我们常用的进制包括：二进制，八进制，十进制与十六进制。

### 1.二进制与十进制之间的转换。

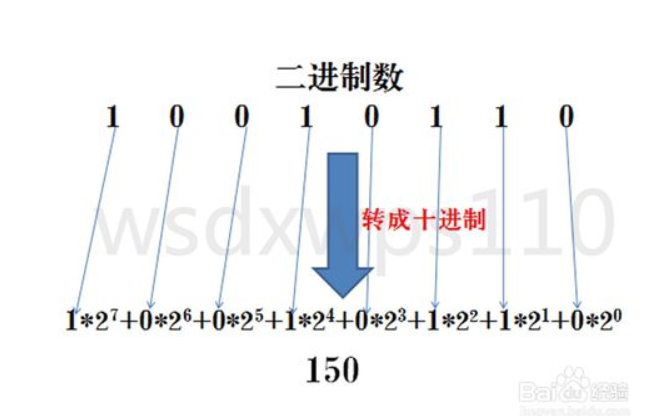
十进制转二进制：

方法为：十进制数除2取余法，即十进制数除2，余数为权位上的数，得到的商值继续除2，以此步骤继续向下运算知道商为0为止。最后得到的玉树，从最底读取到最顶即为二进制数。



二进制转十进制：

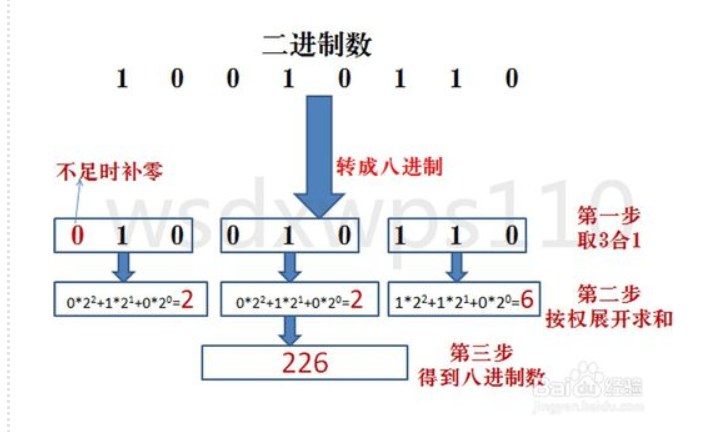
方法为：把二进制数按权展开，相加即得十进制数。



### 2.二进制与八进制的转换

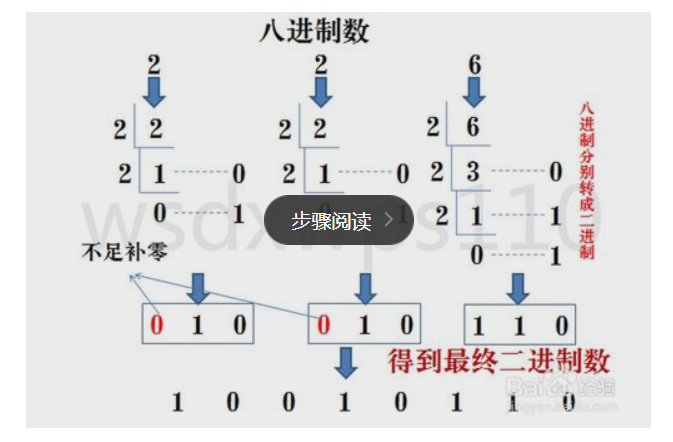
二进制转八进制：

方法为：3位二进制数按权展开相加得到1位八进制数(注意事项，3位二进制转成八进制是从右到左开始转换，不足时补0)



八进制转成二进制

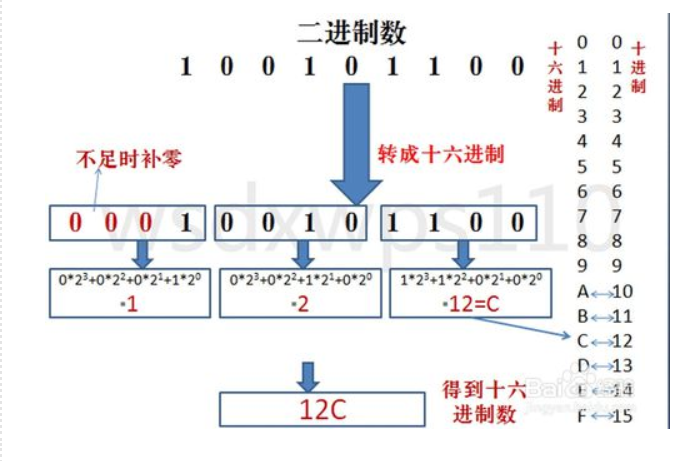
方法为：八进制数用过除2取余法，得到二进制数，对每个八进制为3个二进制，不足时在最左边补零。



### 3.二进制与十六进制之间的转换

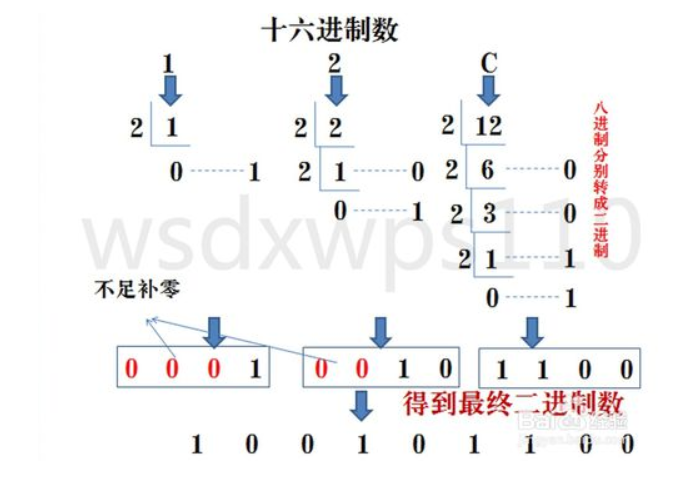
二进制转十六进制

方法为:与二进制转八进制方法近似，八进制是取三合一，十六进制是取四合一。(注意：4位二进制转成十六进制是从右到左开始转换，不足时补0)



十六进制转二进制

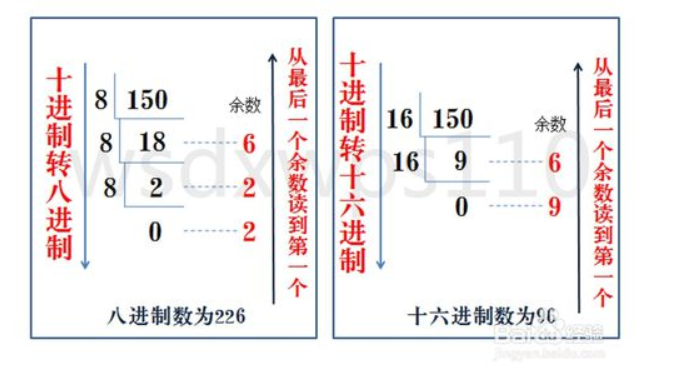
方法为：十六进制数通过除2取余法，得到二进制数，对每个十六进制为4个二进制，不足时在最左边补零



### 4.十进制与八进制与十六进制之间的转换

方法一：间接法—把十进制转成二进制，然后再由二进制转成八进制或者十六进制。这里不再做图片用法解释 。

方法二：



### 5.其他互转，可各种使用间接法

### 6.位运算

程序中所有的数在计算机内存中都是以二进制的形式存储的。位运算就是直接对整数在内存中的二进制位进行操作。比如，and运算本来是一个逻辑运算符，但整数与证书之间也可以进行and运算。举个例子，6的二进制是110，11的二进制是1011，那么6 and 11的结果就是2. 110 and 1011 =>0010 =>2



只看Java

&:（按位与）：通常用于二进制的取[位操作](https://baike.baidu.com/item/%E4%BD%8D%E6%93%8D%E4%BD%9C" \t "/tmp/wps-root/x/_blank)，例如一个数 and 1的结果就是取[二进制](https://baike.baidu.com/item/%E4%BA%8C%E8%BF%9B%E5%88%B6" \t "/tmp/wps-root/x/_blank)的最末位。这可以用来判断一个整数的奇偶，二进制的最末位为0表示该数为[偶数](https://baike.baidu.com/item/%E5%81%B6%E6%95%B0" \t "/tmp/wps-root/x/_blank)，最末位为1表示该数为奇数。 00101 & 11100 = 00100 （相同位的两个数字都为1，则为1；若有一个不为1，则为0.）

|：（按位或）：通常用于二进制特定位上的无条件赋值，例如一个数 | 1 的结果就是把二进制最末尾强行变成1.如果需要把二进制最末尾变成0，对这个数 | 1 之后再减一就可以了，其实际意义就是把这个数强行变成最接近的偶数。00101 | 11100 = 11101 (相同位只要有一个为1即为1.)。

^:（按位异或）：异或的逆运算是他本身，也就是说两次异或同一个数最后结果不变，可以用于简单的加密。00101 ^ 11100 = 11001（相同位不同则为1，相同则为0.）

~:（按位取反)：把内存中的0和1全部取反。使用此运算时需格外小心，需要注意整数类型有没有符号。如果~的对象时无符号整数(不能表述负数)，那么得到的值就是它与该类型上界的差。

<<:(左移)：a << b就表示把a转为二进制后左移b位(在后边填b个0)。例如100的二进制位1100100，而110010000转成十进制是400，那么100<<2 = 400.可以看出，(a<<b的值实际上是a乘2的b次方）因为在二进制后添加一个0就相当于该数乘2。

>>:(右移)：a shr b表示二进制右移b位（去掉末b位），相当于a除以2的b次方（取整）。我们也经常用shr 1来代替除2，比如[二分查找](https://baike.baidu.com/item/%E4%BA%8C%E5%88%86%E6%9F%A5%E6%89%BE" \t "https://baike.baidu.com/item/%E4%BD%8D%E8%BF%90%E7%AE%97/_blank)、堆的插入操作等等。想办法用shr代替除法运算可以使程序效率大大提高。最大公约数的二进制算法用除以2操作来代替慢得出奇的mod运算，效率可以提高60%。

### 位运算应用

判断奇偶：常用的 if( a%2 != 0 ) 可以替换为 if(a&1){奇数}，原理：任何偶数二进制第一位数必定0，而奇数必定是1，而1的二进制就是1，所以可以用这个判断。

交换变量：

一种a = [a=b,b][0],这个确实装逼  
 其实还有更稳重的方法  
 a^=b,b^=a,a^=b  
 原理：  
 第一步没啥好说a = a^b  
 第二步：b=b^a,也就是b=b^a^b,也就是b=a^0,此处换值  
 第三步：a=a^b 也就是a=a^b^a,也就是b

向下取整  
 一般方法 Math.floor()  
 位运算x|0

4乘以2的n次方

x<<n

原理：2进制移动一位相当于乘以2