

<朱世轩 计 2 1752528>

补充:

13、 结合课件“17182-070009. 第 07 章(补 8) 第 13 章 输入输出流(含 C 方式的文件操作). pdf”的 P.13, 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 32bit 用最高位表示符号位 (二进制 0 代表非负, 1 代表负), 接下来 8 位表示指数 (余码方式, 将指数加 127 存储), 剩余 23 位表示尾数。尾数的正负由最高位表示, 用指数域中表示的值减去 127, 得到的就是指数, 结果如果是正, 指数就为正, 如果是负, 指数就为负。

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

指数域表示数值只有 8 位, 确定的指数在 -128 到 127 之间, 所以, 一个 float 型数的最大值为  $+2^{127}$  即  $3.4 \times 10^{38}$ 。

- (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?

double 型数据的 64bit 用最高位表示符号位 (二进制 0 代表非负, 1 代表负), 接下来 11 位表示指数 (余码方式, 将指数加 1023 存储), 剩余 52 位表示尾数。尾数的正负由最高位表示, 用指数域中表示的值减去 1023, 得到的就是指数, 结果如果是正, 指数就为正, 如果是负, 指数就为负。

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

指数域表示数值只有 11 位, 确定的指数在 -1024 到 1023 之间, 所以, 一个 double 型数的最大值为  $+2^{1023}$  即  $1.7 \times 10^{308}$ 。

- (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 (写二进制表示时, 每 8bit 加 1 个 “-” 方便查看, 例: 00100000-01010001)

a) float d=123.456

01000010-11110110-11101001-01111001

0: 符号位 表示正数

10000101: 指数域 133 指数  $133-127=6$

11101101110100101111001: 尾数域 0.11101101110100101111001

则 123.456 表示为 1.11101101110100101111001 小数点右移 6 位, 即

1111011.01110100101111001

b) float d=-123.456

11000010-11110110-11101001-01111001

1: 符号位 表示负数

10000101: 指数域 133 指数  $133-127=6$

11101101110100101111001: 尾数域 0.11101101110100101111001

则 -123.456 表示为 1.11101101110100101111001 小数点右移 6 位, 即

-1111011.01110100101111001

c) float d=0.123e-3

00111001-00000000-11111001-10010000

0: 符号位 表示正数

01110010: 指数域 114 指数  $114-127=-13$

00000001111100110010000: 尾数域 0.00000001111100110010000

则 0.123e-3 表示为 1.00000001111100110010000 小数点左移 13 位, 即

0.000000000000100000001111100110010000

d) float d=-1.23e-4

10111001-00000000-11111001-10010000

1:符号位 表示负数

01110010:指数域 114 指数 114-127=-13

00000001111100110010000:尾数域 0.00000001111100110010000

则-1.23e-4 表示为-1.00000001111100110010000 小数点左移 13 位, 即

-0.000000000000100000001111100110010000

e) double d=123.456

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

0:符号位 表示正数

10000000101:指数域 指数 1029-1023=6

1110110111010010111100011010100111111011111001110111:尾数域

0.1110110111010010111100011010100111111011111001110111

则 123.456 表示为 1.1110110111010010111100011010100111111011111001110111

小数点右移 6 位, 即

1111011.0111010010111100011010100111111011111001110111

f) double d=-123.456

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

1:符号位 表示负数

10000000101:指数域 指数 1029-1023=6

1110110111010010111100011010100111111011111001110111:尾数域

0.1110110111010010111100011010100111111011111001110111

则-123.456 表示为-1.1110110111010010111100011010100111111011111001110111

小数点右移 6 位, 即

-1111011.0111010010111100011010100111111011111001110111

g) double d=0.123e-3

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

0:符号位 表示正数

01111110010:指数域 指数 1010-1023=-13

0000000111110011000111110100011011101101001001000110:尾数域

0.0000000111110011000111110100011011101101001001000110

则 0.123e-3 表示为 1.0000000111110011000111110100011011101101001001000110

小数点左移 13 位, 即

0.00000000000010000000111110011000111110100011011101101001001000110

h) double d=-1.23e-4

10111111-00100000-00011111-00110001-11110100-01101110-11010010-01000110

1:符号位 表示负数

01111110010:指数域 指数 1010-1023=-13

0000000111110011000111110100011011101101001001000110:尾数域

0.0000000111110011000111110100011011101101001001000110

则-1.23e-4 表示为 1.0000000111110011000111110100011011101101001001000110

小数点左移 13 位, 即

-0.00000000000010000000111110011000111110100011011101101001001000110

**【作业要求:】**

- 1、**4月10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