

9.27

# 主要内容

- 关于整数
  - 运算符优先级
  - 类型自动转换
  - 大/小端法
- 关于浮点数
  - 0, NaN与Inf
  - 浮点数的表示方法

# 几个问题

`char c = 1; short s = 2; // sizeof(c + s) = ?`

输出 `5 > 1 << 3 - 1 & 3` 是多少?

输出 `2147483647 * 2` 是多少?

输出 `1ull + 2147483647 * 2` 是多少

`X >> 2` 和 `X/4` 有什么区别?

# 整数——运算符优先级

- 需要记住的
  - 单目运算符 > 四则运算 > 移位操作 > 关系比较 > 与 > 异或 > 或
  - 关系比较中，大小比较 > 等于不等于
  - 与 > 异或 > 或
- 随便的例子
  - `int b = 1 << 2 + 3;          // b == 7 ???`
  - `int a = 5 > 7 & 2;`
- 自己写程序的时候，记得加括号！

# 整数——自动类型转换

- 基本法：“小转大”“有符号转无符号”
- 一些默认情况：
  - 默认字面常量是int，如果超过了int的范围则是int64
    - sizeof( -2147483648)是多少？
  - 短的类型加减的时候自动变成int
- 表达式中的自动转换：
  - 并不是简单的都转成最长的数计算，而是“有需要的时候再转”
  - `long long a = 2147483647 * 2 + 1ull;    // a == ?`
  - `long long b = 2147483647 * 2ull + 1;    // a == b ?`
  - 同时，如果两个操作数中有一个是无符号数，则都转换成无符号数计算，如果都是有符号数，则就在有符号数的范畴下计算。
    - 比较大小的题目会出这个

# 整数——大/小端

- 稍微记忆一下ppt上哪些机器是大端， 哪些是小端
  - 重点记住网络数据传输是大端法， 平时我们用的基本上都是小端法
- Packeted Struct 和数据存放的关系

```
struct A
```

```
{
```

```
    int first : 4;
```

```
    unsigned second : 4;
```

```
}__attribute__((packed));
```

# 整数——一些乱七八糟的注意点

- $\text{TMIN} = -\text{TMAX} - 1$
- $-\text{TMIN} = \text{TMIN}$  （由上式也可以推出来）
- 关于true的取值：貌似有一些平台会转成-1
- 移位操作和除法的区别：
  - 移位是下取整
  - 除法是向零取整

## 小练习

### Initialization

```
int x = foo();  
int y = bar();  
unsigned ux = x;  
unsigned uy = y;
```

- $x < 0$   $((x*2) < 0)$
- $ux \geq 0$
- $x \& 7 == 7$   $(x < 30) < 0$
- $ux > -1$
- $x > y$   $-x < -y$
- $x * x \geq 0$
- $x > 0 \&\& y > 0$   $x + y > 0$
- $x \geq 0$   $-x \leq 0$
- $x \leq 0$   $-x \geq 0$
- $(x | -x) >> 31 == -1$
- $ux >> 3 == ux/8$
- $x >> 3 == x/8$
- $x \& (x-1) != 0$



# 浮点数——NaN与inf

- 表示方式
- 运算规则
  - NaN和任何数进行数学运算都是NaN，和任何数进行逻辑运算都是false
  - Inf基本遵循数学上的无穷大表示 如 $1/\text{inf}$ 是0， $\text{atan}(\text{inf})$ 是 $\pi/2$ 。逻辑运算也是遵循数学意义即最大/最小数。

# 浮点数——浮点数的表示方法

- IEEE的float和double格式需要记住
- 自定义的浮点数构造
  - Normalized 和 Denormalized的区别和计算
    - 为什么要有denormalized这种设计?
  - 通过二进制写出对应浮点数的值
    - $\text{Value} = (-1)^s \times M \times 2^E$
  - 给定一个分数，写出对应浮点数的二进制（不会循环）——涉及到舍入
    - 四舍六入五成双
- \*\* 大家回去可以自己通过写程序模拟任意结构的浮点数的构造来加深自己对这一块的理解