§.基础知识题 - 浮点数机内存储格式(IEEE 754)理解

要求：

1、完成本文档中所有的题目并写出分析、运行结果

2、无特殊说明，均使用VS2022编译即可

3、直接在本文件上作答，写出答案/截图（不允许手写、手写拍照截图）即可；填写答案时，为适应所填内容或贴图，允许调整页面的字体大小、颜色、文本框的位置等

★ 贴图要有效部分即可，不需要全部内容

★ 在保证一页一题的前提下，具体页面布局可以自行发挥，简单易读即可

★ 不允许手写在纸上，再拍照贴图

★ 允许在各种软件工具上完成（不含手写），再截图贴图

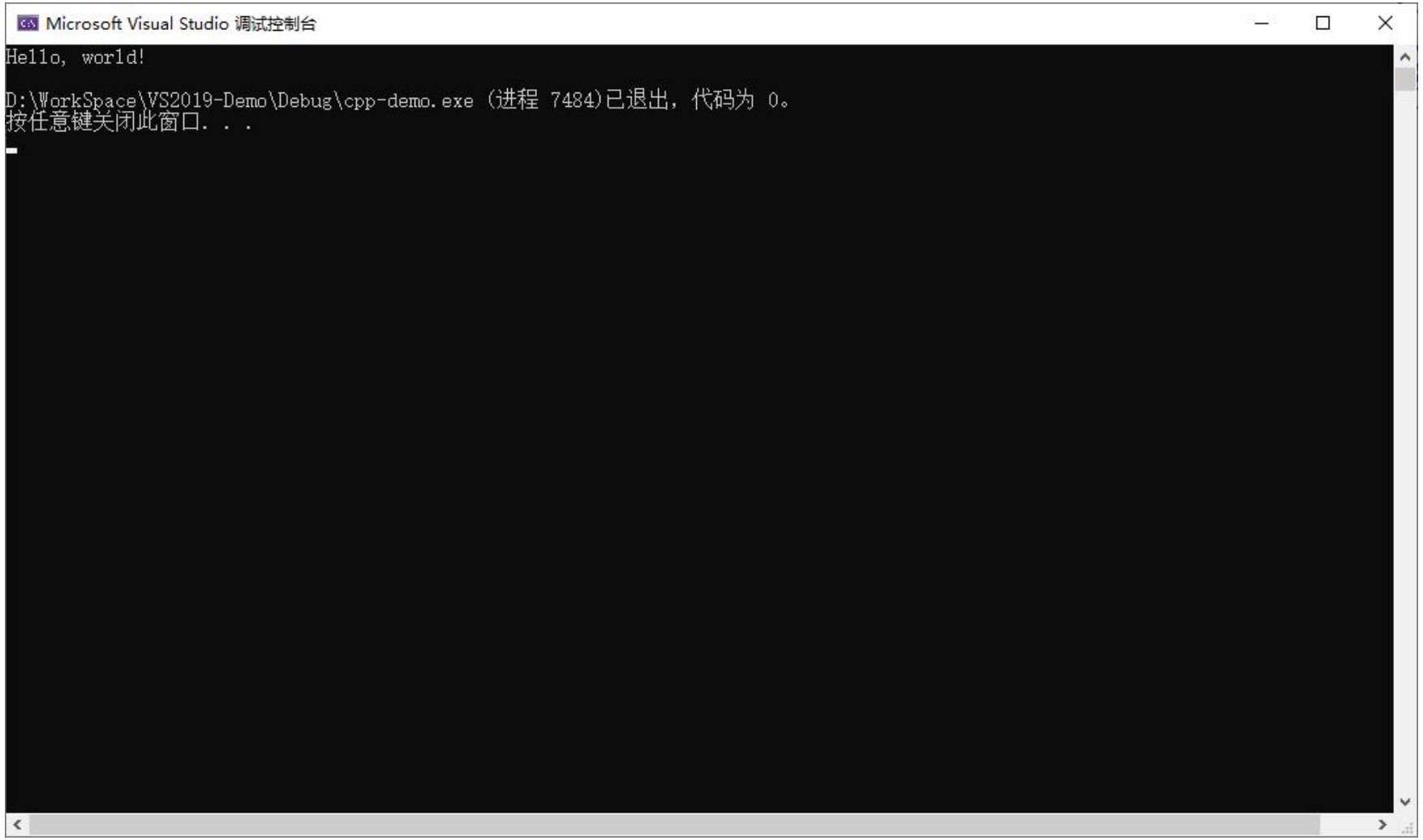
4、转换为pdf后提交

5、3月14日前网上提交本次作业（在“文档作业”中提交）

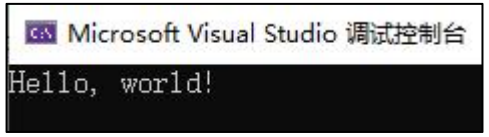
§.基础知识题 - 浮点数机内存储格式(IEEE 754)理解

贴图要求：只需要截取输出窗口中的有效部分即可，如果全部截取/截取过大，则视为无效贴图

例：无效贴图

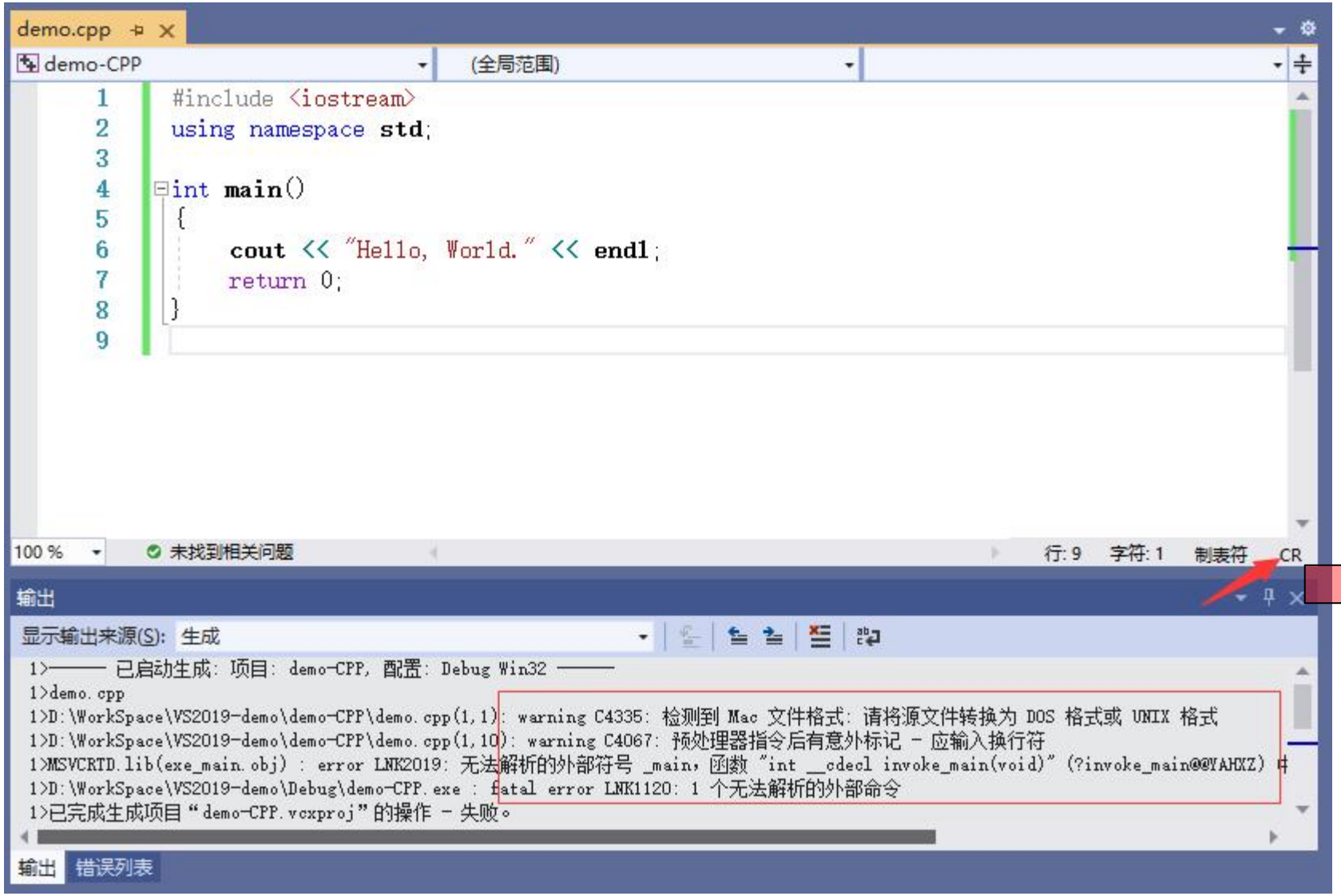


例：有效贴图



§.基础知识题 - 浮点数机内存储格式(IEEE 754)理解

附：用WPS等其他第三方软件打开PPT，将代码复制到VS2022中后，如果出现类似下面的编译报错，则观察源程序编辑窗的右下角是否为CR，如果是，单击CR，在弹出中选择CRLF，再次CTRL+F5运行即可



§.基础知识题 - 浮点数机内存储格式(IEEE 754)理解

基础知识：用于看懂float型数据的内部存储格式的程序如下：

注意：除了对黄底红字的具体值进行改动外，其余部分不要做改动，也暂时不需要弄懂为什么（需要第6章的知识才能弄懂）

```
#include <iostream>
using namespace std;
int main()
{
    float f = 123.456;
    unsigned char* p = (unsigned 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：忽略本题出现的warning
```

上例解读：单精度浮点数123.456，在内存中占四个字节，四个字节的值依次为0x42 0xf6 0xe9 0x79（按打印顺序逆向取）

转换为32bit则为：0100 0010 | 1111 0110 | 1110 1001 | 0111 1001

符号位　8位指数　23位尾数

§.基础知识题 - 浮点数机内存储格式(IEEE 754)理解

基础知识：用于看懂double型数据的内部存储格式的程序如下：

注意：除了对黄底红字的具体值进行改动外，其余部分不要做改动，也暂时不需要弄懂为什么（需要第6章的知识才能弄懂）

```
#include <iostream>
using namespace std;
int main()
{
    double d = 1.23e4;
    unsigned char* p = (unsigned char*)&d;
    cout << hex << (int)(*p)     << endl;
    cout << hex << (int)(*(p+1)) << endl;
    cout << hex << (int)(*(p+2)) << endl;
    cout << hex << (int)(*(p+3)) << endl;
    cout << hex << (int)(*(p+4)) << endl;
    cout << hex << (int)(*(p+5)) << endl;
    cout << hex << (int)(*(p+6)) << endl;
    cout << hex << (int)(*(p+7)) << endl;
    return 0;
}
```

上例解读：双精度浮点数1.23e4，在内存中占八个字节，八个字节的值依次为0x40 0xc8 0x06 0x00 0x00 0x00 0x00 0x00(逆向)

转换为64bit则为：0100 0000 | 1100 1000 | 0000 0100 | 0000 0000 | 0000 0000 | 0000 0000 | 0000 0000 | 0000 0000

符号位　11位指数　52位尾数

§.基础知识题 - 浮点数机内存储格式(IEEE 754)理解

自学内容：自行以"IEEE754"/"浮点数存储格式"/"浮点数存储原理"/"浮点数存储方式"等关键字，
在网上搜索相关文档，读懂并了解浮点数的内部存储机制

学长们推荐的网址：

https://baike.baidu.com/item/IEEE%20754/3869922?fr=aladdin

https://zhuanlan.zhihu.com/p/343033661

https://www.bilibili.com/video/BV1iW411d7hd?is_story_h5=false&p=4&share_from=ugc&share_medium=android&share_plat=android&share_session_id=e12b54be-6ffa-4381-9582-9d5b53c50fb3&share_source=QQ&share_tag=s_i&timestamp=1662273598&unique_k=AuouME0

https://blog.csdn.net/gao_zhennan/article/details/120717424

https://www.h-schmidt.net/FloatConverter/IEEE754.html

§.基础知识题 - 浮点数机内存储格式(IEEE 754)理解

例：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")

例1：100.25

下面是float机内存储手工转十进制的的方法：

(1) 得到的32bit的机内表示是：__0100 0010 1100 1000 1000 0000 0000 0000_ (42 c8 80 00)

(2) 其中：符号位是____0_____

指数是__1000 0101__(填32bit中的原始形式)

指数转换为十进制形式是___133_____(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___6_____(32bit中的原始形式按IEEE754的规则转换)

1000 0101

- 0111 1111

= 0000 0110 (0x06 = 6)

尾数是___100 1000 1000 0000 0000 0000____(填32bit中的原始形式)

尾数转换为十进制小数形式是__0.56640625__(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__1.56640625__(加整数部分的1后)

100 1000 1000 0000 0000 0000 = $2^{0} + 2^{-1} + 2^{-4} + 2^{-8}$

= 0.5 + 0.0625 + 0.00390625 = 0.56640625 => 加1 => 1.56640625

1.56640625 x 2^{6} = 100.25 (此处未体现出误差)

下面是十进制手工转float机内存储的方法：

100 = 0110 0100 (整数部分转二进制为7位，最前面的0只是为了8位对齐，可不要)

0.25 = 01 (小数部分转二进制为2位)

100.25 = 0110 0100.01 = 1.1001 0001 x 2^{6} (确保整数部分为1，移6位)

符 号 位：0

阶 码：6 + 127 = 133 = 1000 0101

尾数(舍1)：1001 0001 => 1001 0001 0000 0000 0000 000 (补齐23位，后面补14个蓝色的0)

100 1000 1000 0000 0000 0000 (从低位开始四位一组，共23位)

注意：

1、作业中绿底/黄底文字/截图可不填

2、计算结果可借助第三方工具完成，没必要完全手算

本页不用作答

§.基础知识题 - 浮点数机内存储格式(IEEE 754)理解

例：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")

例2：1.2

下面是float机内存储手工转十进制的的方法：

(1) 得到的32bit的机内表示是：__0011 1111 1001 1001 1001 1001 1001 1010_ (3f 99 99 9a)

(2) 其中：符号位是____0_____

指数是__0111 1111__(填32bit中的原始形式)

指数转换为十进制形式是___127_____(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___0_____(32bit中的原始形式按IEEE754的规则转换)

0111 1111

- 0111 1111

= 0000 0000 (0x0 = 0)

尾数是___001 1001 1001 1001 1001 1010____(填32bit中的原始形式)

尾数转换为十进制小数形式是__0.20000004768371582031 25___(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__1.20000004768371582031 25___(加整数部分的1后)

001 1001 1001 1001 1001 1010 = $2^{-3} + 2^{-4} + 2^{-7} + 2^{-8} + 2^{-11} + 2^{-12} + 2^{-15} + 2^{-16} + 2^{-19} + 2^{-20} + 2^{-22}$

= 0.125 + ... + 0.00000023841857910156 25(详见右侧蓝色) = 0.20000004768371582031 25

=> 加1 = 1.20000004768371582031 25 (此处已体现出误差)

下面是十进制手工转float机内存储的方法：

1 = 1 (整数部分转二进制为1位)

0.2 = 0011 0011 0011 0011 0011 0011 (小数部分无限循环，转为二进制的24位)

=> 0011 0011 0011 0011 0011 010 (四舍五入为23位，此处体现出误差)

1.2 = 1.0011 0011 0011 0011 0011 010 = 1.0011 0011 0011 0011 0011 010 x 2^0 (确保整数部分为1，移0位)

符 号 位：0

阶 码：0 + 127 = 127 = 0111 1111

尾数(舍1)：0011 0011 0011 0011 0011 010 (共23位)

001 1001 1001 1001 1001 1010 (从低位开始四位一组，共23位)

注意：

1、作业中绿底/黄底文字/截图可不填

2、计算结果可借助第三方工具完成，没必要完全手算

0.125 +
0.0625 +
0.0078125 +
0.00390625 +
0.00048828125 +
0.000244140625 +
0.000030517578125 +
0.0000152587890625 +
0.0000019073486328125 +
0.0000009536743164062 5 +
0.0000002384185791015625

0.20000004768371582031 25

本页不用作答

§.基础知识题 – 浮点数机内存储格式(IEEE 754)理解

1、float型数的机内表示

格式要求：多字节时，每4bit中间加一个空格或-(例："1101 0100 0011 0001"或"1101-0100-0011-0001")

A.2351114.4111532 （此处设学号是1234567，需换成本人学号，小数为学号逆序，非本人学号0分，下同!!!）
注：尾数为正、指数为正

(1) 得到的32bit的机内表示是：_0100 1010 0000 1111 1000 0000 0010 1010_(不是手算，用P.4方式打印)

(2) 其中：符号位是____0_____

指数是_____1001 0100_____(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__000 1111 1000 0000 0010 1010_____(填32bit中的原始形式)
尾数转换为十进制小数形式是____0.12109875679016113281 25_____(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_____1.12109875679016113281 25_____(加整数部分的1)
注：转换为十进制小数用附加的工具去做，自己去网上找工具也行，但要满足精度要求（下同!!!）

§.基础知识题 - 浮点数机内存储格式(IEEE 754)理解

1、float型数的机内表示

格式要求：多字节时，每4bit中间加一个空格或-(例："1101 0100 0011 0001" 或 "1101-0100-0011-0001")

B.-4111532.2351114 （设学号为1234567，按规则更换为学号和学号逆序）

注：尾数为负、指数为正

(1) 得到的32bit的机内表示是：___1100 1010 0111 1010 1111 0010 10110001___(不是手算，用P.4方式打印)

(2) 其中：符号位是___1_____

指数是_____1001 0100_____(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____111 1010 1111 0010 1011 0001_____(填32bit中的原始形式)
尾数转换为十进制小数形式是_0.960531353950500488281 25_(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____1.960531353950500488281 25_____(加整数部分的1)

§.基础知识题 - 浮点数机内存储格式(IEEE 754)理解

1、float型数的机内表示

格式要求：多字节时，每4bit中间加一个空格或-(例："1101 0100 0011 0001" 或 "1101-0100-0011-0001")

C.0.002351114 （设学号为1234567，按规则更换为学号和学号逆序）
注：尾数为正、指数为负

(1) 得到的32bit的机内表示是：__0011 1011 0001 1010 0001 0101 0010 0110__(不是手算，用P.4方式打印)

(2) 其中：符号位是_____0_____

指数是____0111 0110_____(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____001 1010 0001 0101 0010 0110_____(填32bit中的原始形式)
尾数转换为十进制小数形式是__0.203770399093627929687 5__(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___1.203770399093627929687 5___(加整数部分的1)

§.基础知识题 - 浮点数机内存储格式(IEEE 754)理解

1、float型数的机内表示

格式要求：多字节时，每4bit中间加一个空格或-(例："1101 0100 0011 0001" 或 "1101-0100-0011-0001")

D.-0.004111532 （设学号为1234567，按规则更换为学号和学号逆序）

注：尾数为正、指数为负

(1) 得到的32bit的机内表示是：__1011 1011 1000 0110 1011 1010 0000 1000__(不是手算，用P.4方式打印)

(2) 其中：符号位是_____1_____

指数是_____0111 0111_____(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_____000 0110 1011 1010 0000 1000_____(填32bit中的原始形式)
尾数转换为十进制小数形式是__0.05255222320556640625__(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___1.05255222320556640625___(加整数部分的1)

§.基础知识题 - 浮点数机内存储格式(IEEE 754)理解

2、double型数的机内表示

格式要求：多字节时，每4bit中间加一个空格或-(例："1101 0100 0011 0001" 或 "1101-0100-0011-0001")

A.2351114.4111532 （设学号为1234567，按规则更换为学号和学号逆序）

注：尾数为正、指数为正

(1) 得到的64bit的机内表示是：__0100 0001 0100 0001 1111 0000 0000 0101 0011 0100 1010 0000 1010 1011 0000 0110___(不是手算，用P.5方式打印)

(2) 其中：符号位是_____0_____

指数是_____100 0001 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___0001 1111 0000 0000 0101 0011 0100 1010 0000 1010 1011 0000 0110_____(填64bit中的原始形式)

尾数转换为十进制小数形式是___0.121098714424705544701055259793065488338470458984375_(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___1.121098714424705544701055259793065488338470458984375__(加整数部分的1)

§.基础知识题 - 浮点数机内存储格式(IEEE 754)理解

2、double型数的机内表示

格式要求：多字节时，每4bit中间加一个空格或-(例："1101 0100 0011 0001" 或 "1101-0100-0011-0001")

B.-4111532.2351114 （设学号为1234567，按规则更换为学号和学号逆序）

注：尾数为负、指数为正

(1) 得到的64bit的机内表示是：1100 0001 0100 1111 0101 1110 0101 0110 0001 1110 0001 1000 0010 0001 0101 1111 (不是手算，用P.5方式打印)

(2) 其中：符号位是 1

指数是 100 0001 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 1111 0101 1110 0101 0110 0001 1110 0001 1000 0010 0001 0101 1111 (填64bit中的原始形式)

尾数转换为十进制小数形式是 0.960531346851062783898100860824342817068099975585937 5 (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 -1.960531346851062783898100860824342817068099975585937 5 (加整数部分的1)

§.基础知识题 - 浮点数机内存储格式(IEEE 754)理解

2、double型数的机内表示

格式要求：多字节时，每4bit中间加一个空格或-(例："1101 0100 0011 0001" 或 "1101-0100-0011-0001")

C.0.002351114 （设学号为1234567，按规则更换为学号和学号逆序）

注：尾数为正、指数为负

(1) 得到的64bit的机内表示是：0011 1111 0110 0011 0100 0010 1010 0100 1011 0111 1010 0111 0100 0011 0001 1101 (不是手算，用P.5方式打印)

(2) 其中：符号位是 0

指数是 011 1111 0110 (填64bit中的原始形式)

指数转换为十进制形式是 1014 (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 -9 (64bit中的原始形式按IEEE754的规则转换)

尾数是 0011 0100 0010 1010 0100 1011 0111 1010 0111 0100 0011 0001 1101 (填64bit中的原始形式)

尾数转换为十进制小数形式是 0.203770368000000035024754652113188058137893676757812 5 (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 1.203770368000000035024754652113188058137893676757812 5 (加整数部分的1)

§.基础知识题 - 浮点数机内存储格式(IEEE 754)理解

2、double型数的机内表示

格式要求：多字节时，每4bit中间加一个空格或-(例："1101 0100 0011 0001" 或 "1101-0100-0011-0001")

D.-0.004111532 （设学号为1234567，按规则更换为学号和学号逆序）

注：尾数为负、指数为负

(1) 得到的64bit的机内表示是：1011 1111 0111 0000 1101 0111 0100 0000 1111 0111 1001 1111 1001 0001 1101 1111 (不是手算，用P.5方式打印)

(2) 其中：符号位是 1

指数是 011 1111 0111 (填64bit中的原始形式)

指数转换为十进制形式是 1015 (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 -8 (64bit中的原始形式按IEEE754的规则转换)

尾数是 0000 1101 0111 0100 0000 1111 0111 1001 1111 1001 0001 1101 1111 (填64bit中的原始形式)

尾数转换为十进制小数形式是 0.052552192000000053084818318893667310476303100585937 5 (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 -1.052552192000000053084818318893667310476303100585937 5 (加整数部分的1)

§.基础知识题 - 浮点数机内存储格式(IEEE 754)理解

3、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？

分为1bit的符号位，8bit的指数偏移值，23bit的分数值；

尾数的正负由符号位的0/1表示；尾数由原码表示；指数正负就是移码大小，小于127的就是负数；指数是由移码表示；

(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？

float类型的数值由二进制下的后23位决定的，而这后23位表示的十进制的数最大为2^23=8388608。也就是7位数字；

由于float的指数部分对应的指数范围为-128~128，所以取值范围为： -2^128到2^128，约等于-3.4E38 — +3.4E38；

1.563353125*2^6=100.125 （6位有效数字）

```
100.125
单精度浮点数机内表示：01000010 11001000 01000000 00000000
符号位是：0
指数（阶码）是：1000 0101
指数十进制是：133
指数真实值是：6
尾数是：100 1000 0100 0000 0000 0000
尾数十进制是：0.564453125
尾数真实值是：1.564453125
请按任意键继续. . .
```

1.953369140625*2^9=1000.125(7位有效数字)

```
1000.125
单精度浮点数机内表示：01000100 01111010 00001000 00000000
符号位是：0
指数（阶码）是：1000 1000
指数十进制是：136
指数真实值是：9
尾数是：111 1010 0000 1000 0000 0000
尾数十进制是：0.953369140625
尾数真实值是：1.953369140625
请按任意键继续. . .
```

总结（2）

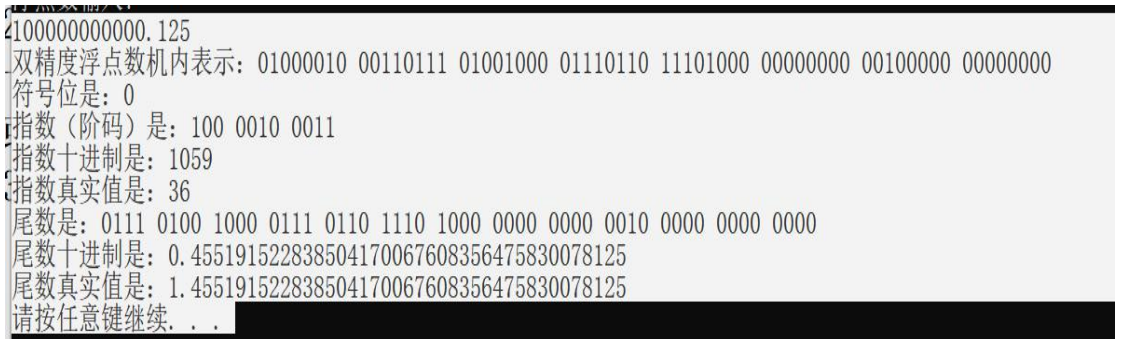
(3) double型数据的64bit是如何分段来表示一个双精度的浮点数的？给出bit位的分段解释
尾数的正负如何表示？尾数如何表示？指数的正负如何表示？指数如何表示？

1bit的符号位，11bit的指数偏移值，53bit的分数值；
尾数的正负由符号位0/1表示；尾数由原码表示；指数正负由移码大小表示，移码小于1023就是负数；
指数由移码表示；

(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？

由于double型数据是用64位二进制来存储的，其中52位用来存储尾数，所以它能表示的最大的二进制数字是2^{52+1} = 9007199254740992。这个数字在十进制中有16位，所以double型数据的有效位数为15/16。第16位不一定有效，因为可能会有四舍五入或精度损失的问题；
由于double的指数部分对应的指数范围为-1023~1023，所以取值范围为： -2^1023到2^1023，约等于-1.7E308 —+1.7E308；

1.4551915228385041700676083564758300781 25*2^36=100000000000.125(15位有效数字)



1.8189894035460838495055213570594787597656 25*2^39=1000000000000.125(16位有效数字)

```
1000000000000.125
双精度浮点数机内表示：01000010 01101101 00011010 10010100 10100010 00000000 00000100 00000000
符号位是：0
指数（阶码）是：100 0010 0110
指数十进制是：1062
指数真实值是：39
尾数是：1101 0001 1010 1001 0100 1010 0010 0000 0000 0000 0100 0000 0000
尾数十进制是：0.8189894035460838495055213570594787597656 25
尾数真实值是：1.8189894035460838495055213570594787597656 25
请按任意键继续
```

§.基础知识题 - 浮点数机内存储格式(IEEE 754)理解

4、思考

(1) 8/11bit的指数的表示形式是2进制补码吗？如果不是，一般称为什么方式表示?

不是，这种叫做移码

(2) double赋值给float时，下面两个程序，double型常量不加F的情况下，左侧有warning，右侧无warning，为什么?

总结一下规律

因为通过例题知道，1.2若是使用float形式储存，会导致产生一定的误差。这是因为float的7位精度导致的，如果用double可能可以更精确，于是把这个1.2的double型赋值给float时就会warning发生截断损失精确度。右侧的100.25用float表示则没有误差。

规律：若float表示的小数出现误差，则该小数的double型赋值给float时就会出warning警告发生截断

```
#include <iostream>
using namespace std;
int main()
{
    float f = 1.2;
    unsigned char* p = (unsigned 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;
}
```

warning C4305: “初始化”: 从“double”到“float”截断

```
#include <iostream>
using namespace std;
int main()
{
    float f = 100.25;
    unsigned char* p = (unsigned 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;
}
```