/\*

体会位运算的妙用

位运算：

&：与 ~：非

|：或 ^：异或

<<左移，删去最后一位，第一位补0.>>右移反之。

利用位运算，我们可以实现一下功能：

\*/

// //判断一个数字的奇偶性

// //判断一个二进制位是1还是0

// //交换两个整型变量的值

// /\*

// \*\*\*异或的性质\*\*\*

// 1.交换律 a^b^c==a^c^b

// 2.结合律 (a^b)^c==a^(b^c)

// 3.任意x,x^x=0,x^o=x

// 4.自反性 A^B^B=A^0=A

//

// \*/

// //不使用判断语句，求整数的绝对值

// /\*

// 注：>>:算术右移

// 最右侧不要了，最左侧补上符号位

// >>>逻辑右移

// 最右侧不要了，最左侧直接补0

// 注：对于int类型的数据，移位超出32位时要模32.

// 如，1<<35==1<<3

// \*/

// //似乎C语言只支持算术右移。。。逻辑右移达咩。。。

// //实战：找出数组中唯一成对的数字

// /\*

// 1-10这10个数放在含有11个元素的数组中

// 只有唯一一个元素重复，其他均只出现一次

// 要求每个数组元素只能够被访问一次，

// 请设计一个算法，将它找出来，

// 不用辅助存储空间.

// \*/

// //woc牛

// /\*

// 让0跟数组中所有数字异或，再跟1-10这十个数字异或，得到的最终结果就是那个数

// 运用性质后两条

// \*/

// //实战2：找出落单的数

// /\*

// 一个数组中除了某一个元素中之外，

// 其他的元素都出现了两次，

// 请写程序找出这份只出现一次的数字

// \*/

// //甚至只用异或一次

// //实战3：输出一个整数二进制中1的个数

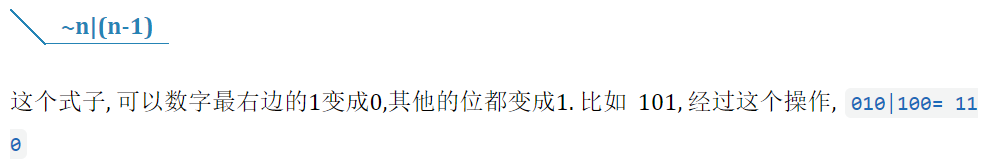
//法一：让整数不断右移，循环32次。

////法二： N&(N-1)可以消掉一个1（为什么？你想想，牛逼）

//// 故而不断循环，直到所有1被消掉，得到整数为0

////这个思路牛逼可拓展到很多有关二进制中1的题目

////如：判断一个数是否为2的幂次方（只有一个1）



//将整数的奇偶位互换。 可以把异或理解为不进位的+，且理解为乘。

