

姓名:王哲源 班级:计算机一班 学号:1652228

补充:

13、 结合课件“15162-070005.第07章(补4)第13章 输入输出流.pdf”的P.27,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?

Float 型数据的 32bits 中, 第一位 1bit 为符号位 s , 0 为正, 1 为负; 第二位到第九位的 8bits 为指数位, 其为一个无符号二进制数 E , 被称作阶码, 范围从 0-255 (当 E 为 0 且尾数为 0 时, 表示 0 (但并非真正的 0 而是一个极小数); 当 E 为 255 时尾数为 0 表示无穷, 尾数非 0 表示 NaN), 阶码的获取方式为对指数 e 的补码符号位取反后-1 (又被称作移码); 第十位到第三十二位共 23bits 为尾数部分, 其为一个无符号二进制数 M , 表示的数为当数字转为二进制小数形式之后将小数点移至第一位非零位右侧时小数点后所表示的二进制数 (不足用 0 补齐多余则根据情况舍入: 若多余部分超过一半 (如 101) 则进位, 不足一半 (如 011) 则舍去, 若刚好为一半 (如 100) 则看前一位, 为 1 进位为 0 舍去)。通过 float 型计算原数时的公式为 $x = (-1)^s * (1.M) * 2^e$, 其中指数 $e = E - 127$

- (2) 为什么 float 型数据只有 7 位有效数字? 为什么最大只能是 3.4×10^{38} ?

根据 (1) 中所提, M 最多只有 23bits, 多出的部分将会被丢失, 而 $2^{23} - 1 = 8388607$, 即 M 的范围只有 0-8388607, 因此 float 型数据只有 7 位有效数字。而当 float 取最大值时, 其值约为 $(2 - 2^{-23}) * 2^{127} \approx 3.4 * 10^{38}$

- (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?

Float 型数据的 64bit 中, 第一位 1bit 为符号位 s , 0 为正, 1 为负; 第二位到第十二位的 11bits 为指数位, 其为一个无符号二进制数 E , 范围从 0-2047 (当 E 为 0 且尾数为 0 时, 表示 0 (但并非真正的 0 而是一个极小数); 当 E 为 2047 时尾数为 0 表示无穷, 尾数非 0 表示 NaN); 第十三位到第六十四位共 52bits 为尾数部分, 其为一个有符号二进制数 M , 表示的数为当数字转为二进制小数形式之后将小数点移至第一位非零位右侧时小数点后所表示的二进制数 (舍入同 float)。通过 double 型计算原数时的公式为 $x = (-1)^s * (1.M) * 2^e$, 其中指数 $e = E - 1023$

- (4) 为什么 double 型数据有 15 位有效数字? 为什么最大是 1.7×10^{308} ?

根据 (1) 中所提, M 最多只有 52bits, 多出的部分将会被丢失, 而 $2^{52} - 1 = 4503599627370495$, 即 M 的范围只有 0-4503599627370495, 因此 double 型数据只有 15 位有效数字可信。而当 double 取最大值时, 其值约为 $(2 - 2^{-52}) * 2^{1023} \approx 1.7 * 10^{308}$

- (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 (写出二进制表示时, 每 8bit 加一个“-”方便对齐查看)

a) float d=123.456

$$(123.456)_{10} = (1111011.0111010010111100011010100111111)_2$$

符号位 $s=0$, 指数 $e=6$, 规格化后尾数为 1.11101101110100101111001

故尾数域为 11101101110100101111001

阶码为 10000101

故最终存储结果为 01000010-11110110-11101001-01111001

b) float d=-123.456

$$(-123.456)_{10} = (-1111011.0111010010111100011010100111111)_2$$

符号位 $s=1$, 指数 $e=6$, 规格化后尾数为 1.11101101110100101111001

故尾数域为 11101101110100101111001

阶码为 10000101

故最终存储结果为 11000010-11110110-11101001-01111001

c) float d=0.123e-3

$$(0.123 * 10^{-3})_{10} = (0.0000000000001000000011111001100011111010)_2$$

符号位 s=0,指数 e=-13,规格化后尾数为 1.00000001111100110010000

故尾数域为 00000001111100110010000

阶码为 01110010

故最终存储结果为 00111001-00000000-11111001-10010000

d) float d=-1.23e-4

$$(-1.23 * 10^{-4})_{10} = (-0.0000000000001000000011111001100011111010)_2$$

符号位 s=1,指数 e=-13,规格化后尾数为 1.00000001111100110010000

故尾数域为 00000001111100110010000

阶码为 01110010

故最终存储结果为 10111001-00000000-11111001-10010000

e) double d=123.456

$$(123.456)_{10}$$

$$= (1111011.011101001011110001101010011111101111100111011011001001)_2$$

符号位 s=0,指数 e=6,

规格化后尾数为 1.1110110111010010111100011010100111111011111001110111

故尾数域为 1110110111010010111100011010100111111011111001110111

阶码为 10000000101

故最终存储结果为

01000000-01011110-11011101-00101111-00011010-10011111-10111110-01110111

f) double d=-123.456

$$(-123.456)_{10}$$

$$= (-1111011.011101001011110001101010011111101111100111011011001001)_2$$

符号位 s=1,指数 e=6,

规格化后尾数为 1.1110110111010010111100011010100111111011111001110111

故尾数域为 1110110111010010111100011010100111111011111001110111

阶码为 10000000101

故最终存储结果为

11000000-01011110-11011101-00101111-00011010-10011111-10111110-01110111

g) double d=0.123e-3

$$(0.123 * 10^{-3})_{10}$$

$$= (0.0000000000001000000011111001100011111010001101110110100100100011000)_2$$

符号位 s=0,指数 e=-13,

规格化后尾数为 1.00000001111100110001111101000110111011010010010001100

故尾数域为 0000000111110011000111110100011011101101001001000110

阶码为 01111110010

故最终存储结果为

00111111-00100000-00011111-00110001-11110100-01101110-11010010-01000110

h) double d=-1.23e-4

$$(-0.123 * 10^{-3})_{10}$$

$$= (-0.0000000000001000000011111001100011111010001101110110100100100011000)_2$$

符号位 s=1,指数 e=-13,

规格化后尾数为 1.00000001111100110001111101000110111011010010010001100
故尾数域为 0000000111110011000111110100011011101101001001000110
阶码为 01111110010
故最终存储结果为
10111111-00100000-00011111-00110001-11110100-01101110-11010010-01000110

【作业要求:】

- 1、**4月5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
- 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