

# 浮点数表示方法报告

18081811-高炜哲

f1 f2 f3 实际上是 3 个不一样但是十分接近的浮点数，由于计算机的浮点数为离散量且表示精度不足，所以根据“就近原则”，f2 无法准确表示，就近表示为 f1，所以 f1, f2 打印均为 217:16777216.000000，而 f3 可以准确表示为 217:16777218.000000。

```
Loading personal and system profiles took 528ms.
GAO@DESKTOP-3F4TKM1 Assignment 2 0 ?main ↑1 100%
f1 16,777,216:16777216.000000
f2 16,777,217:16777216.000000
f3 16,777,218:16777218.000000
f1=f2?: true
GAO@DESKTOP-3F4TKM1 Assignment 2 0 ?main ↑1 100%
```

a 的除数为 0，而 float, double 中存在一类特别的数：-1.#IND，IND 其实就是 Implementation Not define，这个是未定义或者他不是个数。同理，b 为 -1 的开方，1.QNAN0 其实就是 NaN, 即 Not a Number，因此他同样不是一个数。

```
GAO@DESKTOP-3F4TKM1 Assignment 2 0 ?main ↑1 ~1 ?2 100%
a=-1.#IND00 b=1.#QNAN0
GAO@DESKTOP-3F4TKM1 Assignment 2 0 ?main ↑1 ~1 ?2 100%
```

结构体中的 float f 均不赋值，默认为 0，所以 t1.f == t2.f; t1.i = 0x80000000，二进制补码为 1000 0000 0000 0000 0000 0000 0000 0000，为最小负数-2147483648，而 t2.i = 0x00000000 = 0，所以两者并不相等。

```
GAO@DESKTOP-3F4TKM1 Assignment 2 0 ?main ↑1 ~3 ?3 100%
t1.f=0
t2.f=0
float data is equal
t1.i=-2147483648
t2.i=0
int data is not equal
GAO@DESKTOP-3F4TKM1 Assignment 2 0 ?main ↑1 ~3 ?3 100%
```

以 0.1 为例，0.1 表示为二进制时实际上为 0.0001100110011...无限循环小数，截断并将其转化为 10 进制数为

0.1000000000000000055511151231257827021181583404541015625, 与其最接近的 17 位小数即为 0.10000000000000001。b, c, d 三个小数同理, 显然 c 不等于 d 且  $c \leq d$ 。

```
GAO@DESKTOP-3F4TKM1 ■ Assignment 2 0 ?main ↑1 ~5 ?3 100%
a(0.1):0.100000000000000001
b(0.2):0.200000000000000001
c(0.3):0.29999999999999999
d(0.4):0.30000000000000004
c==d?false
c<d>true
```

C 语言中，float 是单精度浮点数，而 double 是双精度浮点数，所以 Double 的精度是 Float 的两倍。所以由于精度不同的原因，我将 d 的值分别以 double 格式输出，可见在 float 下，**3/1.1=3.000...**，而 double 下，**3/1.1=2.999...6**，所以变作 int 时小数点后面截断，分别位 3 和 2。

对于 d 同理, float 下为 3.000..., double 下为 2.999...6, 所以 3/1.1 在 float 下等于 3, 在 double 精度下则不等于。

[illegible]

(3.14+1e20)-1e20: 由于 1e20 远远大于 3.14, 因此前两者相加时精度不足, 所以 **3.14+1e20** 被记为 **1e20**, 所以  $1e20 - 1e20 = 0$ ;

```
GAO@DESKTOP-3F4TKM1 Assignment 2 100%
(3.14+1e20)-1e20 = 0.000000
3.14+(1e20-1e20) = 3.140000
```

前 3 行是输出内存地址，t1.f 和 t2.f 值为该内存地址 7F000000 里的数值，相加时候结果正溢。

[illegible]