

位运算只能对整型使用，如 char, long long，不可以对诸如 float 使用

异或的性质：

- 交换律、结合律、消去律，有单位元0，自己与自己运算得单位元

注意要点：

- 优先级：~; +, -, <<, >>; ==, !=; &; ^; |; &&; ||; ?:  
一个分号代表一个优先级，逗号间是同级
- 移位结果为11时应该是 1LL << k
- 右移位，等同于round(x/2.0)，负数的移位结果不会大于-1

常见应用：

- 取正数 x 的从左往右(从零数)第 i 位：(x>>i)&1
- 对某个正数 x 从左往右(从零数)第 k 位修改取反：x^=(1<<k)
- c&15 或 c^'0' 优化 数字字符转数值

char c;  
相当于c-'0'

## 内建函数

- 注：对 unsigned long long 每个函数名后面加上 11

### 1. \_\_builtin\_popcount(unsigned int n)

该函数时判断n的二进制中有多少个1

```
1 int n = 15; //二进制为1111
2 cout<<__builtin_popcount(n)<<endl; //输出4
```

### 2. \_\_builtin\_parity(unsigned int n)

该函数是判断n的二进制中1的个数的奇偶性

```
1 int n = 15; //二进制为1111
2 int m = 7; //111
3 cout<<__builtin_parity(n)<<endl; //偶数个，输出0
4 cout<<__builtin_parity(m)<<endl; //奇数个，输出1
```

### 3. \_\_builtin\_ffs(unsigned int n)

该函数判断n的二进制末尾最后一个1的位置，从一开始

```
1 int n = 1; //1
2 int m = 8; //1000
3 cout<<__builtin_ffs(n)<<endl; //输出1
4 cout<<__builtin_ffs(m)<<endl; //输出4
```

### 4. \_\_builtin\_ctz(unsigned int n)

该函数判断n的二进制末尾后面0的个数，当n为0时，和n的类型有关

```
1  int n = 1; //1
2  int m = 8; //1000
3  cout<<__builtin_ctzll(n)<<endl; //输出0
4  cout<<__builtin_ctz(m)<<endl; //输出3
```

#### 5. \_\_builtin\_clz (unsigned int x)

返回前导的0的个数。

```
1  int n = 1;  //1
2  int m = 8;  //1000
3  cout<< 32 - __builtin_clz(n) <<endl;    //输出1
4  cout<< 64 - __builtin_clzll(m) <<endl;  //输出4
```