# 信息的表示和处理

## 信息存储：

1. 浮点运算由于表达的精度有限，因此是不符合结合律的运算。
2. 每个程序对象可以简单的视为一个字节块，那么程序本身就是一个字节序列。
3. 一个字节的值域：二进制（00000000～11111111）、十进制（0～255）、十六进制（00～FF）。
4. 将2的n次幂转为十六进制：将n转换为i+4j
5. 保证程序的可移植性即确保程序对不同数据类型的确切大小不敏感。
6. 在大部分机器上，多字节对象都被存储为连续的字节序列，对象的地址为所使用字节中的最小地址。
7. 小端：低地址存储低有效字节（例如数字的右端）；大端：低地址存储高有效字节（例如数字的左端）。
8. 字符串的字节顺序不受字大小或大小端的影响，因此文本数据比二进制数据具有更强的平台独立性。（使用man ascii命令获取一张ascii表）
9. 二进制代码很少能在不同机器和操作系统的组合间移植，因此二进制代码是不兼容的。
10. 布尔运算：满足&对|的分配律、|对&的分配律，由运算^、&、~构成布尔环，其中加法运算^的加法逆元是它自己本身（即a^a=0）。
11. 使用位向量来表示有限集合：向量a=[01101001]，表示集合A={0,3,5,6}，向量b=[01010101]，表示集合B={0,2,4,6}，此时|对应集合的并，&对应集合的交，～对应集合的补集。
12. C语言的逻辑运算符具有短路性质。
13. 移位运算符：左移、右移（算数右移补齐最高有效位的值、逻辑右移补零），几乎所有编译器/机器组合对有符号数据使用算数右移。
14. 移位操作的数量应该保持小于字长。

## 整数表示：

1. 二进制数映射到无符号数（双射）：
2. 补码，将字的最高有效位解释为负权（符号位），二进制数映射到有符号补码数（双射）：
3. 三个值分别表示无符号数最大值，有符号数的最小值和最大值，其中、
4. 一个负数的补码与其对应正数的反码+1的二进制表示一样。
5. C语言的强制类型转换保持位值不变，只是改变了解释这些位的方式。
6. 有符号数转换成无符号数：进一步推得
7. 同理无符号数转换成有符号数：
8. C语言在处理一个运算数是有符号的而另一个是无符号的时，会将有符号数强制类型转换为无符号数，并且假设这两个数都是非负的来执行运算。
9. 无符号数采用零扩展来扩展数字的位表示，而有符号数则采用符号扩展。
10. 当类型转换和扩展同时发生时，先进行扩展再进行类型转换。（例如short变为unsigned，先将16位进行符号扩展到32位再将其解释为unsigned）
11. 将一个w位的数截断为一个k位的数字时，我们会丢弃高w-k位，截断一个数字可能会改变它的值。补码数字的截断结果为：将截断后的k位无符号数转换为k位有符号数。

## 整数运算：

1. 执行两个无符号数值加法时：
2. 判断无符号数是否溢出：令，当且仅当（或者等价地）
3. 模数加法形成的阿贝尔群，其中每个元素都有一个加法逆元
4. 补码加法分为三种情况：情况一，负溢出，即原来两数相加值的区间为，得到的结果为一个正数；情况二，正常，即两数相加值的区间为，得到的结果为；情况三，正溢出，两数相加值的区间为，得到的结果为一个负数。总结如下：
5. 正溢出和负溢出的判断，当都是负数且时发生了负溢出，当都是正数且。
6. 补码的非运算对于范围内的：
7. 快速求一个补码的非，方法1可以将该补码的位级表示求反再+1，方法2可以对其位级表示的最右边一个1不变（只要不是全为0），其他位求反。
8. 有符号和无符号加法、减法和乘法运算的位级表示的结果是相同的。
9. 使用加减和移位运算的组合来替代常数因子的乘法：考虑常数K的二进制表示[(0…0)(1…1)(0…0)…(1…1)]，考虑一组从位位置n到位位置m的连续的1，可以用以下两种形式来计算这些位对乘机的影响：或者
10. 一个数除以2的幂，非负数使用逻辑右移实现；而负数的算术右移则会导致结果向下舍入，因此在有舍入发生时，将一个负数右移k位不等于整除的结果：表达式等价于算术右移的整除结果。

## 浮点数：

1. 二进制小数的表示形式：
2. IEEE浮点标准用的形式来表示一个数：

符号决定这个数是负数还是正数，而对于数值0的符号位解释作为特殊情况处理。

尾数是一个二进制小数，它的范围是，或者是。

阶码的作用是对浮点数加权，这个权重是2的E次幂（可能是负数）。

1. 阶码E的三种情况：

1规格化的值，当位模式不全为0或1时，阶码的值是，其中e是阶码二进制表示的无符号数，偏置值。此时，尾数M定义为，其中f的二进制表示为。

2非规格化的值，当阶码全为0时，。此时尾数。非规格化的数作用：一是提供了一种表示数值0的方法，二是表示那些非常接近于0.0的数。

3特殊值，当阶码全为1时，小数域全为0时，符号位为0表示，符号位为1表示。当小数域为非0时，结果值被称为NaN表示不是一个数。

1. 单精度下k=8，n=23，双精度下k=11，n=52