1.解释无符号整数和补码整数的表示方法，并比较它们的差异。

无符号整数就是使用二进制数进行表示，从低位到高位有不同的权重，分别从2^0到2^n-1,其中n为表示使用的位数。

补码整数使用最高位作为符号位，1表示负数，0表示正数，对于正数补码就是原码，负数补码是符号位不同，其他位取反+1再截断到原位数

差异：补码正负数都可以表示，补码可以认为是最高位是负权的无符号整数编码方式。

1. 如何使用位运算实现乘法和除法操作？

位运算实现乘除法主要是对特定情况使用，即乘除以2^n（n为整数），因为左移相当于原来每位的权重\*2，右移相当于原来每位的权重\*2^-1，所以在结果不会超出表示范围的情况下，就相当于实现了整体乘除2的效果，而乘除2的幂数就相当于移动的位数（除法会相当于进行了舍入操作）

1. 什么是浮点数表示法？解释规范化和非规范化浮点数的概念。

浮点数表示是使用一位表示符号位，若干位表示阶码，若干位表示小数部分的表示形式，如32位表示情况下阶码为8位。其中小数因为人为要求是大于1小于2的小数，所以整数位必定是1，不需要表示，所以（1.小数部分）才是乘以2的（阶码-bias）次方的对象，其中bias可以理解为使得阶码范围从0~2^n-1转换到了-2^n-2~2^n-2。

（这里不确定规范化和非规范化浮点数指的是什么，给出了两种答案）

规格化浮点数是阶码不全为0或1的浮点数，正常使用上方给出计算公式计算。

非规格化浮点数是阶码全为0的情况，此时计算公式改变为（0.小数部分）\*（1-bias）

规范化是IEEE浮点标准，即我回答浮点数表示法时所述的表示方式，好处是使用阶码，可以表示很大的数

非规范化是类似整数表示，使用整数部分加小数点加小数部分的表示方式，