****

**计算机组成原理实验报告**

**题目：**整数和浮点数的表示

**姓 名： 段欣然**

**专 业： 计算机科学与技术**

**年 级： 2020级**

**学 号： 202011081033**

**任课教师： 王志春**

**完成日期： 2022年4月1日**

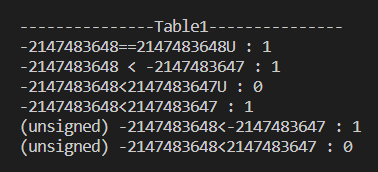
**人工智能学院**

1. 实验要求
2. 通过无符号数和带符号整数之间的相互转换来理解无符号数和补码整数的表示；
3. 掌握 IEEE754 浮点数在机器中的表示；
4. 掌握高级语言中数据类型的转换和移位操作结果，更好地理解指令系统设计和计算机硬件设计需要考虑的因素。
5. 实验结果与分析

## 有符号数和无符号数的比较

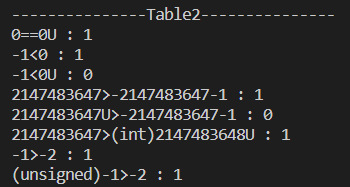
1. 验证表1关系表达式结果





编程输出结果与表1记录结果一致。

1. 编程得出表2的结果

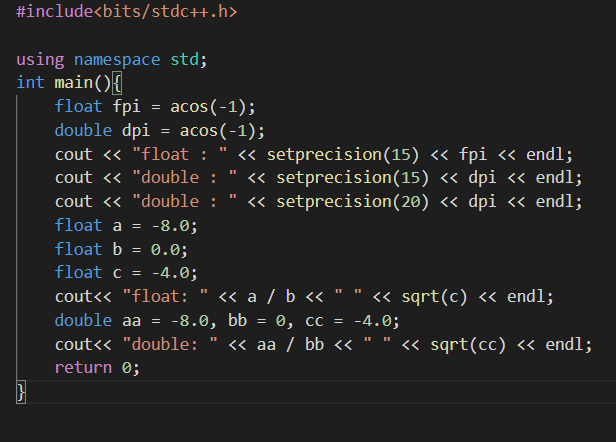


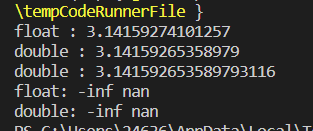
记录并说明结果见下表

|  |  |  |  |
| --- | --- | --- | --- |
| 关系表达式 | 运算类型 | 结果 | 说明 |
| 0==0U | 无符号整数 | 1 |  |
| -1<0 | 有符号整数 | 1 |  |
| -1<0U | 无符号整数 | 0 |  |
| 2147483647>-2147483647-1 | 有符号整数 | 1 |  |
| 2147483647U>-2147483647-1 | 无符号整数 | 0 |  |
| 2147483647>(int)2147483648U | 有符号整数 | 1 |  |
| -1>-2 | 有符号整数 | 1 |  |
| (unsigned)-1>-2 | 无符号整数 | 1 |  |

## 验证 float 和 double 类型的精度

程序代码及运行结果如下：





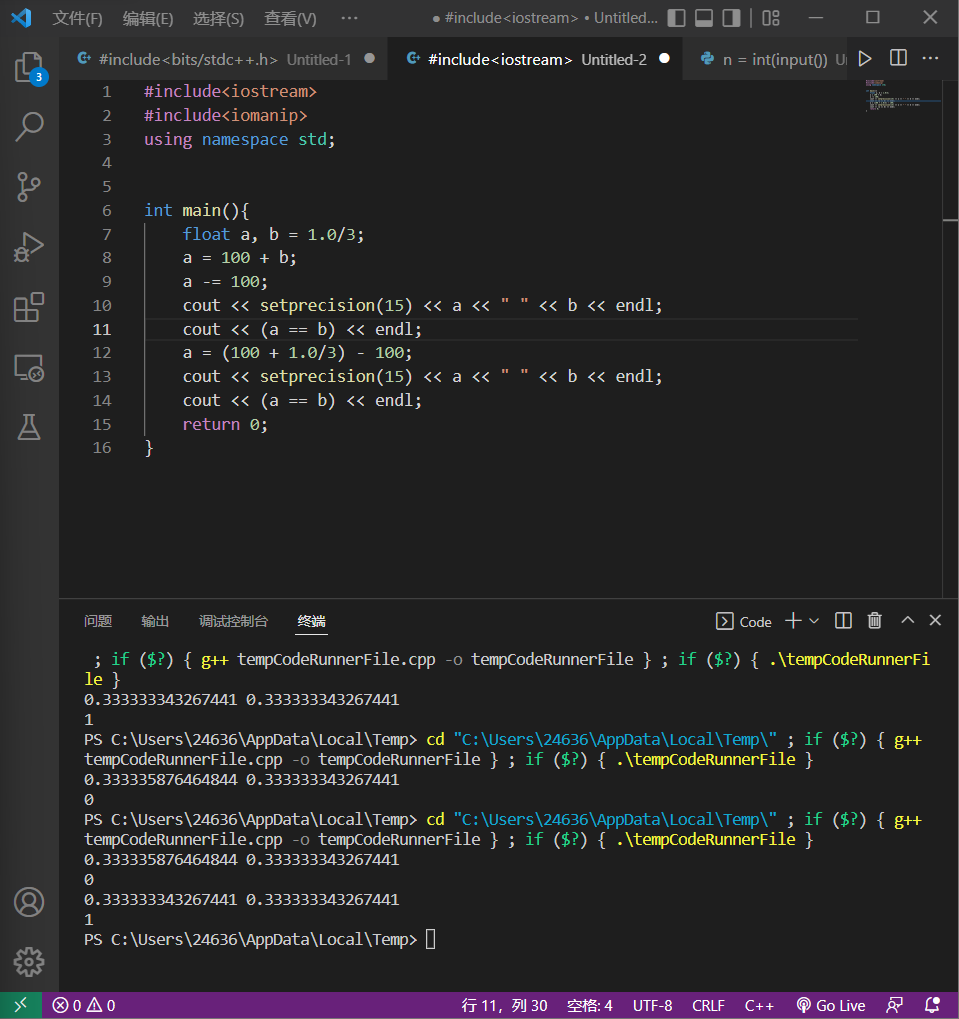
从结果可以看出float类型只有6-7位有效数字，后面的数就不准确了，因为float类型浮点数的尾数位数为位，，所以最大可以表示7位有效数字，一定能准确表示6位有效数字。，故double类型的精度为15-16位有效数字。

运行浮点数除法“-8.0/0”并不会报错，而是输出，这是因为浮点数的表示为，实际上是一个接近的数，所以结果时负无穷。而sqrt(-4.0)的结果与正常对负数进行开方操作的结果一致（），因为浮点数的表示方式仍有符号位。

## 验证逻辑表达式结果

float a = (100+1.0/3) - 100; b=1.0/3;

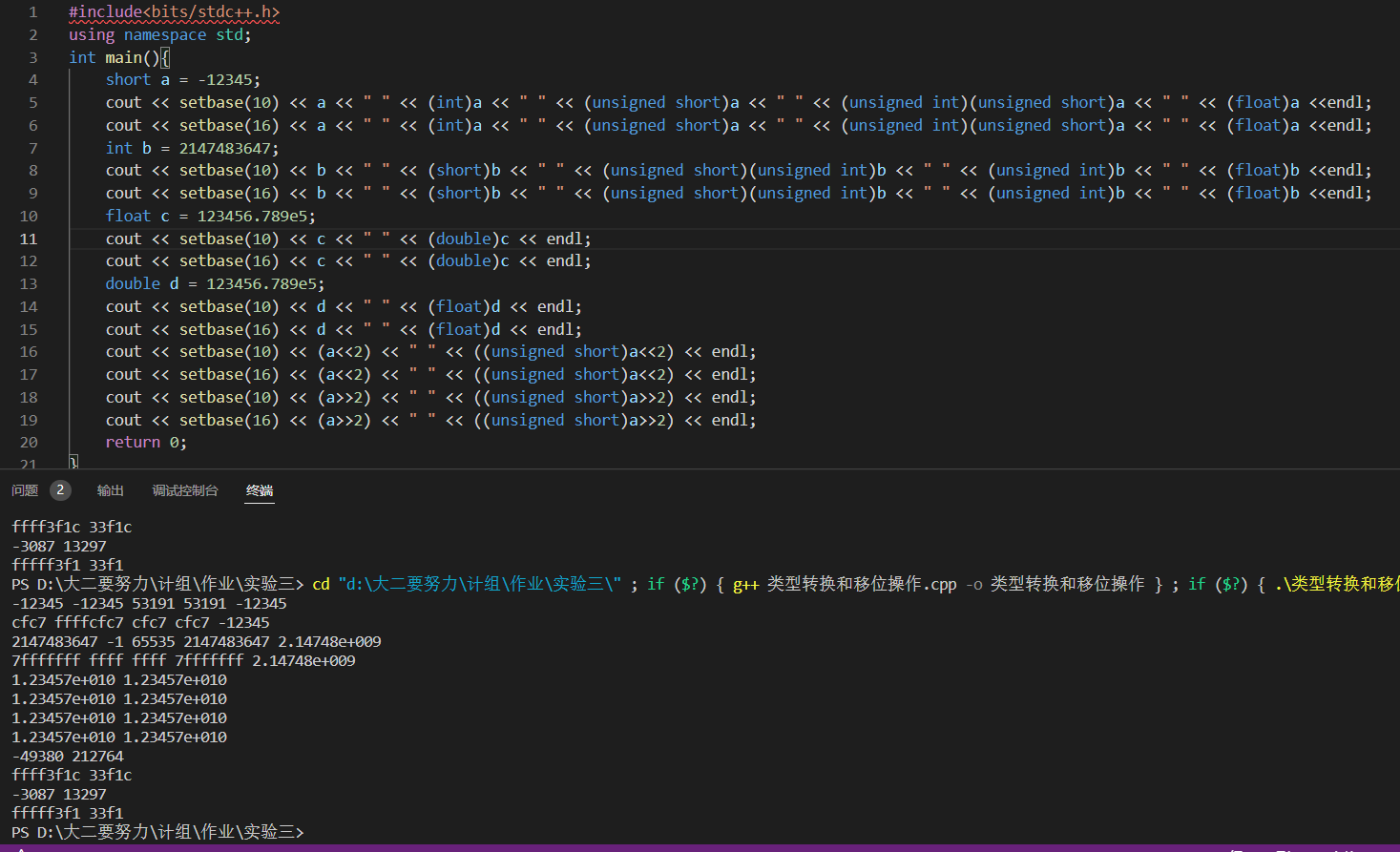
直接用表达式(100+1.0/3) – 100给a赋值则a==b为真，运行程序结果如下。



在程序中第一个a==b的取值为0，因为浮点数与int类型相加int型整数转变为浮点数进行计算，所以实际是对进行计算。两数阶数不同，先统一阶数，令右移位变成，此时原本的数据损失了最后位。然后进行加法，得到。之后再进行减法依然无法将损失的位二进制数（bit）补回来，故此时.

程序中第二个a==b取值为1，虽然与上述操作类似，但是gcc编译时会对源程序有优化作用，实际上相当于a=1.0/3，故.

## 类型转换和移位操作



观察结果可知**无符号数和带符号数扩展方式不同**，无符号数的扩展直接在原数之前补0；带符号数对正数（符号位为0）的扩展直接在原数据之前补0，对负数（符号位为1）的扩展直接在元数据之前补0。

**有符号整数并非总能转换为等值的float类型数据**。由上述[2](#_验证_float_和)知由于尾数位数限制，float类型精度为6-7位，当整数位数位时，可能出现丢失低位数据的情况，例如上述实验中位数严格大于7，必然出现误差，转换为float结果为.

**float型数据能转换为等值的double型数据**，因为double型数据精度均高于float。

对于有符号数和无符号数，**长数被截断成短数后可能发生数据丢失**，例如由int型转化为unsigned short型时由于截断部分数据变成.

**长数被截断成短数后可能发生溢出**，因为原本的正常数位截断后可能变成符号位，例如实验中2147482647由int型转化为short型时由于符号位是原数据二进制的第17位（由高到低），故原本的数由正数变成负数发生溢出。

C 语言中**移位操作规则与操作对象的数据类型有关**，有符号数保留符号位，对其他数位进行移位操作，无符号数直接整体进行移位操作。

由上图可知，左移 2 位和右移 2 位操作分别相当于扩大和缩小4倍。

1. 实验小结

实验中发现将short类型转换为unsigned int时，直接进行转换和先将short转换为unsigned short再进行扩展的结果可能不同。因此可以得到结论short类型转换为unsigned int型时隐含先转换为int再转换为unsigned int的两步操作，本实验为探究有无符号整数扩展方式，故采用先转换为unsigned short再进行扩展操作，例如，对short类型变量b做上述操作。



不同结果如下，该程序非实验要求，故不包含在源程序文件夹内。

