**计算机程序设计基础（C++)**

**实验报告**

专业班级：

学 号：

姓 名：

**实验报告成绩：**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **实验** | **实验一** | **实验二** | **实验三** | **实验四** | **实验五** | **总评** |
| **成绩** |  |  |  |  |  |  |

**批阅教师：**

**实验一、实验环境与简单程序设计**

**一、实验目的**

1、掌握集成开发环境，掌握C++程序的基本要素以及完整的C++程序开发过程。

2、掌握基本数据类型、运算符和表达式的使用。理解隐式转换和强制转换，理解数据超过该数据类型

表示范围时的溢出。掌握不同数据之间的混合算术运算中数据类型的转换。

3、变量的定义与常量的使用。

4、输入、输出的实现。

5、编译信息的理解与错误的修改。

6、简单程序的设计。

**二、实验内容**

熟悉C++编程环境，可以使用VS；对已经能熟练掌握C++开发环境的同学，可

以跳过本部分内容）

**1.编辑输入下列程序，找出下面代码的错误并改正：**

#include<iostream>

using namespace std;

int main()

{

Int i = k + 1;

cout << i++ << endl;

int i = 1;

cout << i++ << endl;

cout << "Welcome to C++"<<endl;

return 0;

}

**2.求圆锥的体积：要求键盘输入圆锥底的半径、锥高，使用标识符常量定义圆周率。**

(1)创建一个控制台项目

(2)在文件中输入程序内容，存盘

(3)编译、连接、运行；观察结果

**3**.**通过下面程序验证你所使用系统上运行的C++编译器中每个基本数据类型的长度。**

#include<iostream>

using namespace std;

int main()

{

cout << "char length:" << sizeof(char) << endl;

cout << "int length:" << sizeof(int) << endl;

}

**4.观察下面程序的执行结果。**

#include <iostream>

#include <iomanip>

using namespace std;

int main()

unsigned int testUnint=65534;//oxfffe

cout << "output in unsigned int 1 type:"" << testUnint<< end;//<<oct;

cout << "output in char type:!" << static\_ cast<char>(testUnint)<< endl;

cout << "output in short type:" << static\_ cast<short>(testUnint)<< endl;//为什么结果为-2?:

cout << "output in int type:" << static\_ cast<int>. (testUnint)<< endl;

cout << "output in double type:"<< static cast<double>(testUnint)<< endl;

cout << "output in double type:" <<setprecision(4)<< static\_ cast<double>(testUnint)<< endl;

cout << "output in Hex unsigned int type:" <<hex<< testUnint<< endl; //16进制输出

system("pause");

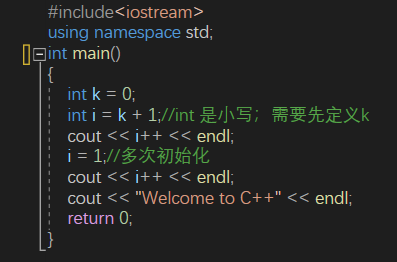
return 0;

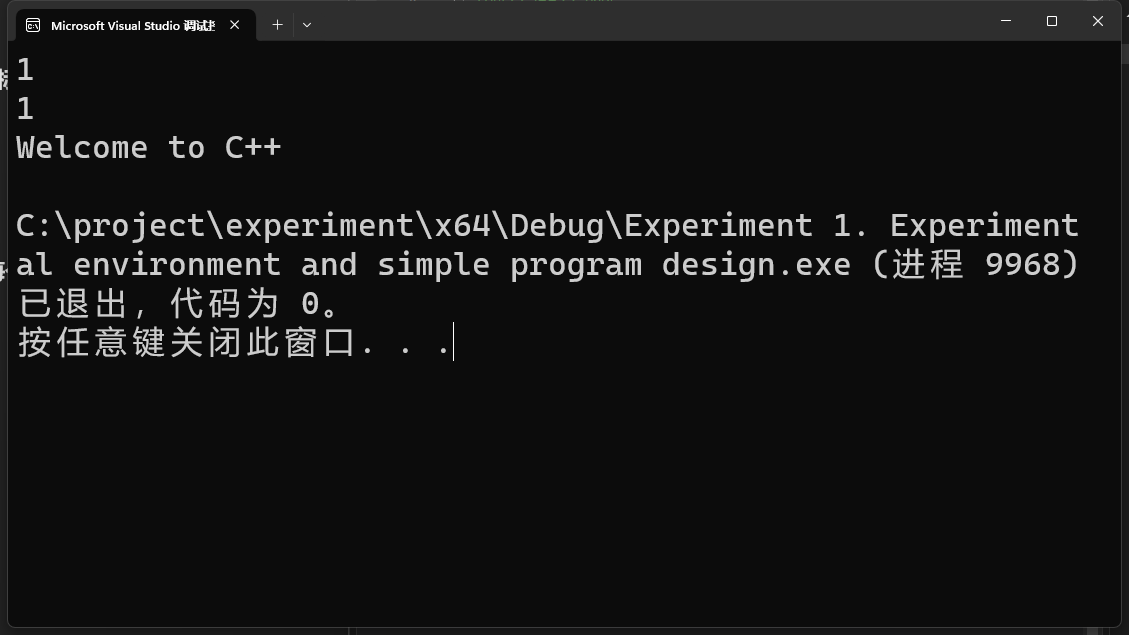
**自己编程测试一下将testUnint按8进制输出<<oct;je\_将一个实数转换成int,观察结果。**

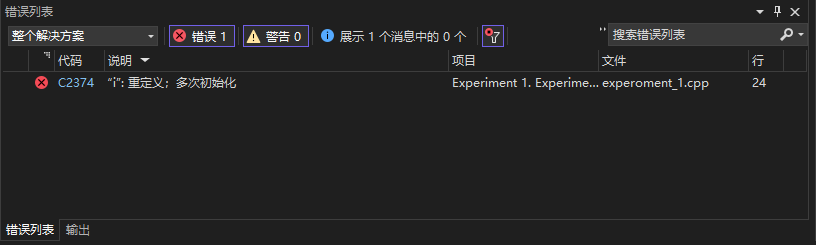
**5.编程，输入华氏温度，将其转换为摄氏温度后输出（保留两位小数）。**

**三、算法分析，程序结果**

1.



根据软件提示信息和编译过程报错提示，找出漏洞并正常运行。

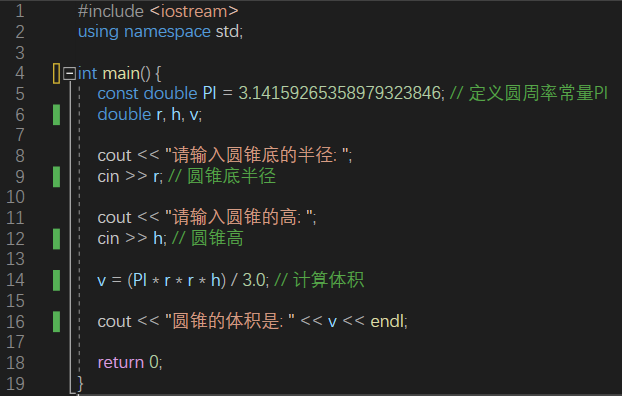


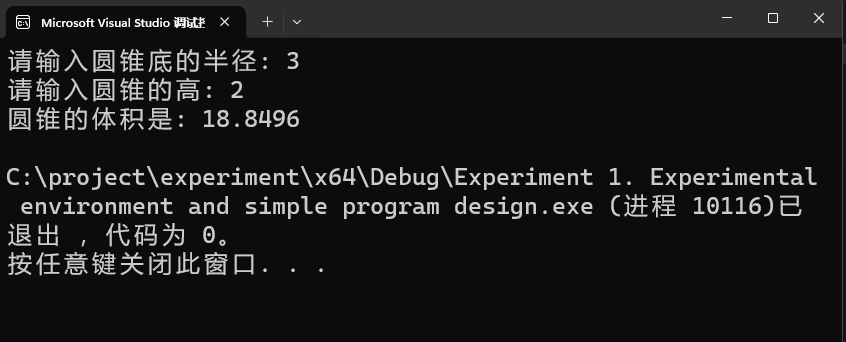
2. 查询圆锥公式，圆周率的近似值

定义初始变量r，h，v。

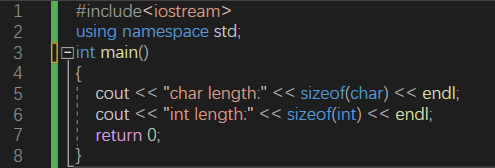
cin输入

代入公式





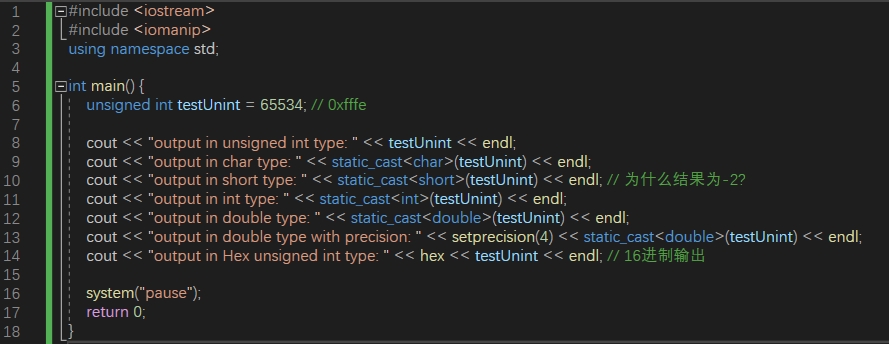
3.

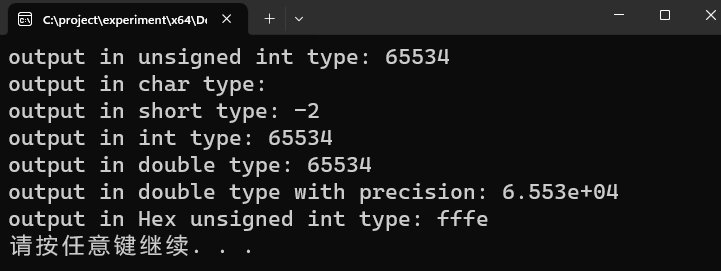


char为一字节，int为4字节



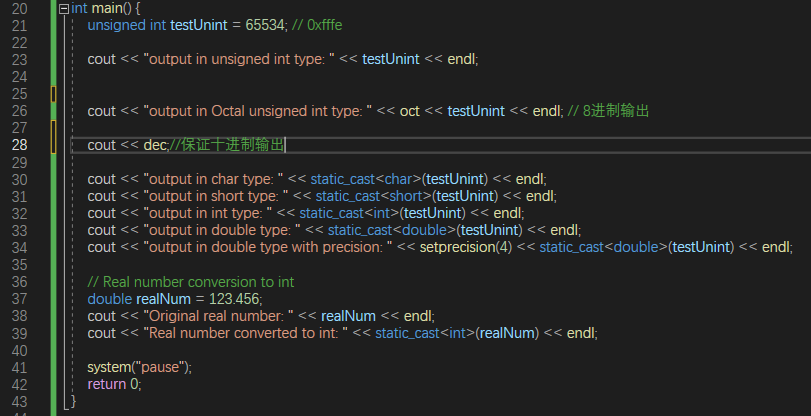
4.提供的代码有误，修改后为：





经过查询，可知老师提出的问题“为什么结果为-2？”的原因为：当 testUnint（一个无符号整数，值为 65534）被转换为 short 类型时，因为 short 类型通常是 16 位并有符号的，所以它会经历一个整型溢出。在大多数系统上，65534 转换为 16 位有符号整数时，其二进制表示与 -2 相同。这是由于二进制补码表示导致的，是整型溢出和有符号/无符号类型之间转换的一个常见例子。

任务



unsigned int是一种数据类型，在C++中用于存储非负整数。不同于 int（普通整型），unsigned int 不表示负数，因此其表示的正整数范围比普通 int 大。例如，如果 int 能表示 -32768 到 32767，则相应的 unsigned int 能表示从 0 到 65535。

由于这个输出内容涉及到我很多的知识盲区，我经过了深入学习，逐行分析代码主体部分：

cout << "output in unsigned int type: " << testUnint << endl;

这行代码将 testUnint 的值以无符号整数的形式输出。

cout << "output in char type: " << static\_cast<char>(testUnint) << endl;

这里，testUnint 被转换为 char 类型。由于 char 通常是 8 位长，而 65534 转换到 8 位会发生溢出，因此输出的字符可能是不可打印的，所以结果显示为空。

cout << "output in int type: " << static\_cast<int>(testUnint) << endl;

这里 testUnint 被转换为 int 类型，但由于 testUnint 本身就是整数类型，这个转换实际上没有改变它的值或表示

cout << "output in double type: " << static\_cast<double>(testUnint) << endl;

testUnint 被转换为 double 类型。这个转换之后数值输出保持不变。

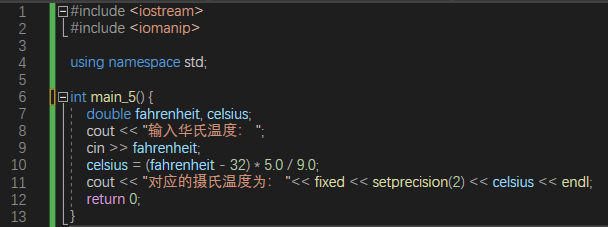
cout << "output in double type with precision: " << setprecision(4) << static\_cast<double>(testUnint) << endl;

这里，setprecision(4) 设定了浮点数的输出精度为 4 位有效数字。然而，由于 65534 是一个整数，所以在这个特定的例子中，精度设置不会影响输出。

cout << "output in Hex unsigned int type: " << hex << testUnint << endl;

这行代码使用 hex 流操纵符以十六进制格式输出 testUnint。由于 testUnint 的值是 65534，对应的十六进制是 fffe。

5.



**四、遇到的问题与解决方法**

详情见各个解答下

**五、体会**

首先是对C++集成开发环境的熟悉程度有了显著的提升。

通过不断地编译和运行程序，我深刻理解了C++程序的基本要素和开发过程。

在寻找和纠正代码错误的过程中，我认识到了程序设计的严谨性，比如变量的声明必须在使用之前。

实验中的类型转换和数据溢出的概念，说明了在进行混合算术运算时，正确处理数据类型转换的重要性。

在计算圆锥体积的练习中，我使用了标识符常量来定义圆周率，这涉及常量的使用，以及如何通过输入和输出语句来交互。

输入华氏温度并转换为摄氏温度的编程任务，锻炼了我在保留小数点位数方面的技巧。

**实验二、数据结构**

**一、实验目的**

1、学习与掌握逻辑运算与逻辑表达式。

2、熟练掌握if、switch、while、do-while，for语句的语法结构与执行过程。

3、掌握选择、循环程序的设计方法

**二、实验内容**

1、输入一个字符，如果为小写，转换为大写输出，否则，输出其后继字符的ASCII码值。

2、输入x计算表达式的值：



分别输入 0.2, 1, 5 , 0,观察输出结果。

1. 输入三角形的三条边，求周长，并判断该三角形是否为等腰三角形（提示：要三边是否可以构成三角形）。

4、 完成计算器程序，实现（+ - \* / %）运算。考虑除数为 0 与运算符非法的情况。

5、输入一行字符，分别统计出其中英文字母、空格、数字字符和其它字符的个数。

提示：从键盘上读入一个字符给变量 c，判断 c 是属于哪种字符并计数，循环读入下个字符，直到回车换行字符'\n'为止。

cin，scanf（）都不能读入空格以及‘\n’字符，查找资料解决输入这两个字符的方法。

（这个题训练大家自主学习能力以及如何获取新知识、探索解决未知问题的能力。）

6、编写一个程序：从键盘上输入两个正整数，求 a 和 b 的最大公约数与最小公倍数。

7、使用循环结构输出下列图形：

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

8、从键盘输入 a，用迭代法求 a 的平方根 x= *a* 。求平方根的迭代公式为：



要求精确到|xn+1 - xn|<10 -5。

提示：迭代法是把 xn代入迭代公式右边，计算出 xn+1来，然后把 xn+1 作为新的 xn ，计算出新的 xn+1，如此重复，直到|xn+1 - xn|<10 -5 时，xn+1 为所求的平方根。可以把 a 作为 xn 的初始值。

思考：（1）如果输入 a 为负，在运行时会出现什么情况? 修改程序使之能处理任何的 a 值。

(2）能否|xn+1 -xn|<10 -10或更小? 为什么? 请试一下。

9、苹果每个 0.8 元，第一天买 2 个，第二天开始，每天买前天的 2 倍，直到购买的苹果数不超过100的最大值，求每天平均花多少钱。

**三、算法分析，程序结果**

1、

首先，程序需要从用户那里获取一个字符。这可以通过标准输入流cin实现。

若判断输入的字符是否为小写字母。这一步可以使用islower函数完成，该函数检查给定的字符是否为小写字母。

如果是小写字母，则进入下一步转换为大写；如果不是，则进入求后继字符的ASCII值的步骤。

转换为大写：如果字符是小写字母，使用toupper函数将其转换为大写字母。toupper函数接受一个字符作为参数，并返回其大写形式。

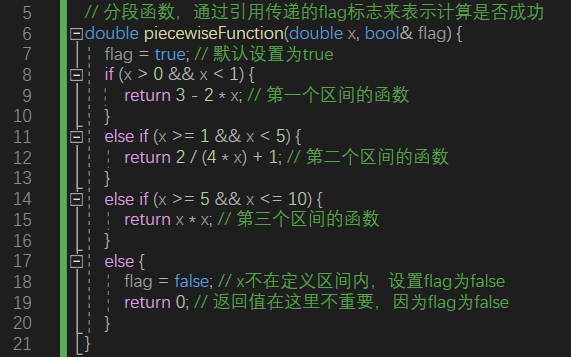
如果字符不是小写字母，则计算它的后继字符的ASCII值。这可以通过将字符加1实现，然后用static\_cast<int>将结果转换为整数以显示ASCII值。

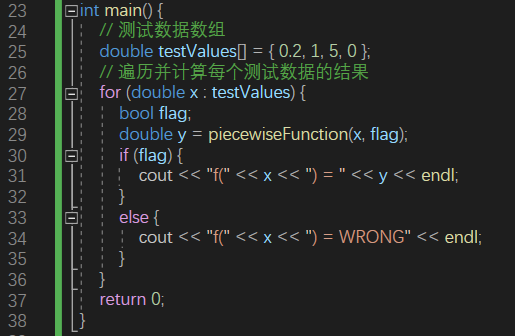


2、

我本来希望返回值可以是具体结果或者“wrong”结果，但是发现函数的返回类型固定，最后通过一个flag来代表函数运行情况，排除定义域之外的情况。

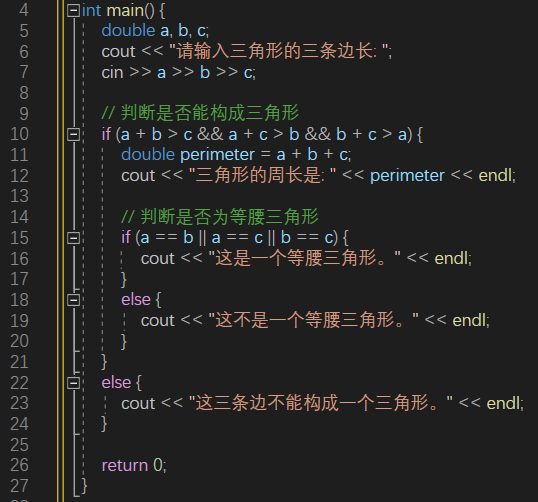
分段函数通过调用的方式解决，避免混乱，区分开来处理。



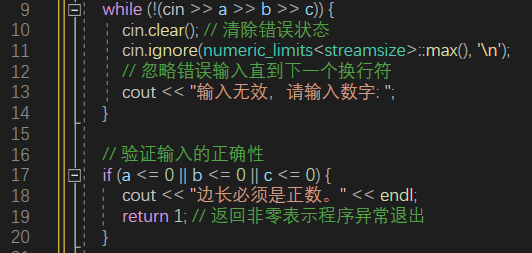


3、

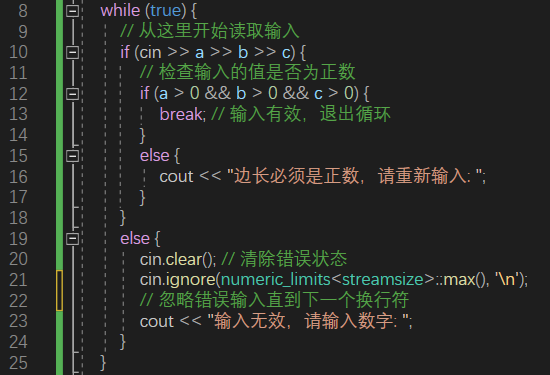
题目较简单，直接一个复合判断即可解决，使用double考虑小数问题，但其实还可能有精度考究，这个有点麻烦，就姑且忽略一下。



我仔细想想，感觉可以加入一个数值判断，如果输入了非法字符可以报错提示，更易于交互处理。输入了负数也可以提示一下。

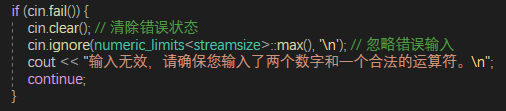


这个时候我发现输入了负数就会报错退出，我希望能重新输入，于是再次修改。这下可以直到正确输入一次再退出了。

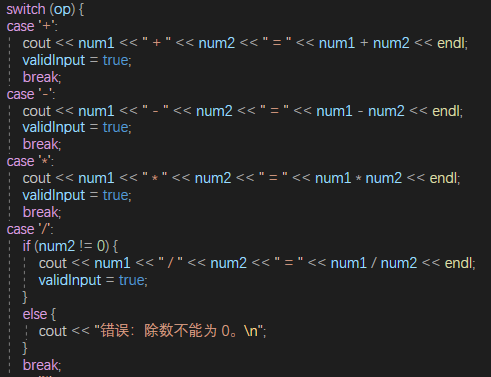


4、

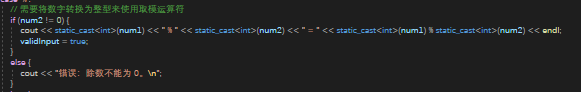
基本判断，类似上题。



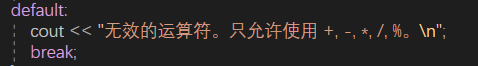
简单分类讨论，特别考虑除数为零的情况。



取模运算也需要特别考虑，他的转换形式不太一样。

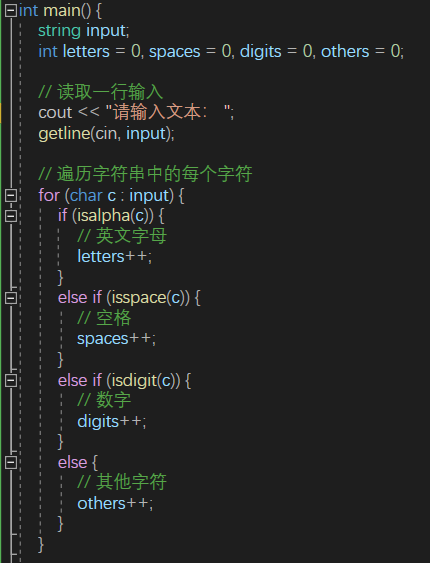


运算符出错情况



5、

查找到getline（）可以完美解决这个问题，它可以用来读取整行输入，包括空格，然后逐个字符进行判断。直接上手。EASY！

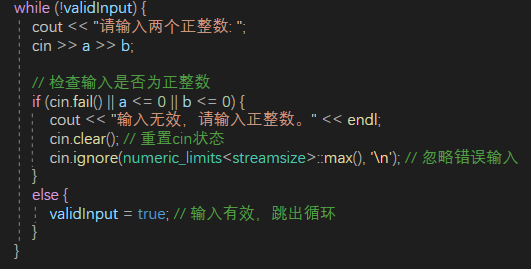


6、

查到有内置欧几里得函数，可以直接调用得到最大公约数。

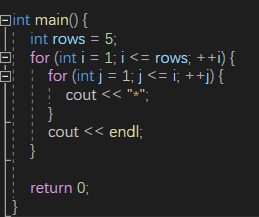


加一个经典输入无效的判断条件。

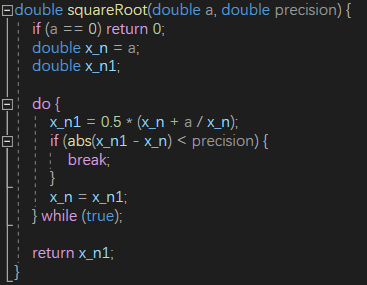


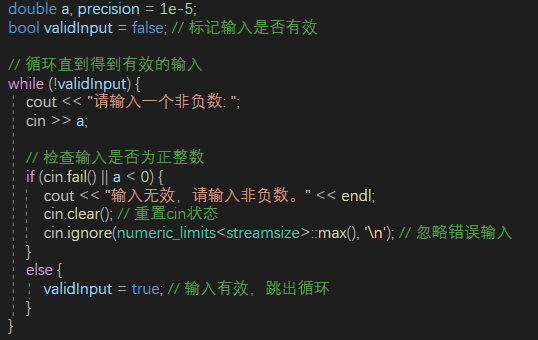
7、

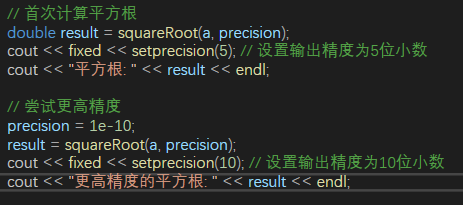
循环嵌套，简简单单。



8、







9、



**四、遇到的问题与解决方法**

**五、体会**