**[c语言的按位运算符](http://www.cnblogs.com/Kazaf/archive/2012/03/19/2406006.html)**

& 按位与   
| 按位或   
^ 按位异或   
1. 按位与运算 按位与运算符"&"是双目运算符。其功能是参与运算的两数各对应的二进位相与。只有对应的两个二进位均为1时，结果位才为1 ，否则为0。参与运算的数以补码方式出现。   
例如：9&5可写算式如下： 00001001 (9的二进制补码)&00000101 (5的二进制补码) 00000001 (1的二进制补码)可见9&5=1。   
按位与运算通常用来对某些位清0或保留某些位。例如把a 的高八位清 0 ， 保留低八位， 可作 a&255 运算 ( 255 的二进制数为0000000011111111)。   
main(){   
int a=9,b=5,c;   
c=a&b;   
printf("a=%d/nb=%d/nc=%d/n",a,b,c);   
}   
2. 按位或运算 按位或运算符“|”是双目运算符。其功能是参与运算的两数各对应的二进位相或。只要对应的二个二进位有一个为1时，结果位就为1。参与运算的两个数均以补码出现。   
例如：9|5可写算式如下： 00001001|00000101   
00001101 (十进制为13)可见9|5=13   
main(){   
int a=9,b=5,c;   
c=a|b;   
printf("a=%d/nb=%d/nc=%d/n",a,b,c);   
}

3. 按位异或运算 按位异或运算符“^”是双目运算符。其功能是参与运算的两数各对应的二进位相异或，当两对应的二进位相异时，结果为1。参与运算数仍以补码出现，例如9^5可写成算式如下： 00001001^00000101 00001100 (十进制为12)   
main(){   
int a=9;   
a=a^15;   
printf("a=%d/n",a);   
}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

4．按位异或运算符^   
　　参与运算的两个值，如果两个相应位相同，则结果为0，否则为1。即：  
　　　　　　　　　0^0=0， 1^0=1， 0^1=1， 1^1=0  
　　例如： 　　　　10100001^00010001=10110000

                                    0^0=0,0^1=1  0异或任何数＝任何数

                                    1^0=1,1^1=0  1异或任何数－任何数取反

                                                              任何数异或自己＝把自己置0  
　　(1)按位异或可以用来使某些特定的位翻转，如对数10100001的第2位和第3位翻转，可以将数与00000110进行按位异或运算。  
　　　　　　　　　 10100001^00000110=10100111 //1010 0001 ^ 0x06 = 1010 0001 ^ 6  
　　(2)通过按位异或运算，可以实现两个值的交换，而不必使用临时变量。例如交换两个整数a，b的值，可通过下列语句实现：  
　　　　a=10100001,b=00000110  
　　　　a=a^b； 　　//a=10100111  
　　　　b=b^a； 　　//b=10100001  
　　　　a=a^b； 　　//a=00000110  
   （3），在汇编语言中经常是变量置零：

     条件运算符(? : )是一个三目运算符，即有三个操作数。

     使用条件运算符（?:）可以实现 If else 的功能，其一般形式为：

                    表达式1 ? 表达式2 : 表达式3

     其意思是：如果表达式1为真，则运算表达式2，整个条件表达式的值为 表达式2的值；反之，则运算表达式3，整个条件表达式的值为表达式3的值。