§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.2151294.4921512 （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：0100 1010 0000 0011 0100 1101 1111 1010

(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 0011 0100 1101 1111 1010(填32bit中的原始形式)
尾数转换为十进制整数形式是1.0258171558380127(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0.0258171558380127(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.-4921512.2151294 （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：1100 1010 1001 0110 0011 0001 0101 0000

(2) 其中：符号位是1

指数是1001 0101(填32bit中的原始形式)
指数转换为十进制形式是149(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22(32bit中的原始形式按IEEE754的规则转换)

尾数是001 0110 0011 0001 0101 0000(填32bit中的原始形式)
尾数转换为十进制整数形式是1.17337989807128906(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0.17337989807128906(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.0.002151294 （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：0011 1011 0000 1100 1111 1100 1011 1001

(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000 1100 1111 1100 1011 1001(填32bit中的原始形式)
尾数转换为十进制整数形式是1.1014624834060669(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0.1014624834060669(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.-0.004921512 （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：1011 1011 1010 0001 0100 0100 1010 0011

(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 0001 0100 0100 1010 0011(填32bit中的原始形式)
尾数转换为十进制整数形式是1.2599071264266968(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0.2599071264266968(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.2151294.4921512 （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：0100 0001 0100 0000 0110 1001 1011 1111 0011 1110 1111 1110 1100 1111 0111 1110

(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0000 0110 1001 1011 1111 0011 1110 1111 1110 1100 1111 0111 1110(填64bit中的原始形式)
尾数转换为十进制整数形式是1.02581715209541313(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0.02581715209541313(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.-4921512.2151294 （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：1100 0001 0101 0010 1100 0110 0010 1010 0000 1101 1100 0100 1010 1110 0001 1010

(2) 其中：符号位是1

指数是100 0001 0101(填64bit中的原始形式)
指数转换为十进制形式是1045(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22(64bit中的原始形式按IEEE754的规则转换)

尾数是0010 1100 0110 0010 1010 0000 1101 1100 0100 1010 1110 0001 1010(填64bit中的原始形式)
尾数转换为十进制整数形式是1.17337994936213486(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0.17337994936213486(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.0.002151294 （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：0011 1111 0110 0001 1001 1111 1001 0111 0010 1011 1111 1000 0111 1000 1100 0101

(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0001 1001 1111 1001 0111 0010 1011 1111 1000 0111 1000 1100 0101(填64bit中的原始形式)
尾数转换为十进制整数形式是1.10146252800000011(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0.10146252800000011(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.-0.007654321 （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：1011 1111 0111 0100 0010 1000 1001 0100 0101 0001 0110 0011 1101 0011 0110 1101

(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0100 0010 1000 1001 0100 0101 0001 0110 0011 1101 0011 0110 1101(填64bit中的原始形式)
尾数转换为十进制整数形式是1.2599070720000001(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0.2599070720000001(保留小数点后)

§6.指针基础 - 浮点数机内存储格式(IEEE 754)理解

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

XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXX

float型数据=数符 + 1.尾数 * 2的指数次方

第1位：数符：尾数正负

第2-9位：指数：2的次方

第10-32位：尾数：小数部分

尾数的正负如何表示?

用数符位表示：0，尾数为正；1，尾数为负。

尾数如何表示?

将float型数据转换成小数 * 2的n次方的形式，将小数的小数部分转换成二进制，取前23位（0舍1入）成为尾数

指数的正负如何表示?

用指数的第一位（数据第2位）表示：0，指数为正；1，指数为负

指数如何表示?

将float型数据转换成小数 * 2的n次方的形式，将（127 + 2的次数n）转换成二进制成为指数

§6.指针基础 - 浮点数机内存储格式(IEEE 754)理解

4、总结

(2) 为什么float型数据只有7位十进制有效数字?

尾数（23位）以及整数部分的1（1位）共24位

可以表示的最大数为2^24-1=16777215，共8位

但第8位也可能不准确，故有效数字为7位

为什么最大只能是$3.4x10^{38}$ ？

尾数最大：111 1111 1111 1111 1111 1111

指数最大：由于float型数在指数为256时，认为不是数字（NaN）

故最大指数为255：1111 1111

最大数：$(1.11111111111111111111111)_2 * 2^{255-127} \approx 3.4*10^{38}$

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?

6位：0.8765433f 存储为：0.8765432835

7位：0.0078125f 存储为：0.0078125000

§6.指针基础 - 浮点数机内存储格式(IEEE 754)理解

4、总结

(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

XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX

double型数据=数符 + 1.尾数 * 2的指数次方

第1位：数符：尾数正负

第2-12位：指数：2的次方

第13-64位：尾数：小数部分

尾数的正负如何表示?

用数符位表示：0，尾数为正；1，尾数为负。

尾数如何表示?

将double型数据转换成小数 * 2的n次方的形式，将小数的小数部分转换成二进制，取前52位（0舍1入）成为尾数

指数的正负如何表示?

用指数的第一位（数据第2位）表示：0，指数为正；1，指数为负

指数如何表示?

将double型数据转换成小数 * 2的n次方的形式，将（1023 + 2的次数n）转换成二进制成为指数

§6.指针基础 - 浮点数机内存储格式(IEEE 754)理解

4、总结

(4) 为什么double型数据只有15位十进制有效数字?

尾数（52位）以及整数部分的1（1位）共53位

可以表示的最大数为2^53-1=9007,1992,5474,0991，共16位

但第16位也可能不准确，故有效数字为15位

为什么最大只能是$1.7x10^{308}$ ？

尾数最大：1111 11111111 11111111 11111111 11111111 11111111 11111111

指数最大：由于double型数在指数为2048时，认为不是数字（NaN）

故最大指数为2047：111 11111111

最大数：$(1.1111111111111111111111111111111111111111111111111111)_2 * 2^{2047-1023} \approx 1.7*10^{308}$

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？

15位：0.9876543210987620 存储为：0.9876543210987619448

16位：0.0000152587890625 存储为：0.0000152587890625000