§6.指针基础 - 浮点数机内存储格式(IEEE 754)理解

要求：

1、完成视频"21221-060002-W1301.第06模块 指针基础-引用及不同类型的指针的互相转换.mp4"的学习

2、用于看懂float/double内部存储格式的例子如下

```
#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：x86系列CPU的多字节数据存储，是低位在前
```

3、完成后续page的内容

4、转换为pdf后在"文档作业"中提交（12.9前）

§6.指针基础 - 浮点数机内存储格式(IEEE 754)理解

2、float型数的机内表示

格式要求：多字节时，每8bit中间加一个空格或-(例："11010100 00110001"或"11010100-00110001")

A. 2151140.0411512 （此处假设学号是1234567，各人换成自己的学号，按1234567做的0分!!!）
注：尾数为正、指数为正

(1) 得到的32bit的机内表示是：__01001010 00000011 01001011 10010000_____

(2) 其中：符号位是____0_____

指数是___10010100_____(填32bit中的原始形式)
指数转换为十进制形式是___148_____(32bit中的原始形式按二进制原码形式转换)
指数表示的十进制形式是____21_____(32bit中的原始形式按IEEE754的规则转换)

尾数是____0000011 01001011 10010000_____(填32bit中的原始形式)
尾数转换为十进制整数形式是_____0.0257434844970703_____(32bit中的原始形式按二进制原码形式转换)
尾数表示的十进制小数形式是____1.0257434844970703_____(保留小数点后)

§6.指针基础 - 浮点数机内存储格式(IEEE 754)理解

2、float型数的机内表示

格式要求：多字节时，每8bit中间加一个空格或-(例："11010100 00110001"或"11010100-00110001")

B.-2151140.0411512 （此处假设学号是1234567，各人换成自己的学号，按1234567做的0分!!!）
注：尾数为负、指数为正

(1) 得到的32bit的机内表示是：__11001010 00000011 01001011 10010000_____

(2) 其中：符号位是____1_____

指数是___10010100_____(填32bit中的原始形式)
指数转换为十进制形式是___148_____(32bit中的原始形式按二进制原码形式转换)
指数表示的十进制形式是____21_____(32bit中的原始形式按IEEE754的规则转换)

尾数是____0000011 01001011 10010000_____(填32bit中的原始形式)
尾数转换为十进制整数形式是_____0.0257434844970703_____(32bit中的原始形式按二进制原码形式转换)
尾数表示的十进制小数形式是____1.0257434844970703_____(保留小数点后)

§6.指针基础 - 浮点数机内存储格式(IEEE 754)理解

2、float型数的机内表示

格式要求：多字节时，每8bit中间加一个空格或-(例："11010100 00110001" 或 "11010100-00110001")

C. 0.002151140 （此处假设学号是1234567，各人换成自己的学号，按1234567做的0分!!! ）
注：尾数为正、指数为负

(1) 得到的32bit的机内表示是：__00111011 00001100 11111010 00100100_____

(2) 其中：符号位是____0_____

指数是___01110110_____(填32bit中的原始形式)
指数转换为十进制形式是___118_____(32bit中的原始形式按二进制原码形式转换)
指数表示的十进制形式是____-9_____(32bit中的原始形式按IEEE754的规则转换)

尾数是____0001100 11111010 00100100_____(填32bit中的原始形式)
尾数转换为十进制整数形式是_____0.10138368606567383_____(32bit中的原始形式按二进制原码形式转换)
尾数表示的十进制小数形式是____1.10138368606567383_____(保留小数点后)

§6.指针基础 - 浮点数机内存储格式(IEEE 754)理解

2、float型数的机内表示

格式要求：多字节时，每8bit中间加一个空格或-(例："11010100 00110001" 或 "11010100-00110001")

D.-0.002151140 （此处假设学号是1234567，各人换成自己的学号，按1234567做的0分!!! ）
注：尾数为负、指数为负

(1) 得到的32bit的机内表示是：__10111011 00001100 11111010 00100100_____

(2) 其中：符号位是____1_____

指数是___01110110_____(填32bit中的原始形式)
指数转换为十进制形式是___118_____(32bit中的原始形式按二进制原码形式转换)
指数表示的十进制形式是____-9_____(32bit中的原始形式按IEEE754的规则转换)

尾数是____0001100 11111010 00100100_____(填32bit中的原始形式)
尾数转换为十进制整数形式是_____0.10138368606567383_____(32bit中的原始形式按二进制原码形式转换)
尾数表示的十进制小数形式是____1.10138368606567383_____(保留小数点后)

§6.指针基础 - 浮点数机内存储格式(IEEE 754)理解

3、double型数的机内表示

格式要求：多字节时，每8bit中间加一个空格或-(例："11010100 00110001" 或 "11010100-00110001")

A. 2151140.0411512 （此处假设学号是1234567，各人换成自己的学号，按1234567做的0分!!! ）
注：尾数为正、指数为正

(1) 得到的64bit的机内表示是：01000001 01000000 01101001 01110010 00000101 01000100 01110001 01001001

(2) 其中：符号位是___0_____

指数是_____1000001 0100_____(填64bit中的原始形式)
指数转换为十进制形式是____1044_____(64bit中的原始形式按二进制原码形式转换)
指数表示的十进制形式是_____21_____(64bit中的原始形式按IEEE754的规则转换)

尾数是__0000 01101001 01110010 00000101 01000100 01110001 01001001 ____(填64bit中的原始形式)
尾数转换为十进制整数形式是___0.025743504119491556_____(64bit中的原始形式按二进制原码形式转换)
尾数表示的十进制小数形式是____1.025743504119491556_____(保留小数点后)

§6.指针基础 - 浮点数机内存储格式(IEEE 754)理解

3、double型数的机内表示

格式要求：多字节时，每8bit中间加一个空格或-(例："11010100 00110001" 或 "11010100-00110001")

B.-2151140.0411512 （此处假设学号是1234567，各人换成自己的学号，按1234567做的0分!!!）
注：尾数为负、指数为正

(1) 得到的64bit的机内表示是：11000001 01000000 01101001 01110010 00000101 01000100 01110001 01001001

(2) 其中：符号位是___1_____

指数是_____1000001 0100_____(填64bit中的原始形式)
指数转换为十进制形式是____1044_____(64bit中的原始形式按二进制原码形式转换)
指数表示的十进制形式是_____21_____(64bit中的原始形式按IEEE754的规则转换)

尾数是___0000 01101001 01110010 00000101 01000100 01110001 01001001____(填64bit中的原始形式)
尾数转换为十进制整数形式是___0.0257435041194915 56_____(64bit中的原始形式按二进制原码形式转换)
尾数表示的十进制小数形式是____1.0257435041194915 56_____(保留小数点后)

§6.指针基础 - 浮点数机内存储格式(IEEE 754)理解

3、double型数的机内表示

格式要求：多字节时，每8bit中间加一个空格或-(例："11010100 00110001" 或 "11010100-00110001")

C. 0.002151140 （此处假设学号是1234567，各人换成自己的学号，按1234567做的0分!!!）
注：尾数为正、指数为负

(1) 得到的64bit的机内表示是：0011111 01100001 10011111 01000100 01111110 01011111 00101011 10001010

(2) 其中：符号位是____0_____

指数是__0111110 1100_____(填64bit中的原始形式)
指数转换为十进制形式是___1004_____(64bit中的原始形式按二进制原码形式转换)
指数表示的十进制形式是___-19___(64bit中的原始形式按IEEE754的规则转换)

尾数是____0001 10011111 01000100 01111110 01011111 00101011 10001010 ____(填64bit中的原始形式)
尾数转换为十进制整数形式是___0.10138368000000009_____(64bit中的原始形式按二进制原码形式转换)
尾数表示的十进制小数形式是___1.10138368000000009_____(保留小数点后)

§6.指针基础 - 浮点数机内存储格式(IEEE 754)理解

3、double型数的机内表示

格式要求：多字节时，每8bit中间加一个空格或-(例："11010100 00110001" 或 "11010100-00110001")

D.-0.002151140 （此处假设学号是1234567，各人换成自己的学号，按1234567做的0分!!!）
注：尾数为负、指数为负

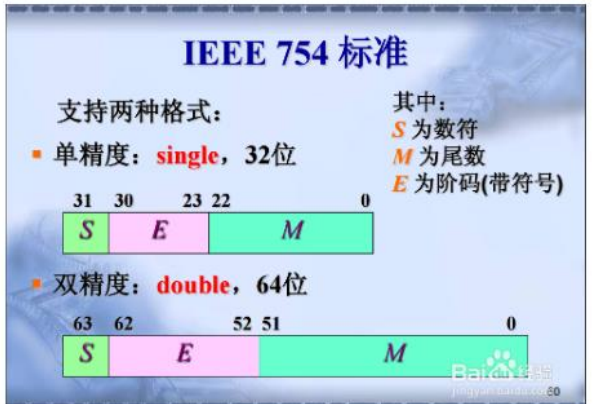
(1)得到的64bit的机内表示是：1011111 01100001 10011111 01000100 01111110 01011111 00101011 10001010

(2) 其中：符号位是____1_____

指数是__0111110 1100_____(填64bit中的原始形式)
指数转换为十进制形式是___1004_____(64bit中的原始形式按二进制原码形式转换)
指数表示的十进制形式是___-19___(64bit中的原始形式按IEEE754的规则转换)

尾数是____0001 10011111 01000100 01111110 01011111 00101011 10001010 ___(填64bit中的原始形式)
尾数转换为十进制整数形式是___0.10138368000000009_____(64bit中的原始形式按二进制原码形式转换)
尾数表示的十进制小数形式是___1.10138368000000009_____(保留小数点后)

§6.指针基础 - 浮点数机内存储格式(IEEE 7

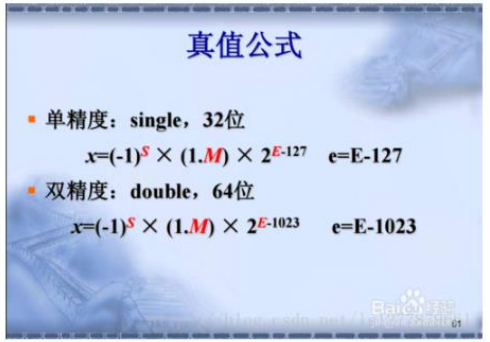


4、总结

(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？

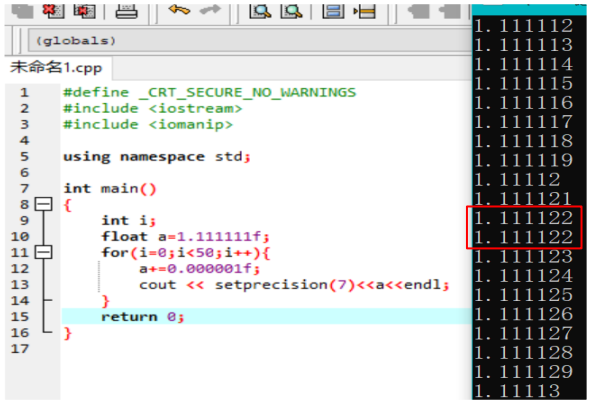
1个bit位（31）用0或1表示正负S（数符），
8个bit位（30-23）表示与2的指数次方有关的E（阶码），
23个bit位（22-0）表示小数部分M（尾数）；
真值 x = $(-1)^S * (1.M) * 2^{E-127}$ (指数e=E-127)

	S(数符)	E(阶码)	M(尾数)	总长
float	1	8	23	32
double	1	11	52	64



(2) 为什么float型数据只有7位十进制有效数字？为什么最大只能是$3.4x10^{38}$ ？
有些资料上说有效位数是6~7位，能找出6位/7位不同的例子吗？

23个bit位最小的分辨率为2^{-23} =0.0000001192……，故只能保证最多7位有效数字；
8个bit位最大表示的阶码E为255，指数e最大为255-127=128，
2^{128} =3.4028…*10^{38} ，故最大只能到这个数量级；
出现6、7位不同的原因是，虽然最小分辨位数可以达到7位，
但是第7位并不是都完全准确覆盖，大部分情况仍然是近似等于，可能会出现误差。



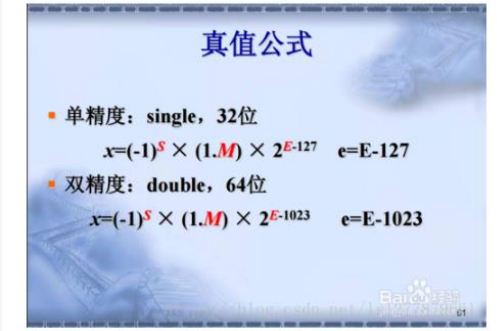
(3) double型数据的64bit是如何分段来表示一个双精度的浮点数的？给出bit位的分段解释
尾数的正负如何表示？尾数如何表示？指数的正负如何表示？指数如何表示？

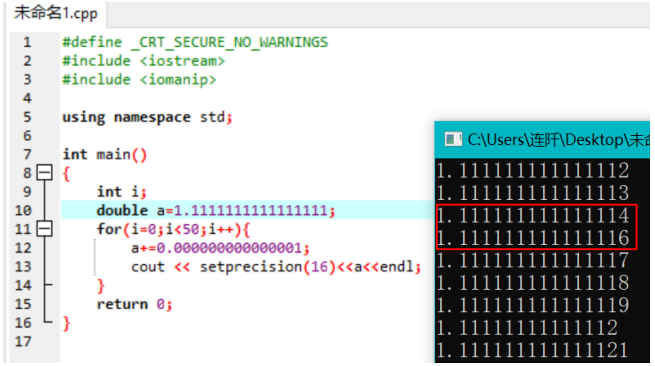
1个bit位（63）用0或1表示正负S（数符），
11个bit位（62-52）表示与2的指数次方有关的E（阶码），
52个bit位（51-0）表示小数部分M（尾数）；
真值 x = $(-1)^S * (1.M) * 2^{E-1023}$ (指数e=E-1023)

	S(数符)	E(阶码)	M(尾数)	总长
float	1	8	23	32
double	1	11	52	64

(4) 为什么double型数据只有15位十进制有效数字？为什么最大只能是$1.7x10^{308}$ ？
有些资料上说有效位数是15~16位，能找出15位/16位不同的例子吗？

52个bit位最小的分辨率为2^{-52} =$2.22*10^{-16}$ ，故只能保证最多16位有效数字；
11个bit位最大表示的阶码E为2047，指数e最大为2047-1023=1024，
2^{1024} =$1.797693\cdots*10^{308}$ ，故最大只能到这个数量级；
出现15、16位不同的原因是，虽然最小分辨位数可以达到16位，
但是第16位并不是都完全准确覆盖，大部分情况仍然是近似等于，可能会出现误差。





注：

- 文档用自己的语言组织
- 篇幅不够允许加页
- 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
- 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
- 不允许在答案处直接贴某网址，再附上“见**”（或类似行为），否则文档作业部分直接总分-50