

§ . 基础知识题 – 浮点数机内存存储格式(IEEE 754)理解



要求:

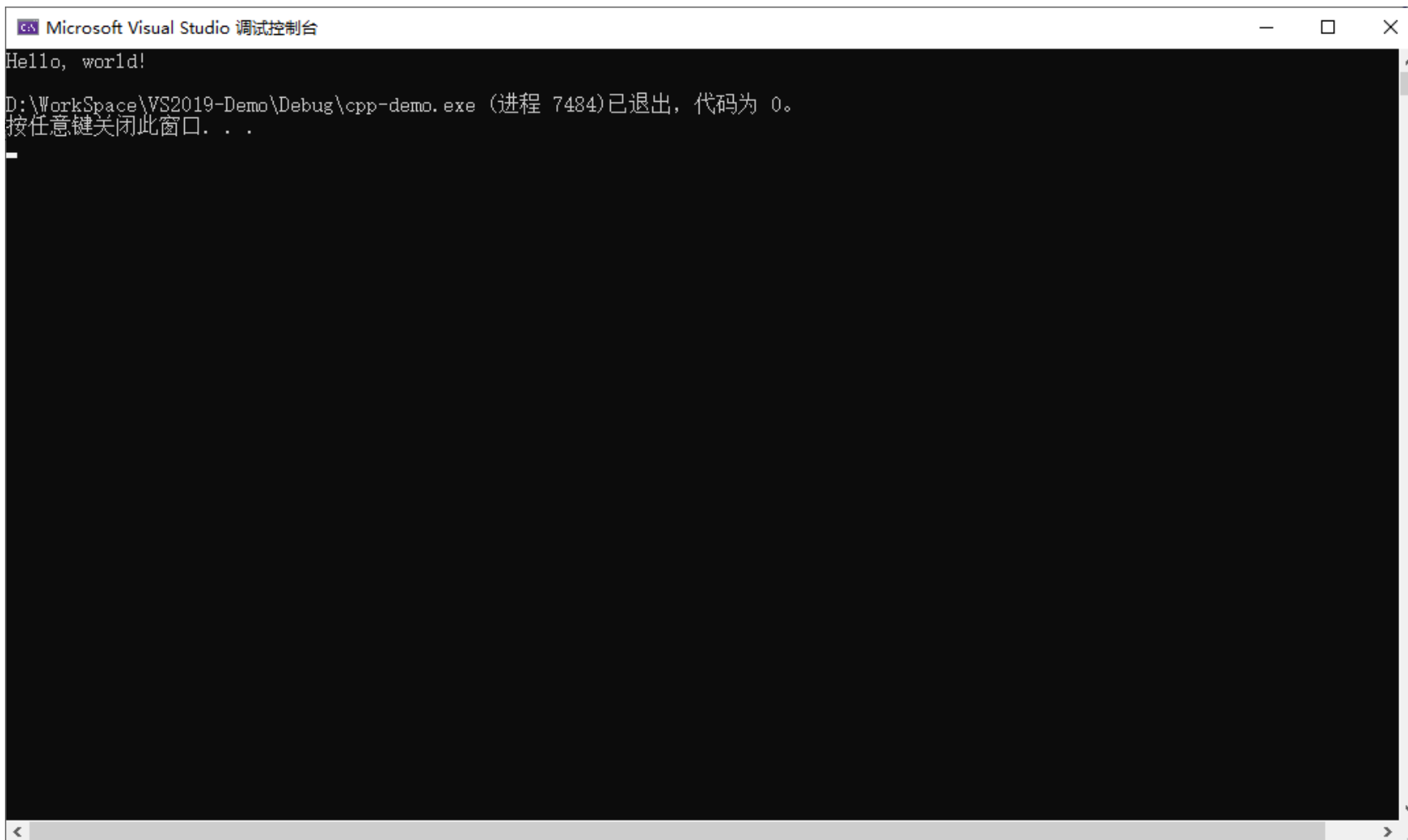
- 1、完成本文档中所有的题目并写出分析、运行结果
- 2、无特殊说明，均使用VS2022编译即可
- 3、直接在本文件上作答，**写出答案/截图（不允许手写、手写拍照截图）**即可；填写答案时，为适应所填内容或贴图，**允许调整**页面的字体大小、颜色、文本框的位置等
 - ★ 贴图要有效部分即可，不需要全部内容
 - ★ 在保证一页一题的前提下，具体页面布局可以自行发挥，简单易读即可
 - ★ **不允许**手写在纸上，再拍照贴图
 - ★ **允许**在各种软件工具上完成（不含手写），再截图贴图
- 4、转换为pdf后提交
- 5、**3月12日前**网上提交本次作业（在“文档作业”中提交）

§. 基础知识题 – 浮点数机内存储格式(IEEE 754)理解

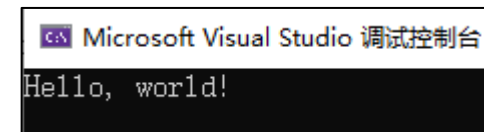


贴图要求：只需要截取输出窗口中的有效部分即可，如果全部截取/截取过大，则视为无效贴图

例：无效贴图

A screenshot of the Microsoft Visual Studio debug console window. The window is titled "Microsoft Visual Studio 调试控制台". The output text is: "Hello, world!", "D:\Workspace\VS2019-Demo\Debug\cpp-demo.exe (进程 7484)已退出, 代码为 0.", and "按任意键关闭此窗口. . .". The window is large, showing the full output area.

例：有效贴图

A screenshot of the Microsoft Visual Studio debug console window, cropped to show only the first line of output. The window is titled "Microsoft Visual Studio 调试控制台". The output text is: "Hello, world!".



§. 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

```
demo.cpp
demo-cpp (全局范围) main()
1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     cout << "Hello, 同济!" << endl;
6     return 0;
7 }
8
```

100 % 未找到相关问题 行: 4 字符: 2 空格 SYS CR

输出

显示输出来源(S): 生成

生成开始于 22:23...

1>—— 已启动生成: 项目: demo-cpp, 配置: Debug Win32 ——

1>demo.cpp

1>D:\Workspace\VS2022-demo\demo-cpp\demo.cpp(1,1): warning C4335: 检测到 Mac 文件格式: 请将源文件转换为 DOS 格式或 UNIX 格式

1>D:\Workspace\VS2022-demo\demo-cpp\demo.cpp(1,10): warning C4067: 预处理器指令后有意外标记 - 应输入换行符

1>MSVCRTD.lib(exe_main.obj) : error LNK2019: 无法解析的外部符号 _main, 函数 "int __cdecl invoke_main(void)" (?invoke_main@YAHKZ) 中引用了该符号

1>D:\Workspace\VS2022-demo\Debug\demo-cpp.exe : fatal error LNK1120: 1 个无法解析的外部命令

1>已完成生成项目 "demo-cpp.vcxproj" 的操作 - 失败。

==== 生成: 0 成功, 1 失败, 0 最新, 0 已跳过 =====

==== 生成于 22:23 完成, 耗时 01.132 秒 =====

错误列表 输出

§. 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

Microsoft
79
e9
f6
42

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 0110 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/BV1iW41ld7hd?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×tamp=1662273598&unique_k=AuouMEO

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: 尾数的符号位是 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方法:

(1) 得到的32bit的机内表示是: 0011 1111 1001 1001 1001 1001 1010 (3f 99 99 9a)

(2) 其中: 尾数的符号位是 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 1010 (填32bit中的原始形式)

尾数转换为十进制小数形式是 0.2000000476837158203125 (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 1.2000000476837158203125 (加整数部分的1后)

001 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 + ... + 0.0000002384185791015625 (详见右侧蓝色) = 0.2000000476837158203125

=> 加1 = 1.2000000476837158203125 (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 +
0.00000095367431640625 +
0.0000002384185791015625

0.2000000476837158203125

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. 52545.54525 (此处设学号是1234567，需换成本人学号后5位，小数为学号后五位逆序，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 0111 0100 1101 0100 0001 1000 1100__ (不是手算，用P.4方式打印)

(2) 其中：尾数的符号位是__0__

指数是__1000 1110__ (填32bit中的原始形式)

指数转换为十进制形式是__142__ (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__15__ (32bit中的原始形式按IEEE754的规则转换)

尾数是__100 1101 0100 0001 1000 1100__ (填32bit中的原始形式)

尾数转换为十进制小数形式是__0.603562831878662109375__ (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__1.603562831878662109375__ (加整数部分的1)

注1：转换为十进制小数用附加的工具去做，精度足够；

自己去网上找工具也行，但找到工具要满足精度要求 (下同!!!)

注2：数据超过了float的精度范围，但P.4方式打印后不影响理解 (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. -54525.52545 (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 0111 0101 0100 1111 1101 1000 0111__ (不是手算，用P.4方式打印)

(2) 其中：尾数的符号位是_____1_____

指数是__1000 1110__ (填32bit中的原始形式)

指数转换为十进制形式是_____142_____ (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_____15_____ (32bit中的原始形式按IEEE754的规则转换)

尾数是__101 0100 1111 1101 1000 0111__ (填32bit中的原始形式)

尾数转换为十进制小数形式是__0.66398704051971435546875__ (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__1.66398704051971435546875__ (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. 0.0052545 (设学号为1234567，按规则更换为学号)

注：尾数为正、指数为负

(1) 得到的32bit的机内表示是：___0011 1011 1010 1100 0010 1101 1111 0001___ (不是手算，用P.4方式打印)

(2) 其中：尾数的符号位是___0___

指数是___0111 0111___ (填32bit中的原始形式)

指数转换为十进制形式是___119___ (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___-8___ (32bit中的原始形式按IEEE754的规则转换)

尾数是___010 1100 0010 1101 1111 0001___ (填32bit中的原始形式)

尾数转换为十进制小数形式是___0.34515202045440673828125___ (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___1.34515202045440673828125___ (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. -0.0054525 (设学号为1234567，按规则更换为学号逆序)

注：尾数为负、指数为负

(1) 得到的32bit的机内表示是：__1011 1011 1011 0010 1010 1010 1110 0011__ (不是手算，用P.4方式打印)

(2) 其中：尾数的符号位是____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____119____ (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____-8____ (32bit中的原始形式按IEEE754的规则转换)

尾数是__ 011 0010 1010 1010 1110 0011 ____ (填32bit中的原始形式)

尾数转换为十进制小数形式是_ 0.39584004878997802734375 ____ (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__ 1.39584004878997802734375 __ (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. 52545.54525 (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 0000 1110 1001 1010 1000 0011 0001 0111 0010 1011 0000 0010
0000 1100 0101___ (不是手算，用P.5方式打印)

(2) 其中：尾数的符号位是___0___

指数是___ 100 0000 1110 ___ (填64bit中的原始形式)

指数转换为十进制形式是___1038___ (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___15___ (64bit中的原始形式按IEEE754的规则转换)

尾数是___1001 1010 1000 0011 0001 0111 0010 1011 0000 0010 0000 1100 0101___ (填64bit中的原始形式)

尾数转换为十进制小数形式是___ 0.6035627822875977432914851306122727692127227783203125 ___ (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___ 1.6035627822875977432914851306122727692127227783203125 ___ (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. -54525.52545 (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 0000 1110 1010 1001 1111 1011 0000 1101 0000 0111 1100 1000 0100 1011 0110__ (不是手算，用P.5方式打印)

(2) 其中：尾数的符号位是__1__

指数是__100 0000 1110__ (填64bit中的原始形式)

指数转换为十进制形式是__1038__ (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__15__ (64bit中的原始形式按IEEE754的规则转换)

尾数是__1010 1001 1111 1011 0000 1101 0000 0111 1100 1000 0100 1011 0110__ (填64bit中的原始形式)

尾数转换为十进制小数形式是__0.663986982727050811803337637684307992458343505859375__ (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__1.663986982727050811803337637684307992458343505859375__ (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. 0.0052545 (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 0111 0101 1000 0101 1011 1110 0001 1010 1000 0010 0110 0010 0100 0101__ (不是手算，用P.5方式打印)

(2) 其中：尾数的符号位是____0____

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__ 0101 1000 0101 1011 1110 0001 1010 1000 0010 0110 0010 0100 0101 ____ (填64bit中的原始形式)

尾数转换为十进制小数形式是__ 0.3451519999999999033235553724807687103748321533203125 ____ (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__ 1.3451519999999999033235553724807687103748321533203125 ____ (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. -0.0054525 (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 0110 0101 0101 0101 1100 0101 0010 1110 0111 0010 1101 1010 0001__ (不是手算，用P.5方式打印)

(2) 其中：尾数的符号位是_____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__0110 0101 0101 0101 1100 0101 0010 1110 0111 0010 1101 1010 0001_____ (填64bit中的原始形式)

尾数转换为十进制小数形式是

__0.3958399999999999696598251830437220633029937744140625__ (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__1.3958399999999999696598251830437220633029937744140625_ (加整数部分的1)



§. 基础知识题 – 浮点数机内存格式(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？
- (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 3.4×10^{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？
- (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？
- (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 1.7×10^{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？
- (5) 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？

本页不用作答，自己能知道
正确答案即可
(属于考试知识点)

§. 基础知识题 – 浮点数机内存储格式(IEEE 754)理解



4、思考

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

```
#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;
}
```

答：因为1.2转换为二进制是无限循环小数，不能被float类型精确表示，编译器默认这是一个double型变量，将double赋值给float可能导致精度损失，所以有warning；而100.25可以被float型精确表示。规律就是编译器默认不带后缀说明的浮点数是double类型，如果这个数被二进制表示是无限循环小数就会在赋值给float变量时出现截断的warning。

