§12.指针进阶 - 浮点数机内存储格式(IEEE 754)的理解

补充：

```
//用于看懂 float/double 内部存储格式的例子
#include <iostream>
using namespace std;
int main()
{
    float f = 123.456f;
    char* p = (char*)&f;
    cout << hex << (int)(*p)     << endl;
    cout << hex << (int)(*(p+1)) << endl;
    cout << hex << (int)(*(p+2)) << endl;
    cout << hex << (int)(*(p+3)) << endl;
    return 0;
}
//注意：x86 系列 CPU 的多字节数据存储，是低位在前
```

```
79
ffffffe9
fffffff6
42
```

2、结合上课内容及上面的示例程序，自行查阅相关资料，并回答一下的问题

（1） float 型数据的 32bit 是如何分段来表示一个单精度的浮点数的？给出 bit 位的分段解释，尾数的正负如何表示？尾数如何表示？指数的正负如何表示？指数如何表示？

符号位 阶码 尾数

一共 32 位

1 8 23

符号位 阶码 尾数

SEEE EEEE EMMM MMMM MMMM MMMM MMMM MMMM

S 是符号位，E 为阶码，M 为尾数。

符号位 0 为正，1 为负

阶码是 8 位，2 的指数你并不能直接当作阶码来处理，需要与 127 相加才可得到2^n的阶码

尾数的位域长度是 23 位，实际是 24 位，不可见为 1.

阶码=阶数+127

十进制转二进制

小数点前部分转二进制，小数点之后乘 2 取正序

然后写出数规格化

以 8.25 为例，则得到 1000.01，之后为 $1.00001*2^3$ 3 是因为小数点向前移 3 位。

阶数+127 为 130=1000 0010，尾数为 00001

所以为 1100 0001 0000 0100 0000 0000 0000 0000

C1 04 00 00

（2） 为什么 float 型数据只有 7 位有效数字？为什么最大只能是 $3.4x10^{38}$？

因为单精度的尾数用 23 位存储，加上默认的小数点前的一位 1，2^{24}=16777216

因为10^7<16777216<10^8，所以单精度的浮点数有效位数为 7 位。

（3） double 型数据的 64bit 是如何分段来表示一个双精度的浮点数的？给出 bit 位的分段解释，尾数的正负如何表示？尾数如何表示？指数的正负如何表示？指数如何表示？

一共 64 位

1　　11　　52

符号位　　阶码　　尾数

SEEE EEEE EEEE MMMM MMMM MMMM MMMM MMMM MMMM MMMM MMMM MMMM MMMM MMMM MMMM MMMM

S 是符号位，E 为阶码，M 为尾数。

（4） 为什么 double 型数据有 15 位有效数字？为什么最大是 $1.7x10^{308}$？

双精度尾数用 52 位存储，2^{52+1}=9007199254740992，因为10^{16}<9007199254740992<10^{17}，所以双精度为 16，最大保证为 15.

（5） 给出下列 8 个小题（float/double 各自有尾数正负/指数正负）对应变量的 32/64bit 的具体值及解释（写二进制表示时，每 8bit 加 1 个“-”方便查看，例：00100000-01010001）

a) float d=123.456

123 的二进制 1111011　　0.456 二进制为 01110010010

1111011.01110010010=1.11101101110010010×2^6

阶码 133=10000101，符号位为 0

01000010 11110110 11100100 10000000

b) float d=-123.456

123 的二进制 1111011　　0.456 二进制为 01110010010

1111011.01110010010=1.11101101110010010×2^6

阶码 133=10000101，符号位为 1

11000010 11110110 11100100 10000000

c) float d=0.123e-3

00111001 00000000 11111000 00000000

d) float d=-1.23e-4

10111001 00000000 11111000 00000000

e) double d=123.456

01000010 11110110 11101001 01111000

f) double d=-123.456

11000010 11110110 11101001 01111000

g) double d=0.123e-3

00101001 00000000 11111001 10001111

h) double d=-1.23e-4

10101001 00000000 11111001 10001111

【文档格式要求：】

1、文档用自己的语言组织
2、如果用到某些小测试程序进行说明，可以贴上小测试程序的源码及运行结果
3、为了使文档更清晰，允许将网上的部分图示资料截图后贴入
4、不允许在答案处直接贴某网址，再附上“见**”（或类似行为），否则实验报告部分直接总分-50
（注：上学期 VS2019 的 Debug 报告，有同学直接将官方文档复制后贴入，后果是优变及格）

【作业要求：】

1、3 月 18 日前网上提交本次作业，直接在本文档上作答，转换为 pdf 后提交即可
2、每题所占平时成绩的具体分值见网页（本题在“实验报告”中提交）
3、超过截止时间提交作业会自动扣除相应的分数，具体见网页上的说明