符和普通字符两个部分。

1. 格式控制说明(表2.7.1)

%

字符

%c

按字符型输出

%d

按十进制整数输出

%u

按无符号整数输出

%f

按浮点型小数输出

%o

按八进制整数输出

%x

按十六进制整数输出

%s

按字符串输出

使用修饰符,可以控制输出的宽度、精度、小数位数、对齐方式等。

例如：

(1) 字段宽度

%6d 表示以字段宽度为6个字节显示一个整数。

设 $i=123$; `printf(" %6d\n" ,i);` 输出: `__123`

`printf(" %06d\n" ,i);` 输出: `000123`

设 $x=1234.567$

`printf(" %9.3f\n" ,x);` 输出: `_1234.567`

(2) 负号

不用负号为右对齐,用负号为左对齐

`printf(" %-9.3f\n" ,x);` 输出: `1234.567_`

(3) 字符*

`printf(" %*d\n" ,5,i);` 输出: `__123`

`printf(" %*d\n" ,w,i);` 输出: 以w宽度显示l

`printf(" %*.2f\n" ,7,2,x);` 输出: `1234.56`

2. 普通字符:原样输出的字符

可打印字符__直接写出字符

不可打印字符__使用转义字符\n, \t, \a

例如: m=3; n=12;

```
printf(" num1=%d \t num2=%d\n" ,m,n);
```

显示:

```
num1=3__ num2=12
```

例如 :

```
printf(" %f%%\n" ,95.6);
```

显示:

```
95.6%
```

2.7.2 字符输出函数——putchar

功能:

将一个字符输出在显示屏上,显示参数可以是字符变量、字符常量、字符ASCII码。

`putchar('A');` 显示大写字母A

`putchar(97);` 显示小写字母a

如果: `c= '!' ;`

`putchar(c);` 显示字符!

输入输出函数被包含在库函数“`stdio.h`”中,使用这些函数时,要使用预处理命令。

`#include "stdio.h"` 但 `printf` 和 `scanf` 可缺省

2.7.3 格式输入函数——scanf

功能：从键盘输入数据，按指定的输入格式把数据赋给相应的输入项。

格式：scanf(" 控制字符串 ",输入项列表);

注意：接收数据的变量必须用其地址表示,控制字符串与输出的表示方法相似，注意不同点。

1. 字符*

输入时,用于跳过其相应的数据.

```
scanf(" %d%d" , &i,&j);
```

输入 34 45 时, i为34 j为45

```
scanf(" %*d%d" , &i,&j);
```

输入 34 45 时, i为45

2. 空格字符

功能: 输入字符型数据时,跳过输入流中所有空格字符.

```
scanf(" %d%d%c" ,&i,&j,&ch);
```

输入: 45 20 a 时, i为45, j为20, ch为空格

```
scanf(" %d%d %c" ,&i,&j,&ch);
```

输入: 45 20 a 时, i为45, j为20, ch为a

3. 可打印字符

功能: 去除输入数据流中与可打印字符相同的字符。

```
scanf(" %d,%d,%c" ,&i,&j,&ch);
```

输入应为: 100,200,b (表示以逗号为分隔符)

结果: i为100, j为200, ch为b

2.7.4 字符输入函数——getchar

格式：getchar() 无参数, 只能接收一个字符。

功能：从键盘上输入一个可打印字符, 函数的返回值为该字符的ASCII码。

使用时, 通常将其返回值赋于某个字符变量或整型变量。

例如:

```
#include "stdio.h"
main( )
{
    char c;
    c=getchar( );    putchar(c);
    c=007;  putchar(c );  putchar(007);
}
```

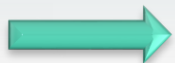


程序举例

例：输入一个字符，输出其前导字符，字符和后续字符。

```
#include "stdio.h"
main( )
{
    char c;    int c1, c2;
    c=getchar( );
    c1=c-1;    c2=c+1;
    printf(" %c %c %c\n" ,c1,c,c2);
    printf(" %d %d %d\n" ,c1,c,c2);
}
```

输入： B



输出： A B C
65 66 67



printf函数的格式控制符号没有二进制输出形式！

我看整数可以以10进制，
8进制，16进制形式输出，
二进制呢？





可我现在遇到一个编程问题，需要按照二进制形式输出一个十进制，怎么办？



程序举例

例：输入一个0 → 255间的十进制整数,分别以十六进制数、八进制数、二进制数形式输出该数。

```
main( )
{
    unsigned char i,j;
    printf(" Please input a number less than 256\n" );
    scanf(" %d" ,&i);
    printf(" The hexadecimal express of i is 0x%x\n" ,i);
    printf(" The octal express of i is %04o\n" ,i);
    printf(" The binary express of i is " );
```




比如，我们把十进制17转换为2进制，怎么转换？

除2取余法！

$$17/2=8...1$$

$$8/2=4...0$$

$$4/2=2...0$$

$$2/2=1...0$$

$$1/2=0...1$$



二进制是10001

0-255 不超过8位
移位操作代替除2运算
先输出最高位
最后输出最低位

$17 \gg 7 = 0$

$17 \gg 6 = 0$

$17 \gg 5 = 0$

$17 \gg 4 = 1$

$17 \gg 3 = 0$

$17 \gg 2 = 0$

$17 \gg 1 = 0$

0-255 不超过8位
移位操作代替除2运算
先输出最高位
最后输出最低位



$17 >> 7 = 0$

$17 >> 6 = 0$

$17 >> 5 = 0$

$17 >> 4 = 1$

$17 >> 3 = 2$

$17 >> 2 = 4$

$17 >> 1 = 8$

0-255 不超过8位
移位操作代替除2运算
先输出最高位
最后输出最低位

$17 \gg 7 = 0$

$17 \gg 6 = 0$

$17 \gg 5 = 0$

$17 \gg 4 = 1$

$17 \gg 3 = 10$

$17 \gg 2 = 100$

$17 \gg 1 = 1000$

实际上



$17 \gg 7 = 0$

$(17 \gg 6) \& 0x01 = 0$

$(17 \gg 5) \& 0x01 = 0$

$(17 \gg 4) \& 0x01 = 1$

$(17 \gg 3) \& 0x01 = 0$

$(17 \gg 2) \& 0x01 = 0$

$(17 \gg 1) \& 0x01 = 0$

$17 \& 0x01 = 1$



$17 \gg 7 = 0$

$17 \gg 6 = 0$

$17 \gg 5 = 0$

$17 \gg 4 = 1$

$17 \gg 3 = 10$

$17 \gg 2 = 100$

$17 \gg 1 = 1000$

注意！
0的ascii码

$17 \gg 7 = 0$

$(17 \gg 6) \& 0x01 = 0$

$(17 \gg 5) \& 0x01 = 0$

$(17 \gg 4) \& 0x01 = 1$

$(17 \gg 3) \& 0x01 = 0$

$(17 \gg 2) \& 0x01 = 0$

$(17 \gg 1) \& 0x01 = 0$

$17 \& 0x01 = 1$



```
j=(i>>7)+48;
printf(" %1c" ,j);
j=((i>>6)&0x01)+48;
printf(" %1c" ,j);
j=((i>>5)&0x01)+48;
printf(" %1c" ,j);
j=((i>>4)&0x01)+48;
printf(" %1c" ,j);
j=((i>>3)&0x01)+48;
printf(" %1c" ,j);
j=((i>>2)&0x01)+48;
printf(" %1c" ,j);
j=((i>>1)&0x01)+48;
printf(" %1c" ,j);
j=((i&0x01)+48;
printf(" %1c\n" ,j);
}
```

运行结果是:

Please input a number less than 256

195

The hexadecimal express of i is 0xc3

The octal express of i is 0303

The binary express of i is 11000011