**计算机程序设计基础（C++)**

**实验报告**

专业班级： 软件工程2406

学 号： 8209240627

姓 名： 谭舒月

**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）。**

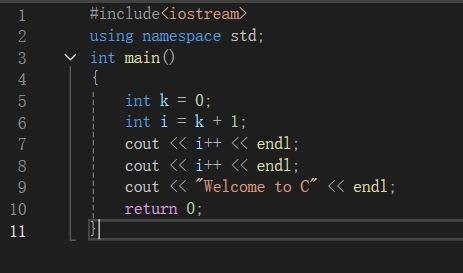
**三、算法分析，程序结果**

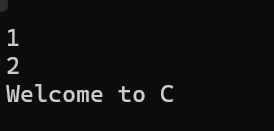
**1.**

1. **算法分析**

**找出错误：错误一-第五行int首字母大写；错误二-变量i重复定义**

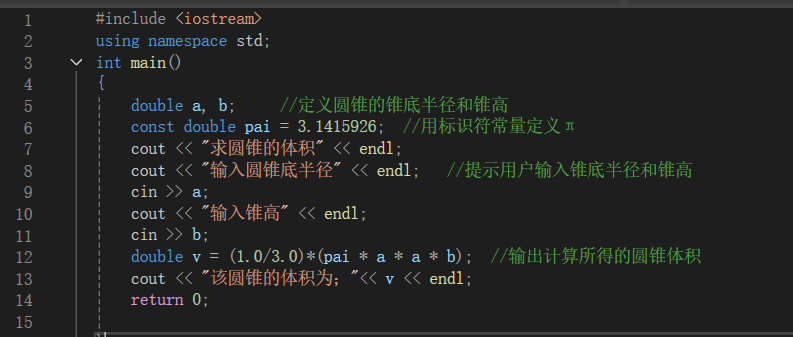
**修改错误：**



1. **程序结果**

**2.**

1. **算法分析**

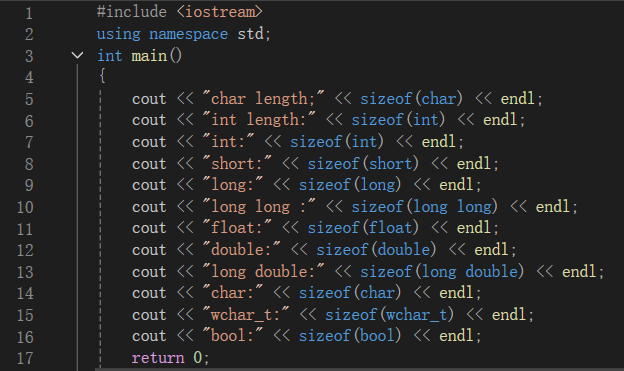


1. **程序结果**

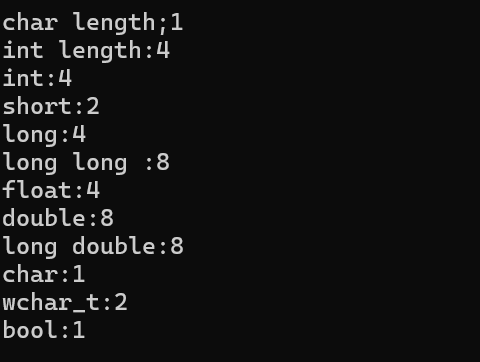


**3.**

**a.算法分析**



**b.运行结果**



**4.**

1. **算法分析**

**要使用八进制输出，可以对第七行代码作出更改，更改结果如下**

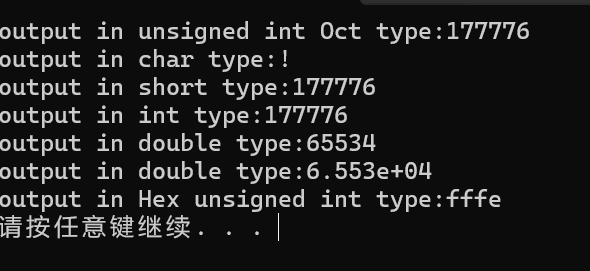


**解决第九行代码注释提出的问题：在 C++ 中，static\_cast<short>(testUnint) 将 unsigned int 类型的值 65534 转换为 short 类型时，发生了溢出。这是因为 short 通常是 16 位的，最大值为 32767，而 65534 超出了这个范围。**

**当将一个超过 short 范围的 unsigned int 转换为 short 时，会出现未定义行为，通常会将高位截断。具体来说，65534 在二进制中表示为 1111111111111110，当转换为 short 时，只保留低 16 位，结果变为 1111111111111110（十进制为 -2，因为 short 采用补码表示负数**）。

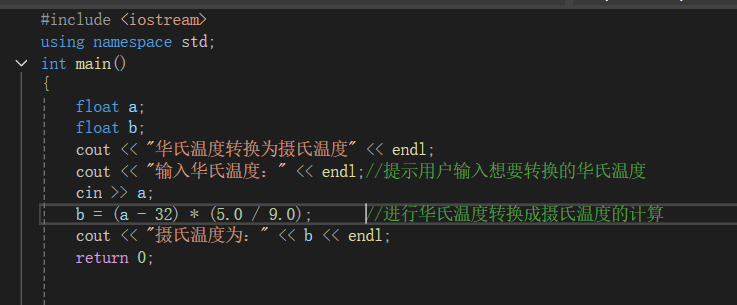


1. **运行结果**

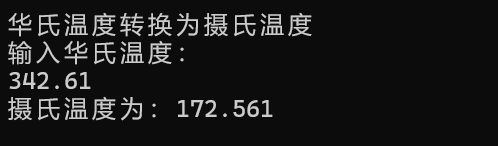


**5.**

1. **算法分析**



1. **运行结果**



**四、遇到的问题与解决方法**

**Q1:题目一第7行和第8行代码运行结果是1,2而不是2,3**

**A1:通过翻阅课本易知，在这两行代码中，i+＋表示先返回i的当前值，然后再将i 增加1。因此，第一次输出是1，第二次输出是2。**

**Q2:题目二在设计圆锥体积公式的时候使用计算式（1/3）**

**A2：询问chatGPT，得到解答如下：**

**整数除法：在C+＋中，如果两个整数相除，结果会被截断为整数。也就是说，1/3计算结果是e，因为整数相除只保留整数部分，去掉小数部分。**

**解决方法：通过将 1或3转换为浮点数，比如使用1.0或3.0，可以确保计算使用浮点数运算，从而得到正确的体积值。修改后的表达式（1.0/3.0)\*(pai \*a\*a\*b）就能返回一个正确的浮点数结果。**

1. **体会**

**解决问题需要实践，在实践过程中碰到的每一个问题都需要认真记录并进行课后巩固。对于常见的错误，要在日后的实践过程中多敲，多想，注重细节。在代码的世界里面，不完全对等同于完全不对。需要培养自己严谨的习惯，锻炼逻辑思维能力，掌握并熟练运用基本语句，熟悉基本概念。碰见报错或是输出结果不符合预期有时会让人产生挫败感和烦躁感，但是通过自己的能力去亲自解决，得到最终结果的那一瞬间让人很有成就感。**

**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。

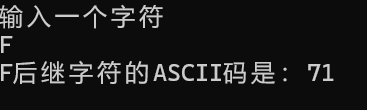
9、苹果每个 0.8 元，第一天买 2 个，第二天开始，每天买前天的 2 倍，直到购买的苹果数不超过100的最大值，求每天平均花多少钱。

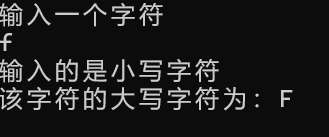
**三、算法分析，程序结果**

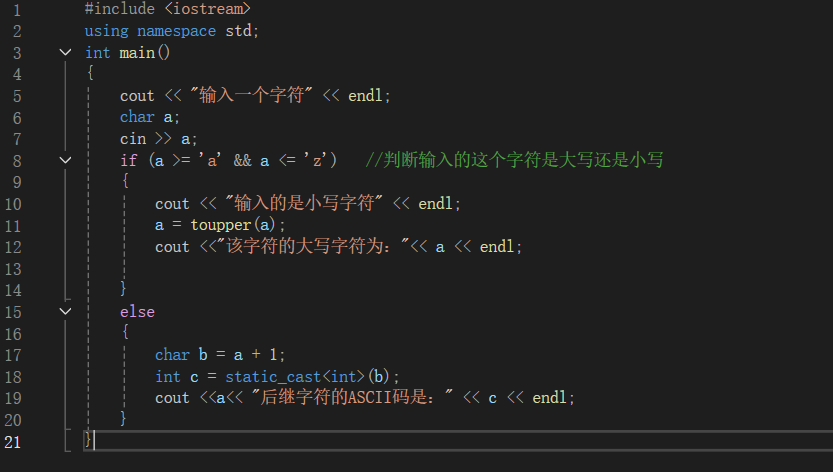
**1.**

**a.算法分析**

**b.程序结果**

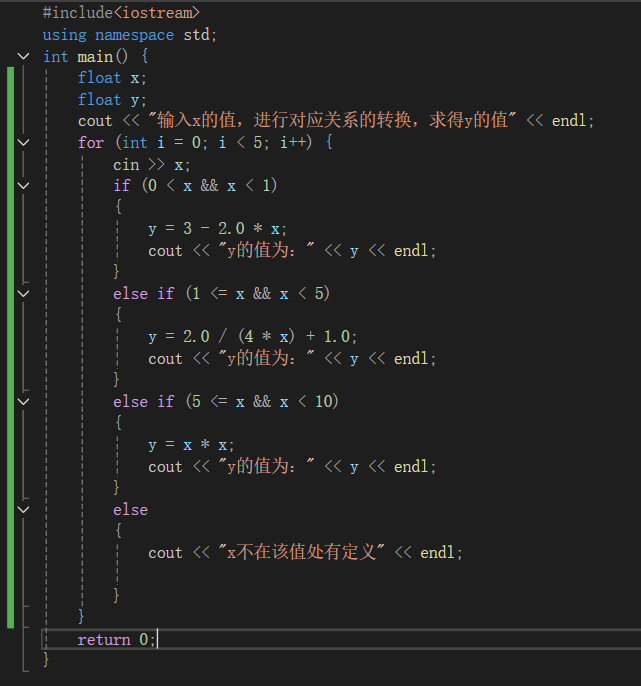




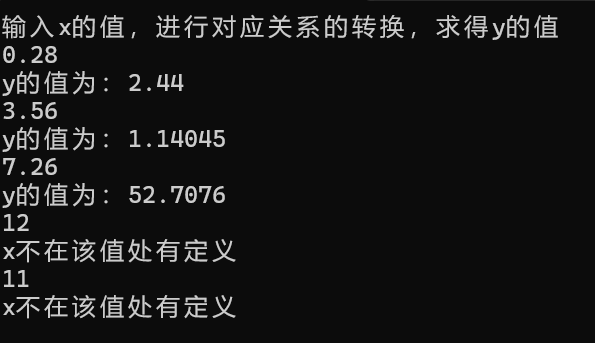


**2.**

**a.算法分析**

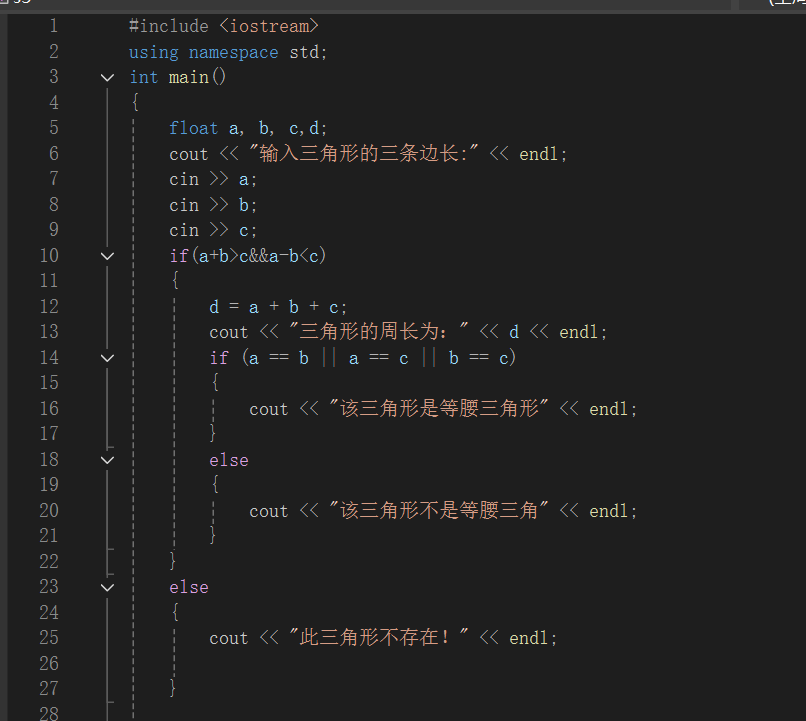


**b.程序结果**

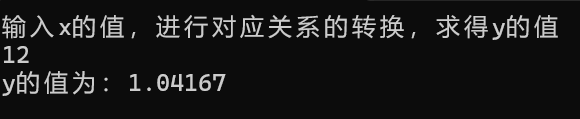


**3.**

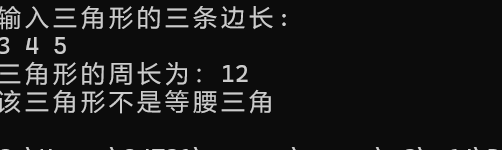
**a.算法分析**

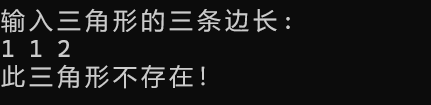


**四、遇到的问题与解决方法**



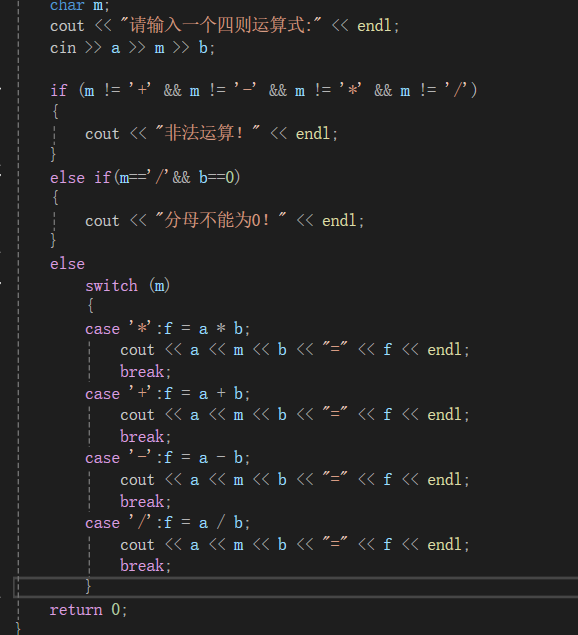
**b程序结果**



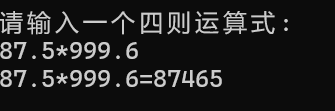


**4.**

1. **算法分析**

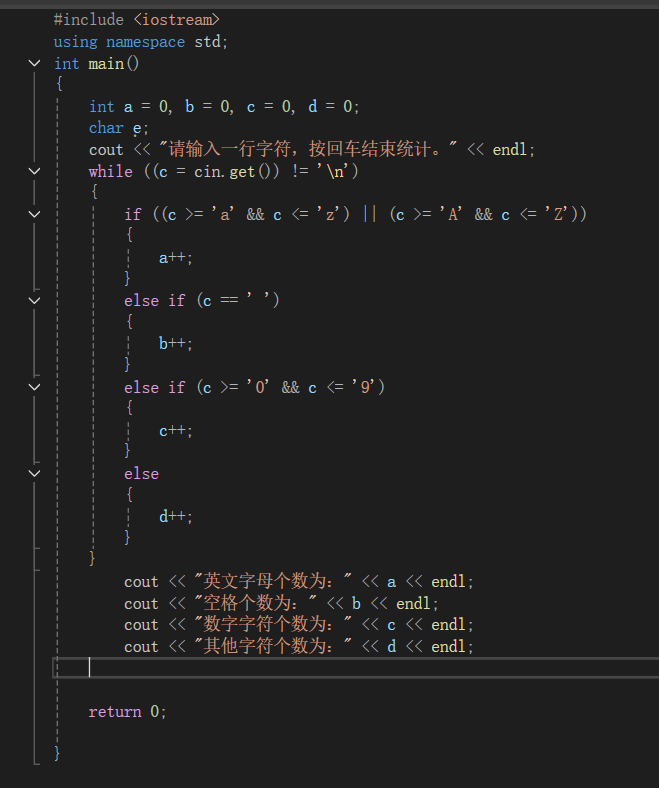


1. **程序结果**

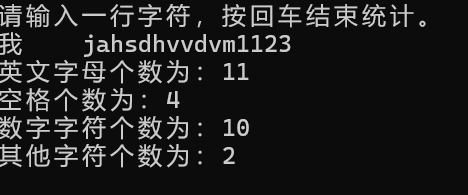


**5.**

1. **算法分析**

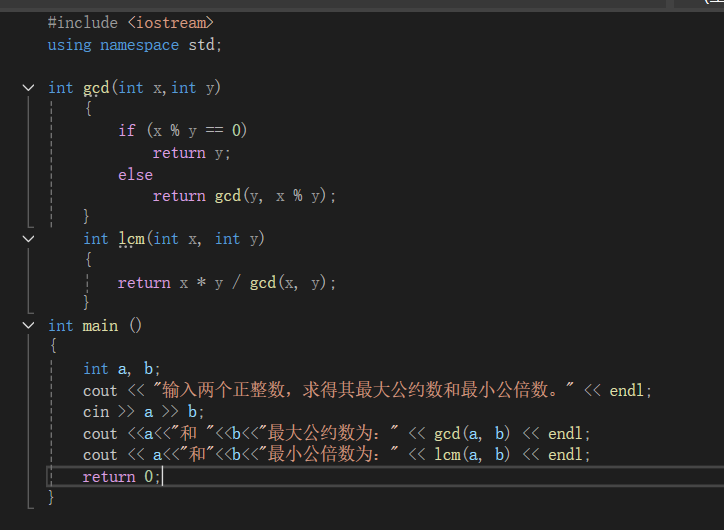


1. **程序结果**

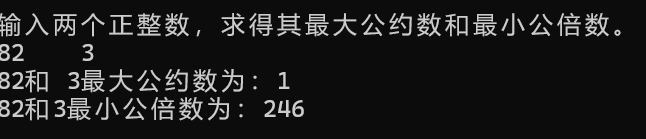


**6.**

1. **算法分析**

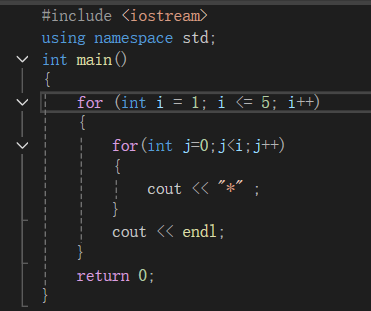


**b.程序结果**

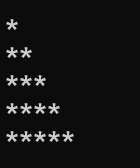


**7.**

1. **算法分析**

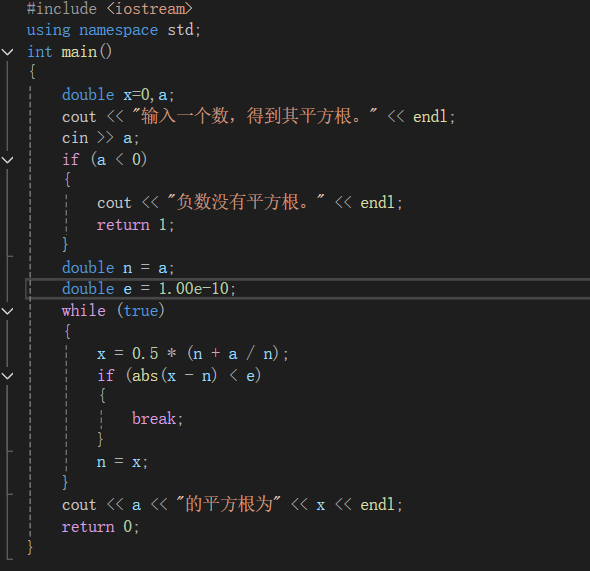


1. **程序结果**

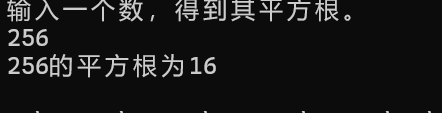


**8.**

1. **算法分析**



1. **程序结果**

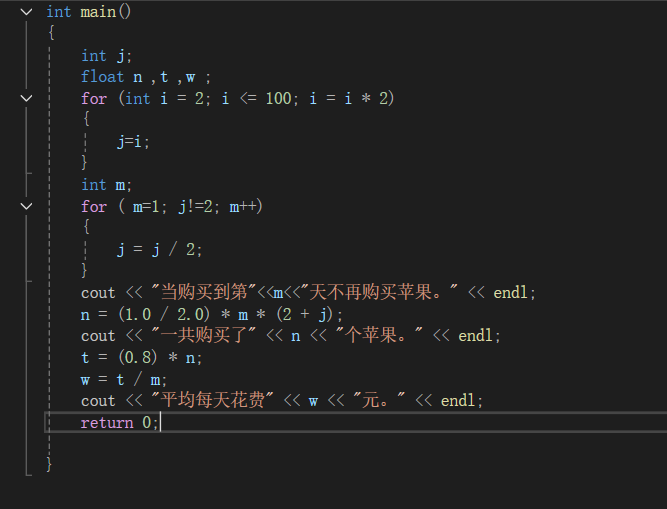


**q1 若a为负数，则程序执行不了，修改结果如上。**

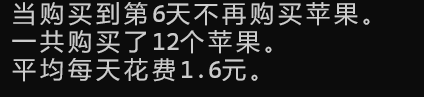
**q2 可以**

**9.**

1. **算法分析**



1. **程序结果**



1. **遇到的问题与解决方法**

**Q1：第五题时不知道如何读取空格**

**A1：在网上查阅资料发现使用可以cin.get()**读取字符**：cin.get()能够读取所有类型的字符，包括空格和换行符**

**Q2：在计算平方根时优化代码了解到一些书写习惯**

**A2：**

**while (true) 常用于需要不确定循环次数的算法，比如不断迭代直到满足某个特定条件。在你的平方根求解程序中，我们使用 while (true)，因为我们不知道需要多少次迭代才能达到指定的精度。**

**五、体会**

**while 特别适合用于未知迭代次数的情况。在实现牛顿-拉夫森法求平方根时，迭代次数无法预知，需要直到满足精度要求才停止，因此 while (true) 成为了理想的选择。它表示无限循环，需要在循环体内用 break 语句手动终止。这种结构在一定程度上增加了代码的灵活性，使我能够专注于在内部设置精确的退出条件。**

**与 while 循环相比，for 循环更适合用于已知迭代次数的场景，如遍历数组或执行固定次数的任务。虽然它简洁易读，但在实现诸如平方根求解这类无法预估迭代次数的算法时，for 循环并不适用。这次学习帮助我理解了选择合适的循环结构的重要性。**

**编写程序时，我注意到不同的 return 值传达了不同的信息。return 0 通常用于表示程序成功执行，而 return 1 则用于表示程序因错误或异常情况退出。这种区别在代码中提高了可读性和可维护性。例如，当输入负数时，程序输出提示并返回 1，清晰地表明程序因无效输入而终止。**

**在初次接触时，while (true) 这种无限循环看起来会让人觉得不太直观。然而，通过这次练习，我理解了它在一些需要不定次迭代的算法中的重要作用。在这种结构中，退出条件不由循环头部控制，而是在循环体内通过 break 来实现，这种方式提供了更强的控制力，使得循环在满足特定条件时才能退出。**

**总的来说，这次学习让我深刻认识到，不同循环结构的选择取决于具体的编程需求。理解其应用场景和控制逻辑的差异，是编写简洁高效代码的关键。**