浮点数在内存中的存储方式为：符号位，指数，尾数

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 符号位 | 指数 | 尾数 |
| float | 1位（第31位） | 8位  （第23-30位） | 23位  （第0-22位） |
| double | 1位（第63位） | 11位  （第52-62位） | 52位  （第0-51位） |

float与double类型的数据在计算机内部的表示法是相同的，但由于所占存储空间的不同，其分别能够表示的数值范围和精度不同。

浮点数的转换

1. 将浮点数转换成二进制
2. 用科学计数法表示二进制浮点数
3. 计算指数偏移后的值

注意：计算指数时需要加上偏移量，而偏移量的值与类型有关

指数6：

* float：127 + 6 -> 133
* double：1023 + 6 -> 1029

10进制浮点数的内存表示

实数8.25在内存中的float表示：

* 8.25的二进制表示：1000.01 -> 1.00001\*(2^3)
  + 符号位：0
  + 指数127 + 3 -> 130 -> 10000010
  + 小数：00001

内存中8.25的float表示：

0 10000010 00001000000000000000000 -> 0x41040000

注：十进制小数转换成二进制小数采用"乘2取整，顺序排列"法

十进制整数转换为二进制整数采用"除2取余，逆序排列"法

有趣的问题

int类型的范围：[, ]

float类型的范围：[]

思考：int和float都占4个字节的内存，为什么float却比int的范围大得多呢？

浮点类型的秘密：

* float能表示的具体数字的个数与int相同
* float可表示的数组之间不是连续的，存在间隙
* float只是一种近似的表示法，不能作为精确数使用
* 由于内存表示法相对复杂，float的运算速度比int慢得多

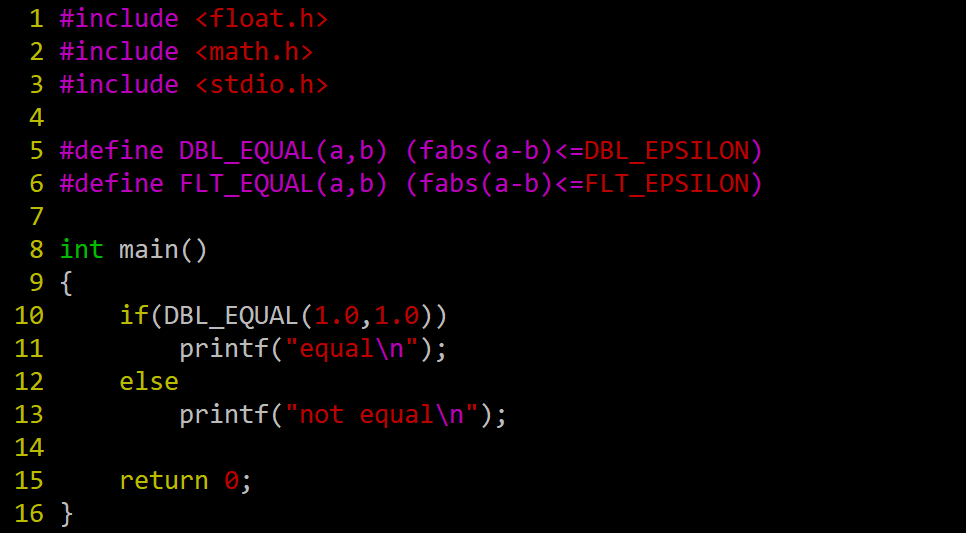
注意：double与float具有相同的内存表示法，因此double也是不精确的，由于double占用的内存较多，所能表示的精度比float高

浮点型变量不能直接和0进行比较，需要定义精度

float：#define EPSINON 0.00000001

-EPSINON<= 0.0 && 0.0 <= EPSINON

float.h头文件中定义了float，double两种浮点数的误差值，可以直接使用



https://github.com/penghuidong/tinySTL