7.5-位运算符

1.按位与&

对两位操作数的二进制形式的每一位进行逻辑与运算。

2.按位或|

对两位操作数的二进制形式的每一位进行逻辑或运算。

3.按位异或^

对两位操作数的二进制形式的每一位进行逻辑异或运算。

异或运算法则：对应位相同运算结果为0，不同为1

0^0=0 0^1=1 1^0=1 1^1=0

4.按位取反~

对一个数的二进制形式的每一位求反，即0变为1，1变为0

这里在编译器中得不到我们想象的结果，这和计算机的存储单元的位数以及有符号数、无符号数有关，在计算机组成原理课程中会详细说明，这里大家可以不用管它

5.移位运算

左移（<<）和右移（>>），操作数为二进制数，a >> 1

左移：将最高位舍弃，其余所有位向左移一位，最低位补0；

右移：将最低位舍弃，其余所有位向右移一位，最高位补0；

代码示例：

int a=3,b=5;

cout<<(a&b)<<endl;

/\*

a=011

b=101

1&&0=0，1&&1=1，结果为001(2)=1(10)

\*/

cout<<(a|b)<<endl;

/\*

a=011

b=101

1||0=1，1||1=1，结果为111(2)=7(10)

\*/

cout<<(a^b)<<endl;

/\*

a=011

b=101

1^0=1，1^1=0，结果为110(2)=6(10)

\*/

cout<<(b>>1)<<endl;

/\*

b=101，右移1位，最低位的1被丢弃，其他所有位右移，最高位补0，则结果是10（2）=2（10）

\*/

cout<<(a<<2)<<endl;

/\*

a=011，左移2位，结果是1100（2）=12（10）

实际上，左移1位相当于十进制数\*2

\*/