1：取反，使用 | 语句将 x 与 y 原先同为 1 的位改为 0，再取反

2：计算出需要右移的位数，右移，再保留低位即可

3：右移之后，将高位改为 0，低位保持不变

4：使用二分法，通过以为设置五个字符串：

0101 0101 0101 0101 0101 0101 0101 0101
0011 0011 0011 0011 0011 0011 0011 0011
0000 1111 0000 1111 0000 1111 0000 1111
0000 0000 1111 1111 0000 0000 1111 1111
0000 0000 0000 0000 1111 1111 1111 1111
分别通过按位与统计 1 的个数，并将个数记录在下一个字符串 1 出现的位置。
最后可以得到 1 的个数。

5：除了 0，其他数与其相反按位或，符号位均为 1。所以异或之后，判断符位即可。

6：最小数为 0x80000000，用 1 移位即可

7：减 n 计算位移的位数，左移再右移之后判断与原数异或，

若两者相同表示 x 可被表示为一个 n 位整数，！0 为 1

8：先计算出偏置量，加上右移即可

9：取反加一

10：判断符号位，判断是否为 0，按位与即可

11：取符号位，分三种情况，符号不同，符号相同且不等，相等

符号不同时，对 x 符号取非两次，y 符号取非一次，按位与即可
符号相同不等，两者相减，判断差的符号位即可
判断是否相等按位异或即可

12：使用二分法，先右移 16 位后若大于 0 即得到有效数字，否则得到 0，

判断最高位是否为 0，若不为 0，则包含 2 的 16 次方。即得到最高位的 log 数。
同理其他

13：将该数字的最高位进行取反。然后分类讨论，比较最高位为零时，

是否大于 0 11111111 0000 0000 0000 0000 0000。即阶为最大时，此时，为 NaN，返回即可
否则返回取反结果

14：Int 型整数在转化为 float 型数的时候需要注意的是负数的表示，

在 int 型中负数使用补码的形式表示，而 float 直接表示，所以先要对负数进行转化。

然后进行循环，每移位一次阶码记录一次。最后把得到的三部分综合起来即可

15：先进行判断，如果阶码全零，则需要对尾数进行移位操作，并令阶码加一，判定符号位。如果阶码不为零，则只需要领阶码加一即可。当阶码全为 1，不操作直接输出