1.shift

left shift <<

right shift >>

2.XOR

2.0 语法

 0^num = num; 1^num = ~num; num ^ num = 0;

 a^a^b^b^c = (a^a)^(b^b)^c=0^0^c=c;

2.1 实现两个数的交换

a=a^b;

b=a^b;//b=a^b^b=a^0=a;

a=a^b;//a=a^a^b=0^b=b;

2.2 在一排数中找一个独一无二的数

全部异或，出现相同的数，异或为0

2.3 判断两个数是否相等

a^b == 0

2.4 符号判断

if ((dividend<0)^(divisor<0)) res=-re

3. 奇数偶数

(a & 1) == 1

(a & 1) == 0

4. INT 32bit

00000000 00000000 00000000 00000101 = 5

5 >>= 1

00000000 00000000 00000000 00000010 = 2

5 -> -5

00000000 00000000 00000000 00000101 = 5

11111111 11111111 11111111 11111010 取反

11111111 11111111 11111111 11111011 +1

11111111 11111111 11111111 11111011 = -5

01111111 11111111 11111111 11111111 = INT\_MAX

INT\_MAX = 2\*\*31 - 1

00000000 00000000 00000000 00000100 = 4

00000000 00000000 00000000 00000011 = 3

00000000 00000000 00000000 00000010 = 2

00000000 00000000 00000000 00000001 = 1

00000000 00000000 00000000 00000000 = 0

11111111 11111111 11111111 11111111 = -1

11111111 11111111 11111111 11111110 = -2

11111111 11111111 11111111 11111101 = -3

11111111 11111111 11111111 11111100 = -4

10000000 00000000 00000000 00000000 = INT\_MIN

INT\_MIN = -2\*\*31

11111111 11111111 11111111 11111011 = -5

-5 >>= 1

11111111 11111111 11111111 11111101 = -3

5. find the right most 1 number;

diff &= -diff;

101 = 5     011 = -5     diff = 1;

100 = 4     100 = -4     diff = 100;

6.  figuring out if n is either 0 or an exact power of two

(n & (n-1)) == 0