2.86

Smallest positive denormalized

(0 0…00 00…001)2  2-16445

Smallest positive normalized

(0 0…01 10…000) 2  2-16382

Largest normalized

(0 1…10 11…111)2  (2-2-63)\*216384

2.95

unsigned float\_half(unsigned uf) {

int S=uf&0x80000000;

int E=(uf>>23)&255;

int M=uf&0x7fffff;

if (!(E^255)) return uf;

if (E)

{

E--;

if (!E) M|=0x800000;

}

if (!E)

{

int t=M&1;

M>>=1;

if (t) M+=M&1;

}

return S|(E<<23)|M;

}

S为符号位，E为指数exp，M为有效数字frac

若uf是denormalized的inf和nan，则直接返回uf

若uf是normalized，则E--，此时若E为0，则M不变（实际上已经），反之将隐式的1加上

若处理后的E不为0，则将M右移一位，然后处理舍入

2.97

unsigned float\_i2f(int x) {

int E=0x4e800000;

int S=x&0x80000000;

int t=0;

unsigned ux;

if (x)

{

if (x<0) x=-x;

while (x>0)

{

x<<=1;E-=0x800000;

}

ux=x;

if (ux&0x80)

if (ux&0x17f) t=1;

return S+E+(ux>>8)+t;

}

return 0;

}

特判x为0的情况，如果x为负数则取反。

将x左移，直到变为负数（也就是x的符号位为1）。这样有两个作用：

1. 将x的最左边的0，舍去
2. 在x舍入到23位的时候，可以知道舍入的位置

在x左移的同时，frac相当于＊2，那么我们的exp要相应的－1，exp的初始值是0x4e800000，而不是0x4f000000，的原因是我们不需要消去x符号位上的1，直接将其加到exp上即可。

X需要转换为unsigned的原因是，有符号的右移是算数右移而不是逻辑右移。