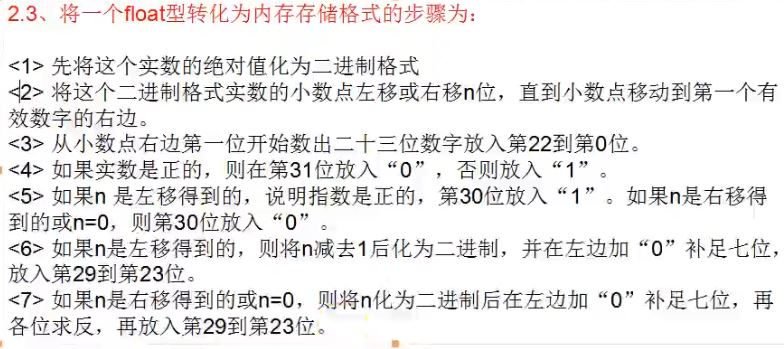
<参考答案见视频源码下载>





## 指数偏移值

浮点的二进制表示中指数位 e 的计算值（即转换成十进制后的值），要在实际指数值（十进制）的基础上加上一个 **偏移值**，标准中规定偏移值为 2^(e-1) - 1，如 32 为中 e 是 8 位，所以偏移值为 127，64 位的就为 1023；所以 32 位指数实际值的范围为 -126 ~ 127；

举个例，指数 e 的实际值为 -3，那么 32 位中加上偏移值就是 -3 + 127 = 124，换算成二进制的 8 位指数位就是 01111100;

这种指数位偏移后的指数值，又叫做 **阶码**，因为科学计数法中的指数是有正负之分的，所以实际指数值加上一个正的适中偏移值，就可以使得浮点表示法中的指数位为无符号的整型（就是变成正整数），利于浮点数的比较大小，就是可以直接从浮点的二进制表示中，由高位向低位逐位进行比较（如果是负数二进制比较大小要复杂一点）。

32位中 float， 为4字节

0.25D

整数部分为： 0

小数部分： 0.25 \* 2 = 0.5 0

0.5 \* 2 = 1.0 1

0.01B

0.01向右移两位为1.0用-2来表示向右移（如果向左移2位用2来表示）（这里始终保证小数点左边只有一个1）

-2 -1（固定不变的减1）= -3D（十进制）= FD H(十六进制）= 1111 1101 B（二进制）把它的后7位 放到下面8的那列的后7位中

1（表示是正数还是负数）8 23 = 4字节（32位）

0 0（向左为1，向右为0）111 1101 00000000000000000000000（代表移动后1.0中的小数部分）

那么计算机中0.25在内存中的表示为： 0011 1110 1000 0000 0000 0000 0000 0000 B = 3e800000 H

https://blog.csdn.net/uvarandmethod/article/details/78992793

1、0.25(Float) 存储到内存中的值是多少？

3E800000

2、-7.25 存储到内存中的值是多少？

7/2 1

3/2 1

1/2 1

0.25 0

0.5 1

111.01

1.1101 左移第一位就是1 ,2位-1,

符号位 指数位

1 10000001 1101000000000000

3、编写一个函数，能够实现英寸与厘米之间的转换(1英寸 = 2.54厘米)。

说明：

1) 函数需要什么样的参数、返回值等自由发挥。

2) \*代表乘法 /代表除法

float transformation(float inch)

{

return inch \* 2.54F;

}

void main()

{

float temp = transformation(3.8);

return;

}