

8.5 位运算符_物联网 / 嵌入式工程师 - 慕课网

“ 慕课网慕课教程 8.5 位运算符涵盖海量编程基础技术教程，以图文图表的形式，把晦涩难懂的编程专业用语，以通俗易懂的方式呈现给用户。

\5. 位运算符

运算符	格式	功能	结果
&	表达式 1 & 表达式 2	按位与	1 按位与 1 结果是 1 1 按位与 0 结果是 0 0 按位与 0 结果是 0 速记：都 1 则 1,1 零则 0
	表达式 1 表达式 2	按位或	1 按位或 0 结果是 1 1 按位或 1 结果是 1 0 按位或 0 结果是 0 速记：一 1 则 1, 都 0 则 0
^	表达式 1 ^ 表达式 2	按位异或	相同为 0，不同为 1
~	~ 表达式	按位取反	1 变成 0, 0 变成 1
<<	表达式 << 位数	左移	无论是正数还是负数, 左移都是对应的二进制位向左移动高位丢弃, 低位补 0 左移的时候, 等价于将这个数据乘以 2 的 n(移动的位) 次方
>>	表达式 >> 位数	右移	逻辑右移 (针对无符号数据) 对应的是正数和 0, 逻辑右移是对应的二进制位向右移动, 低位丢弃, 高位补 0。 算术右移 (针对有符号数据): 若是数据为正数, 算术右移是对应的二进制位向右移动, 低位丢弃, 高位补 0。 若是数据为负数, 算术右移是对应的二进制位向右移动, 低位丢弃, 高位补 1。 速记： 右移, 向右移动, 低位丢弃, 高位正数补 0, 负数补 1

```
#include <stdio.h>

int main()
{
    unsigned char reg1 = 0x13;

    unsigned char reg2 = 0x78;
    unsigned char data = 0;

    data = reg1 & reg2;
    printf("data = %#x\n",data);
}
```

```
data = reg1 | reg2;
printf("data = %#x\n",data);

data = ~reg1;
printf("data = %#x\n",data);

data = reg1 ^ reg2;
printf("data = %#x\n",data);

return 0;
}
```

```
data = 0x10
data = 0x7b
data = 0xec
data = 0x6b
```

```
#include <stdio.h>

int main()
{
    unsigned char  a = 10,b = 15;

    unsigned char  t = 0;

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

    a = a ^ b;
    b = a ^ b;
    a = a ^ b;

    printf("a = %d b = %d\n",a,b);
    return 0;
}
```

```
#include <stdio.h>

int main()
{
    unsigned char reg = 3;
    char a = -3;

    reg = reg << 3;
    printf("reg = %d\n",reg);
```

```
a = a << 3;
printf("a = %d\n",a);
```

```
a = a >> 6;
printf("a = %d\n",a);

return 0;
}
```

```
reg = 24  
a = -24  
a = 3
```

```
unsigned char a = 0x58;  
unsigned char b = 0xd7;  
unsigned char c = 0;
```

要求输出下列c的值。

c = a & b,

c = a | b.

c = (a & ~b) ^ 0xf1;

要求先笔算，然后书写代码验证。

全文完

本文由 简悦 SimpRead 优化，用以提升阅读体验

使用了 全新的简悦词法分析引擎 beta，点击查看详细说明

