1 请写出单精度浮点数的“非负值最小规格化数”的小数表示 和 “最小非规格化数”的二进制表示(单精度浮点数的阶码字段占用 8 位)。

0 0000 0001 000···0

1 0000 0000 111···1

2 写出 8 位浮点数（阶码采用 4 位，小数位采用 3 位）“0 0110 110”所表示的数值。

0.111=0.875

3 现有以下代码:

```
short si = -8196;   //8196 = 0x2004
int i = si;
```

经过运算后， i 的机器数表示是多少?

FF FF DF FC

4 采用类似 IEEE754 浮点格式的 8bit 浮点数中(1 个符号位，3 个阶码位，4 个小数位)，正无穷（+∞）的表示是什么？NaN 的表示是什么？最大的非规格化的正数是什么？如果减少 1 位阶码位，将其用于小数部分，将会有怎样的效果?

0 111 0000; 0 1111 /=0; 0 000 1111;

表示的数值范围变小了，靠近 0 周围的数表示精度变低，越大的数表示精度越大。

5 考虑一种遵从IEEE规范的新浮点格式，包含3个阶码位和3个小数位（即该浮点数不考虑符号位，只用来表示正数）。请回答下列问题。

1）Bias 值为多少？ 3

2）除 0 和 Infinity 外，该浮点数能表示的数值范围为多少？1/32--15

3) 尝试填下以下表格的空白处。如果一个数值太大而无法表达，使用 infinity 的表达式；如果一个数值太小而无法表达，使用 0 的表达式。

二进制表达	十进制数值
011001	1.125
Infinity	17
110001	9
110010	9+1/2

二、已知 f(n)=1111…111B（n+1 个 1），计算 f(n)的 C 语言函数 f1 如下：

```
int f1(unsigned n)
{
    int sum = 1, power = 1;
    for (unsigned i = 0; i <= n - 1; i++)
    {
            power *= 2;
            sum += power;
    }
    return sum;
}
```

将 f1 中的 int 都改为 float，可得到计算 f(n)的另一个函数 f2。

```
float f2(unsigned n)
{
    float sum = 1, power = 1;
    for (unsigned i = 0; i <= n - 1; i++)
    {
            power *= 2;
            sum += power;
    }
    return sum;
}
```

假设 unsigned 和 int 型数据都占 32 位，float 采用 IEEE754 单精度标准，请回答如下问题:

(1) 当 n = 0 时，f1 会出现死循环，为什么？若将 f1 中的变量 i 和 n 都定义为 int 型，则 f1 是否还会出现死循环？为什么？

-1 的二进制 111…1 被翻译为一个无符号数，类似进入一个死循环。为有符号数时不会出现死循环。

sum = \sum_{i = 1}^{n + 1} 2^i = 2^{n + 1} - 1

(2) 若使 f2(n)的结果不溢出，则最大的 n 是多少？若使 f2(n)的结果精确（无舍入），则最大的 n 是多少？ **127 ， 23**