**计算机程序设计基础（C++)**

**实验报告**

专业班级： 软件工程2402

学 号： 8209240206

姓 名： 王适涵

**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.未定义k的变量，且i重复定义

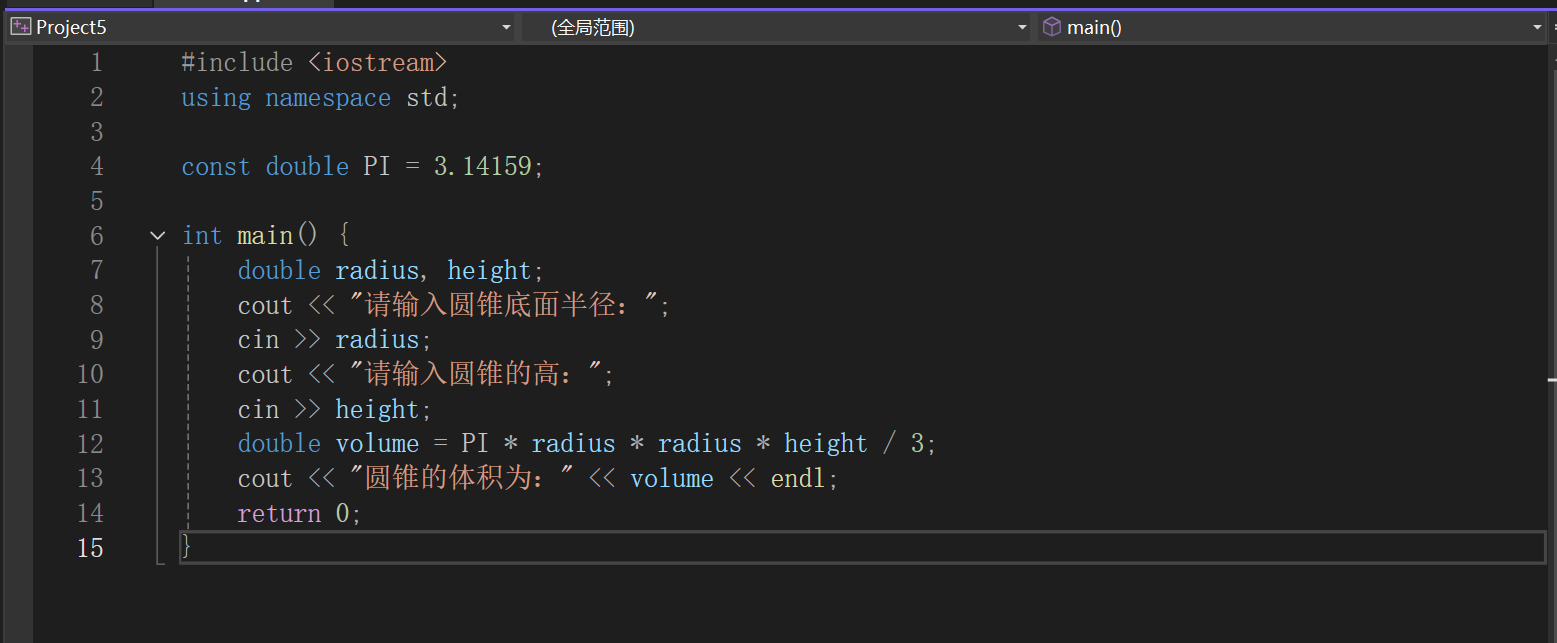
正确代码如下：

文本

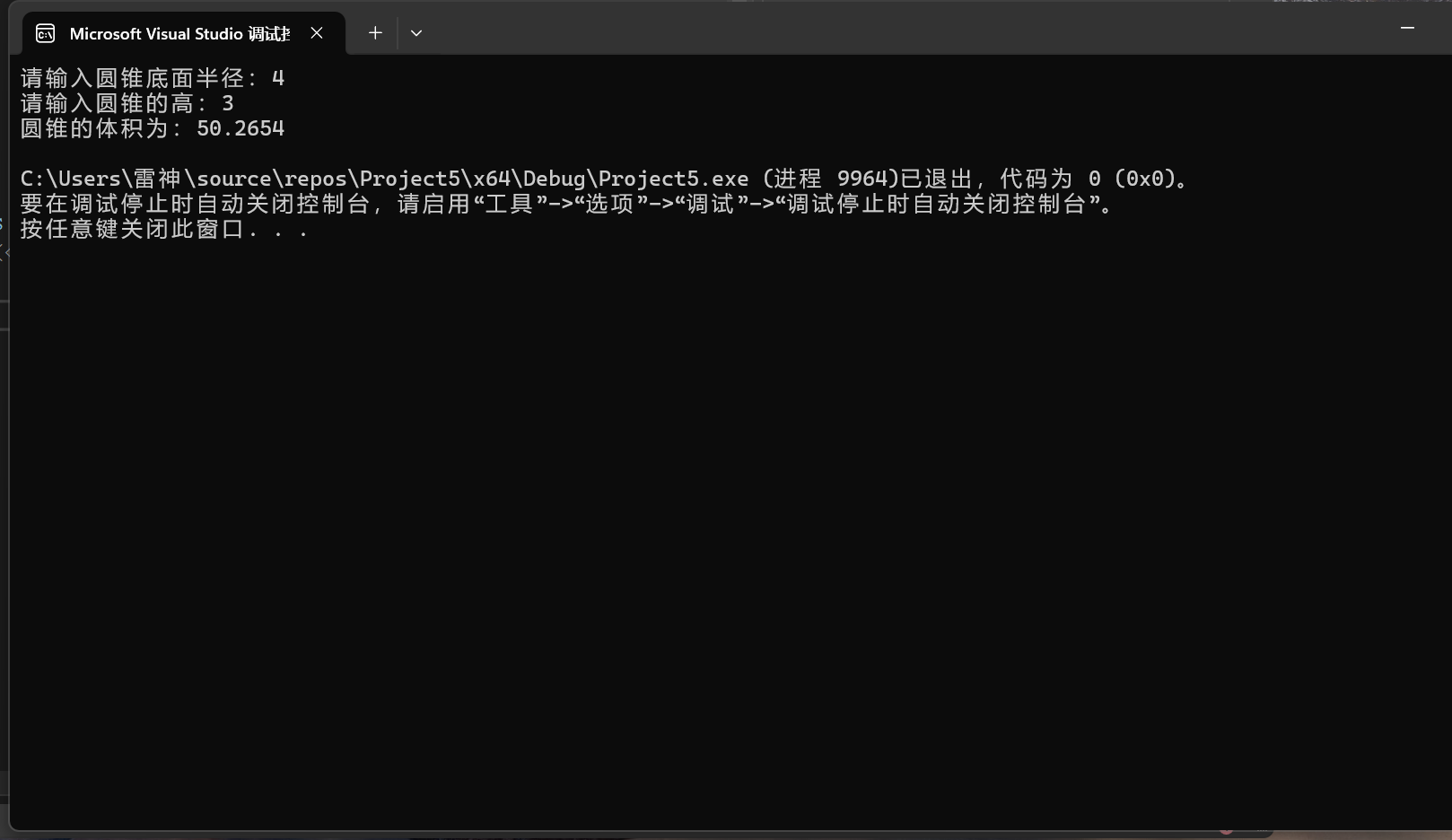
描述已自动生成电脑屏幕截图

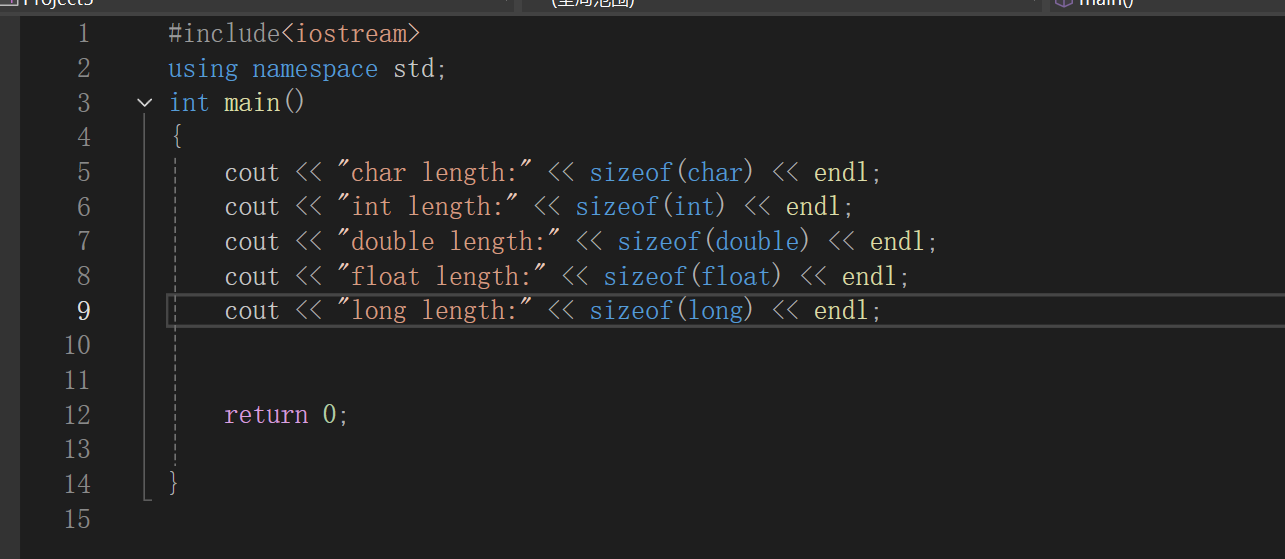
描述已自动生成

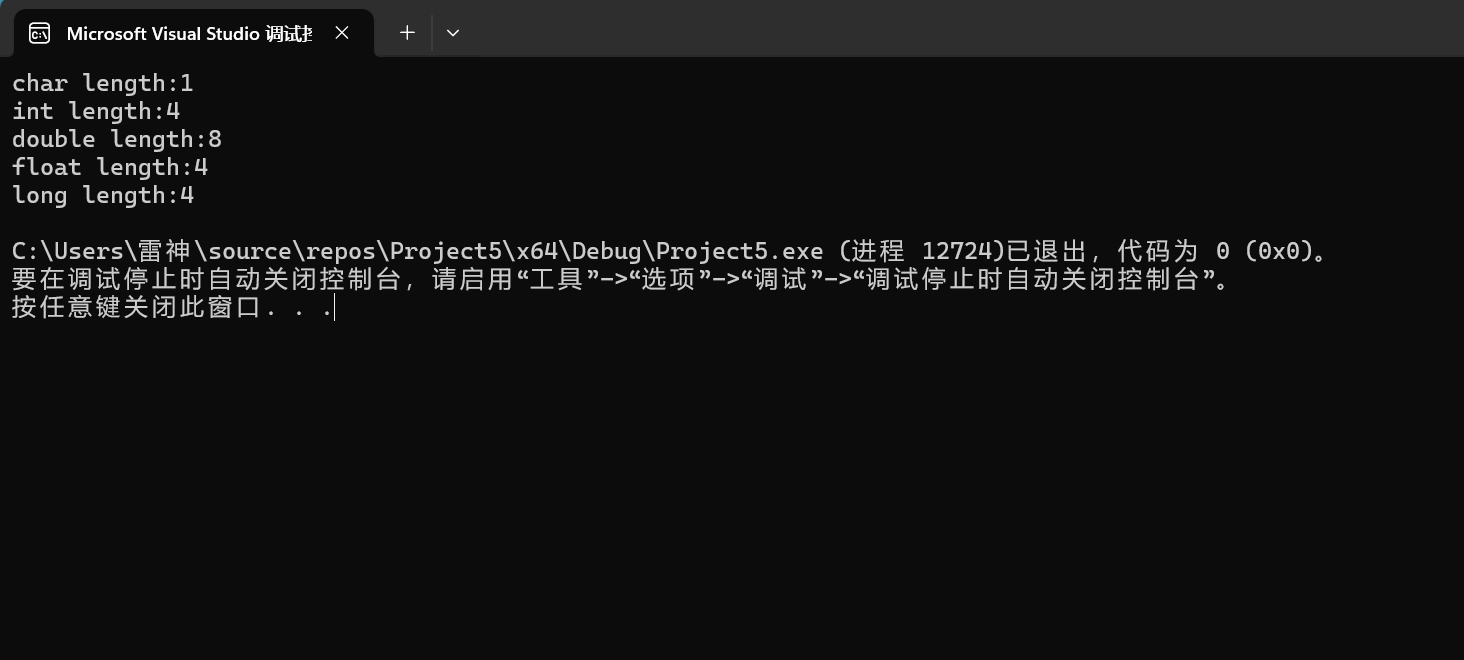
运行结果：

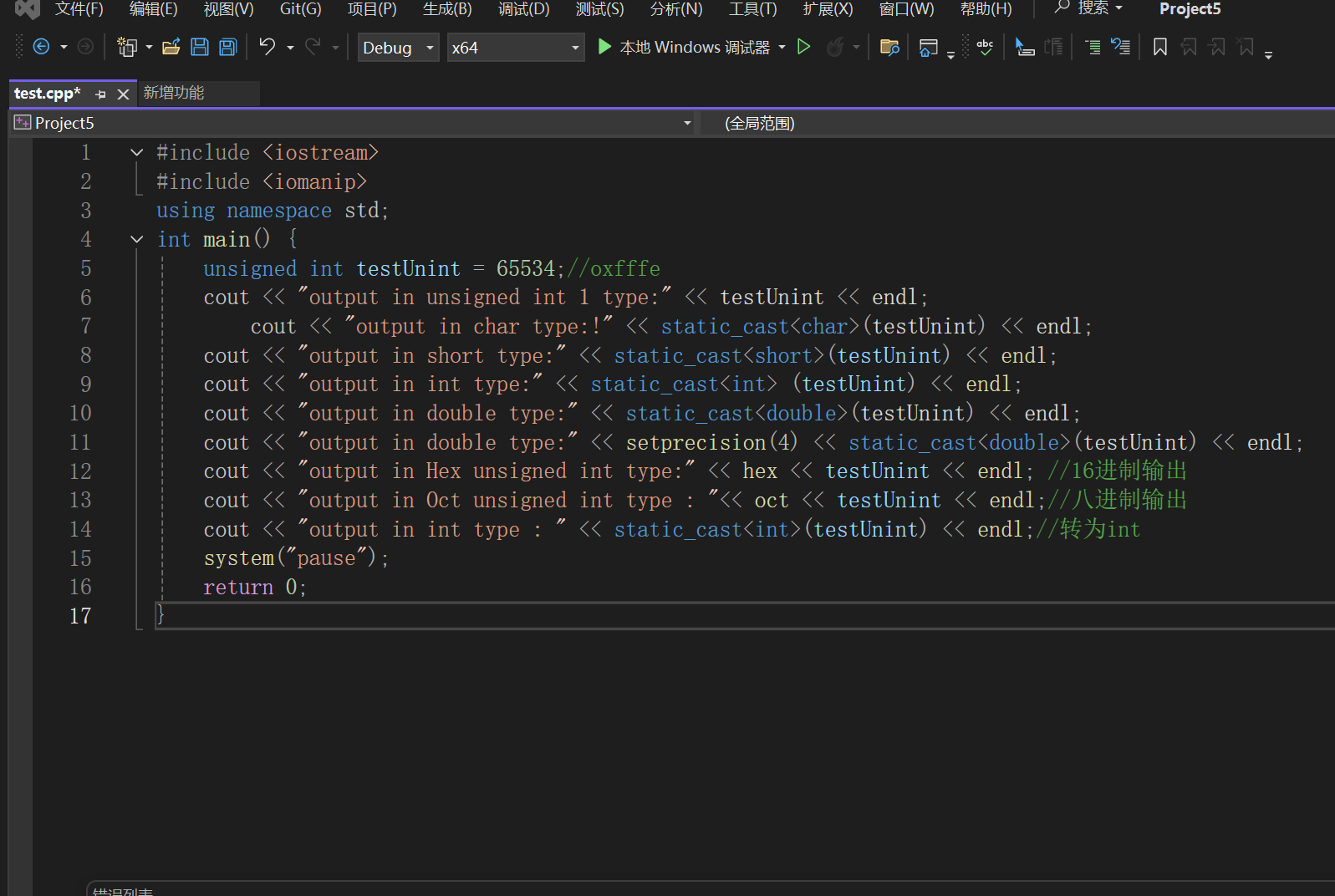
2.代码如下：

运行结果：

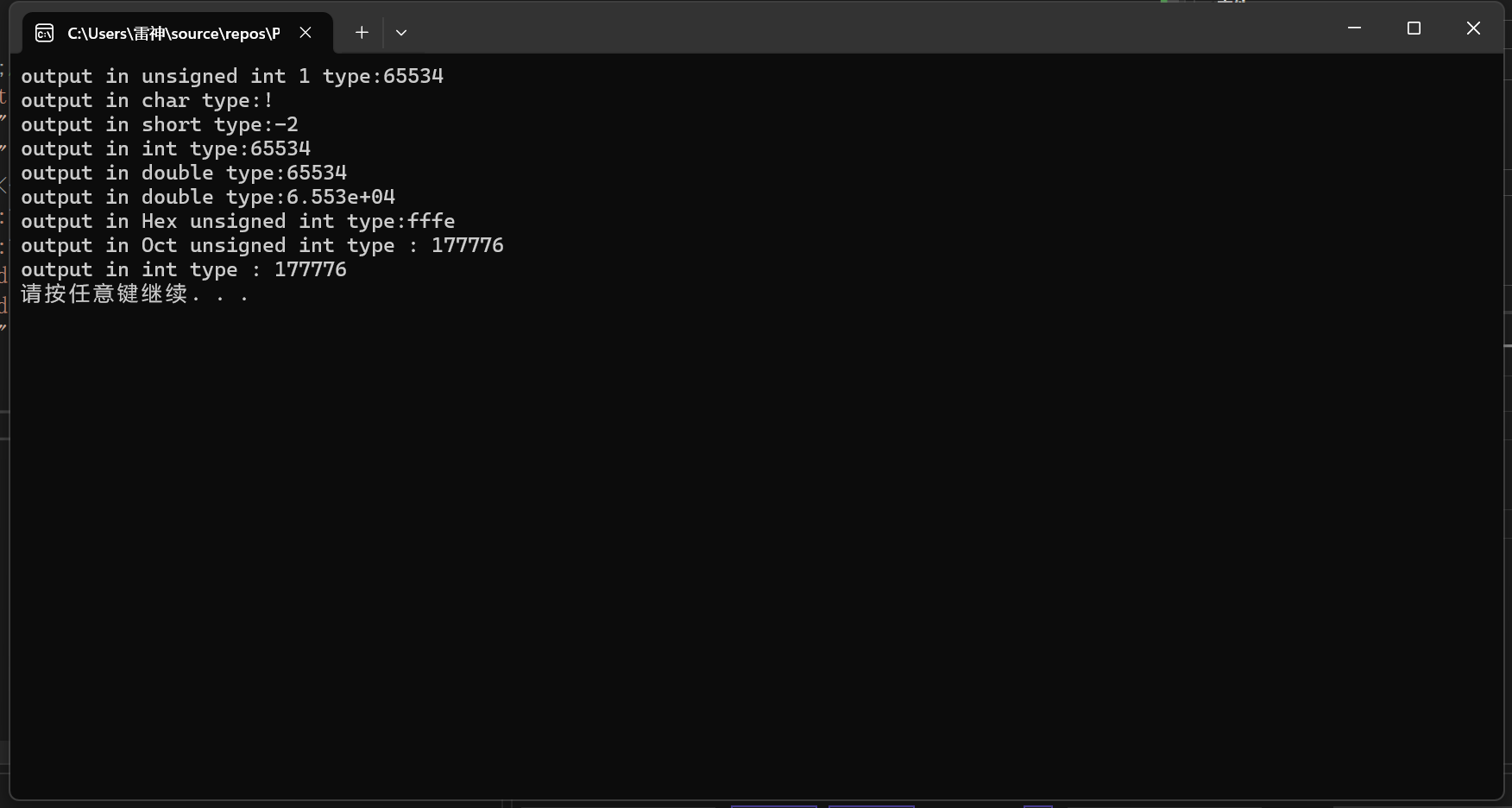
3代码如下：



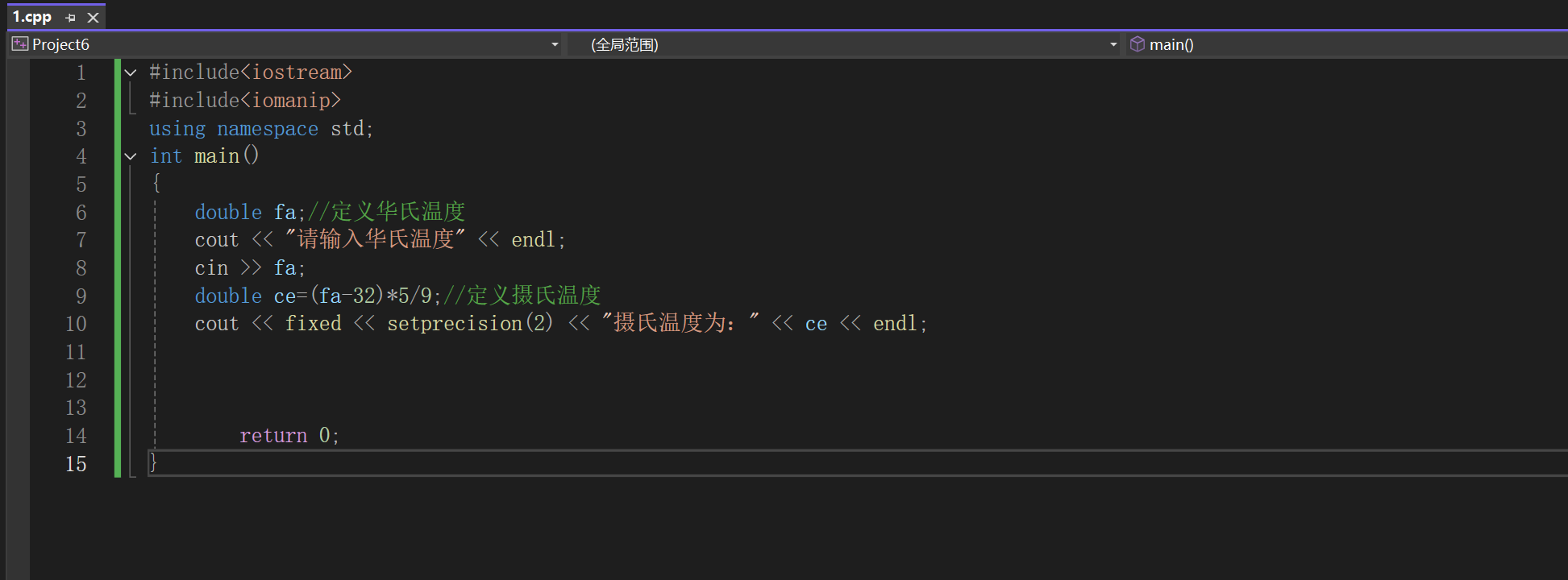
运行结果：

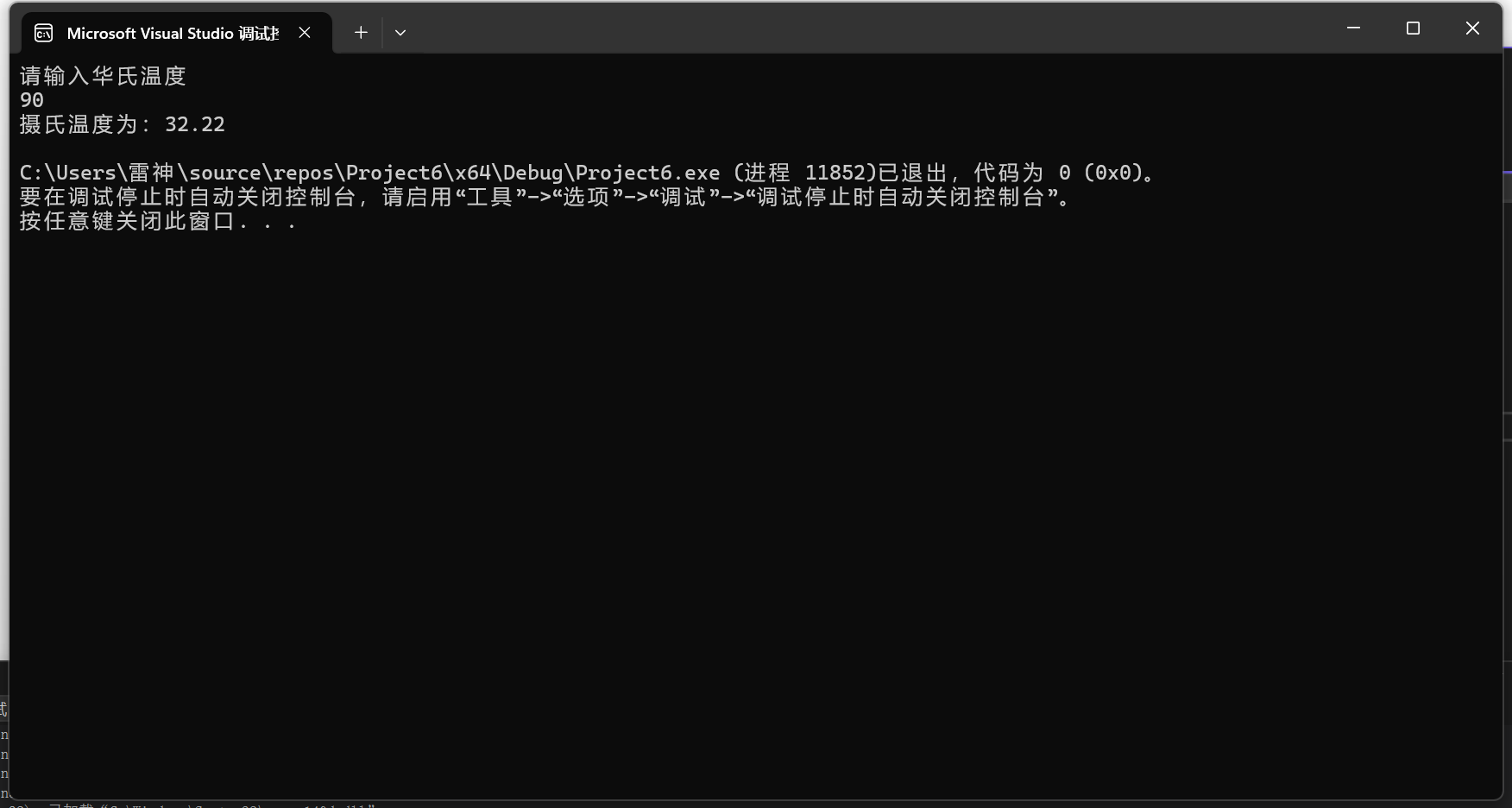
1. 代码如下：

运行结果：



5.代码如下：

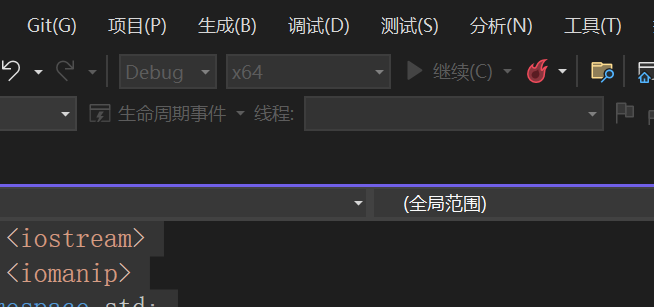
运行结果：



1. **遇到的问题与解决方法**
2. **在完成第四个习题时，不理解static\_cast 代码如何使用**

**通过上网查询以及询问讨论了解到static\_cast 的功能适用于数据类型的转换，从而解决了代码的问题。**

**2.在进行保存到新建文件夹的时候，进行运行会运行上一个项目的运行结果**

**需要在生成一栏添加新项目，避免成为无用项目。**

**3.在进行第五个项目时，先查询了保留两位小数的函数 setprecision**

**但是没有注意到忘记添加了iomanip的头文件。**

1. **体会**

**1.在日常C++的学习中，掌握实践技能固然重要，但不能遗忘一些重要的理论知识。**

在此次实验一的实验过程中，存在一些对于代码底层逻辑以及代码运算规则的考察。在编程的过程中，发现我可能更擅长代码的应用编写，而在面对底层逻辑运算规则的考察时，需要考编译器提供提示，才能注意到自己犯错的点。

2.**在进行自学时，不能只看重速度，还要注意质量**

在平日里进行自学时，我的学习目标可能是运用这一方面的知识，不会很对知识原理分散过多的注意力，这可能导致学习质量不是很高。

在自学时，可以通过网上查询，询问学长学姐进行自学，同时要注意优先自己解决问题，实在解决不了在寻求帮助。

**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：



1

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。

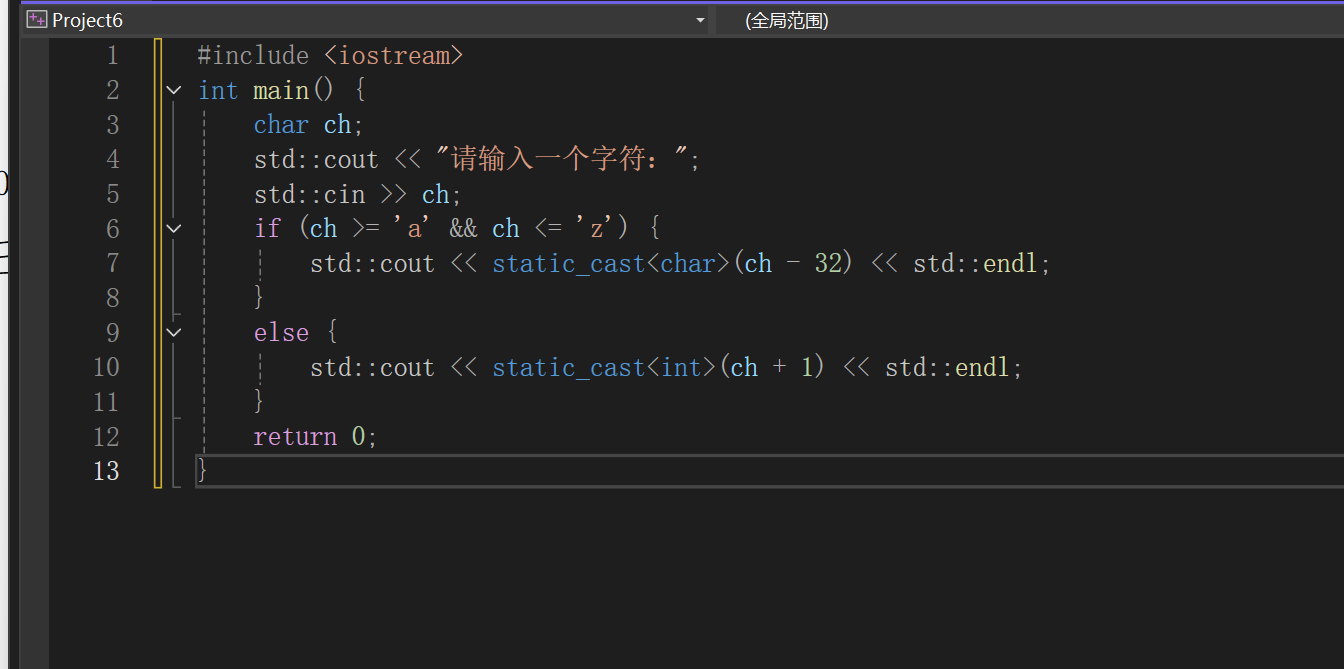
提示：迭代法是把 xn代入迭代公式右边，计算出 xn+1来，然后把 xn+1 作为新的 xn ，计算出新的 xn+1，如此重复，直到|xn+1 - xn|<10 -5 时，xn+1 为所求的平方根。可以把 a 作为 xn 的初始值。

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。

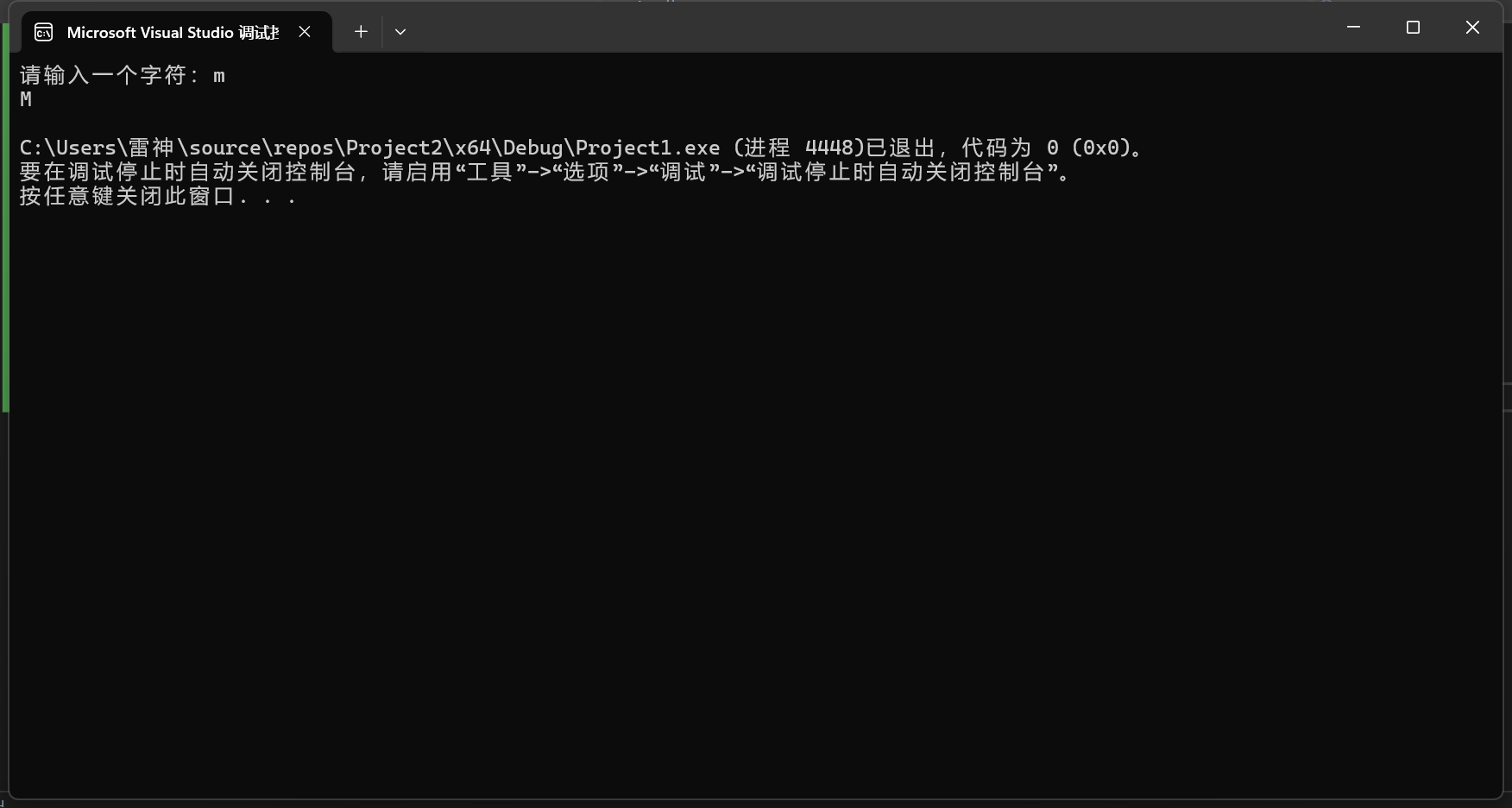
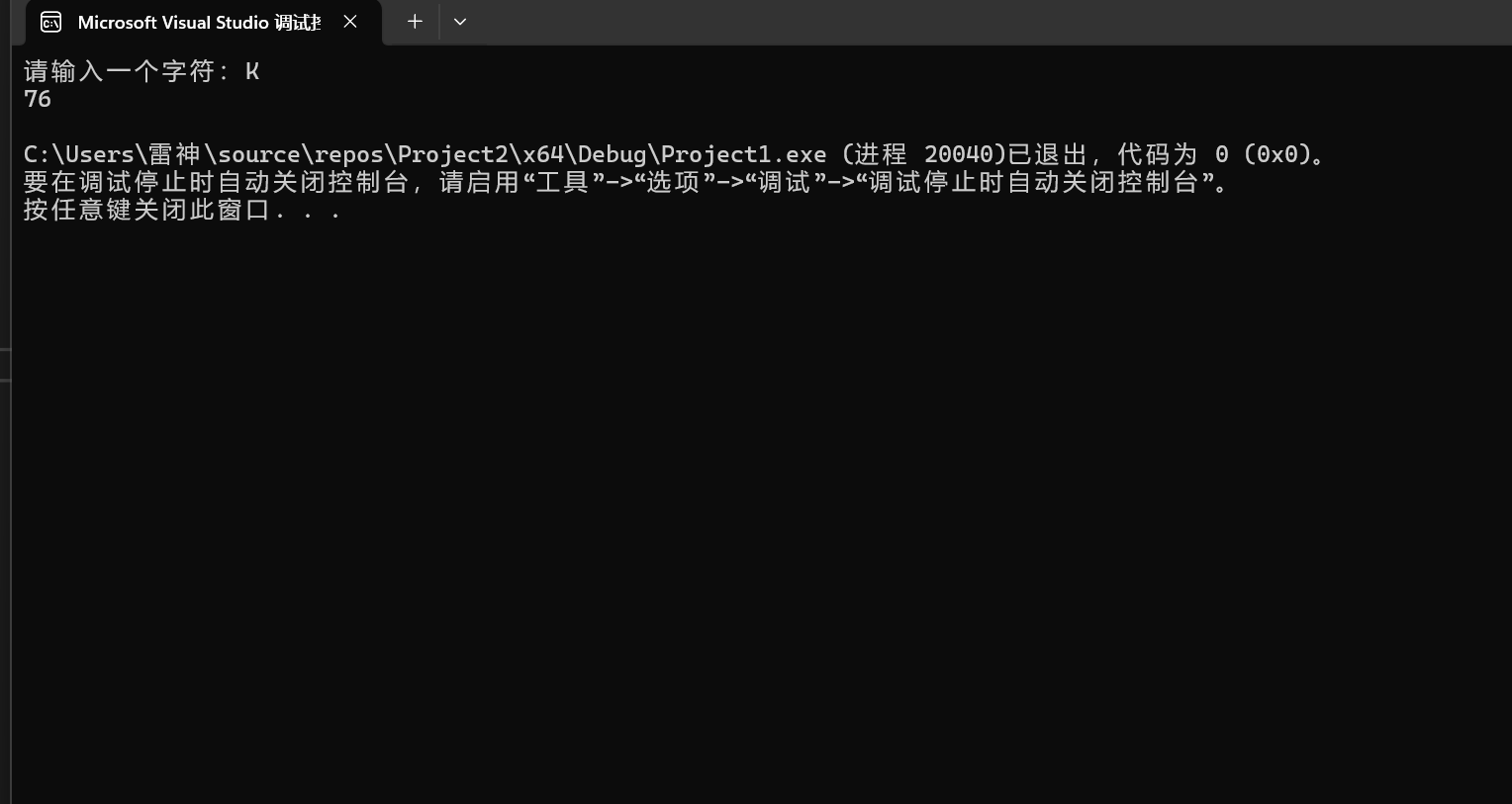
(2）能否|xn+1 -xn|<10 -10或更小? 为什么? 请试一下。

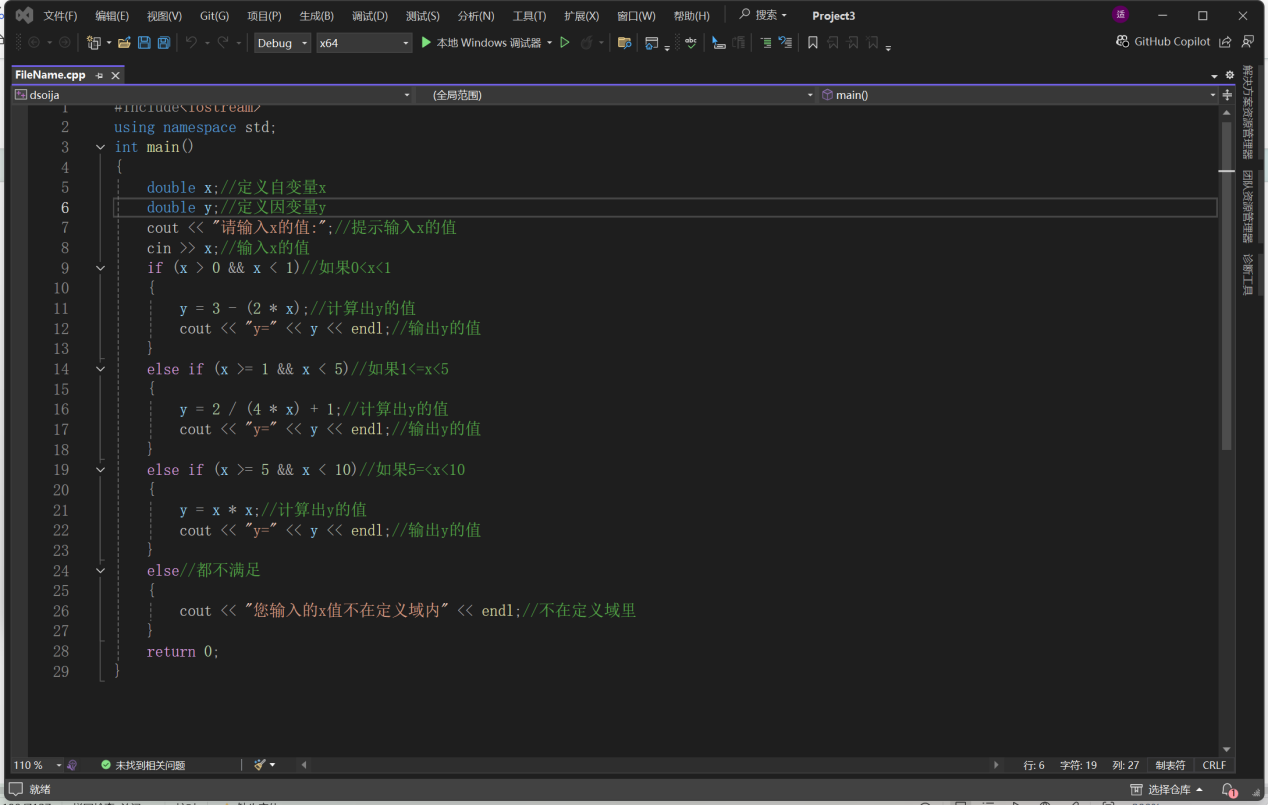
9、苹果每个 0.8 元，第一天买 2 个，第二天开始，每天买前天的 2 倍，直到购买的苹果数不超过100的最大值，求每天平均花多少钱。

1. **算法分析，程序结果**
2. **代码如下：**

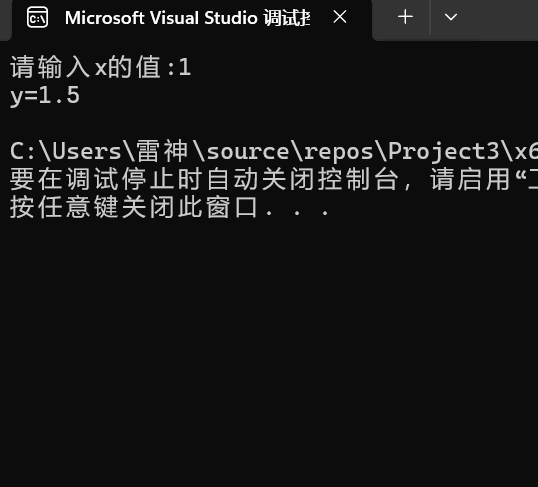
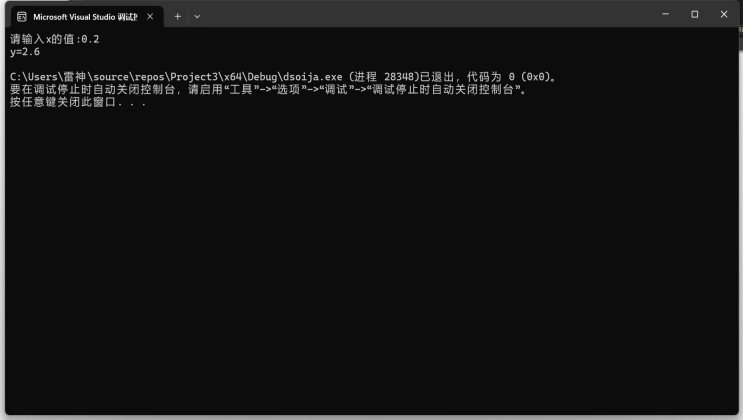
****

**运行结果：**

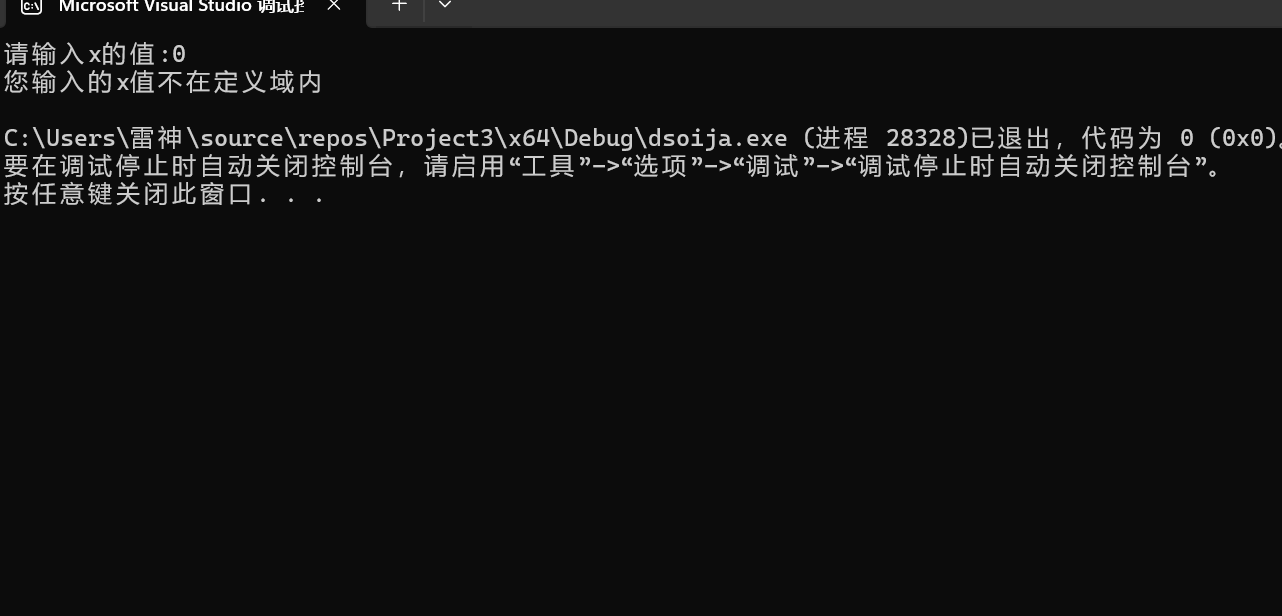
****

**2.代码如下：**

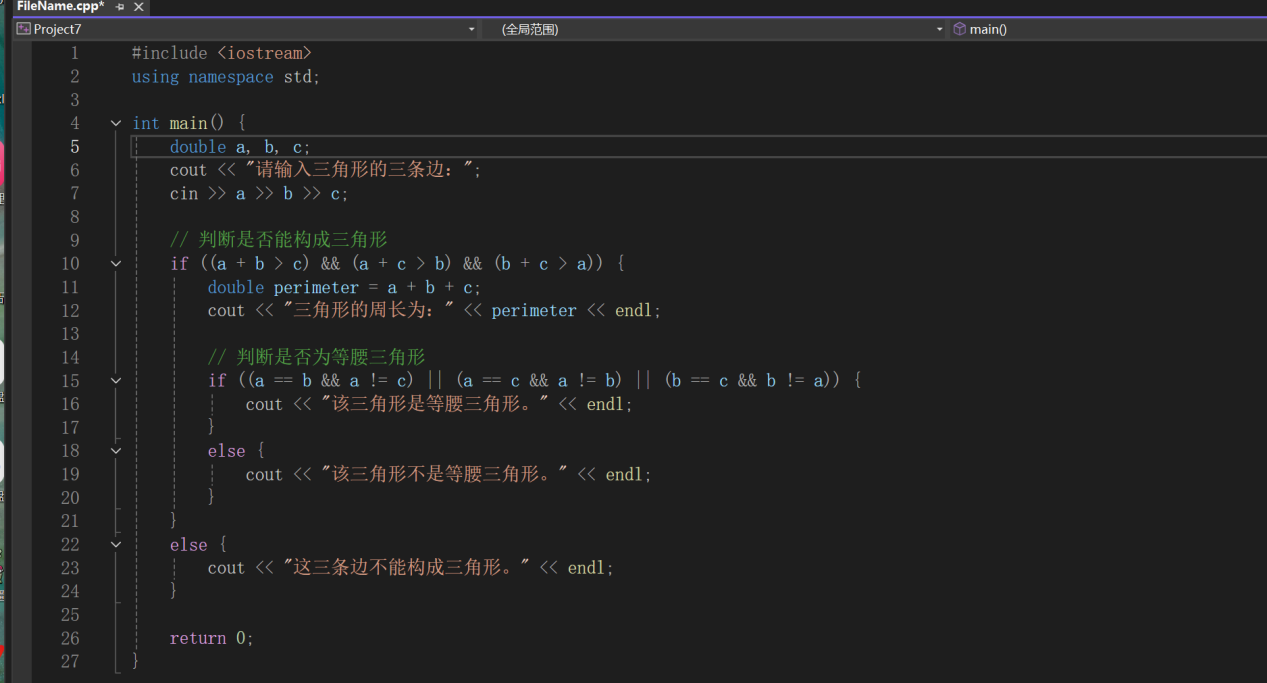
**运行结果：**

****

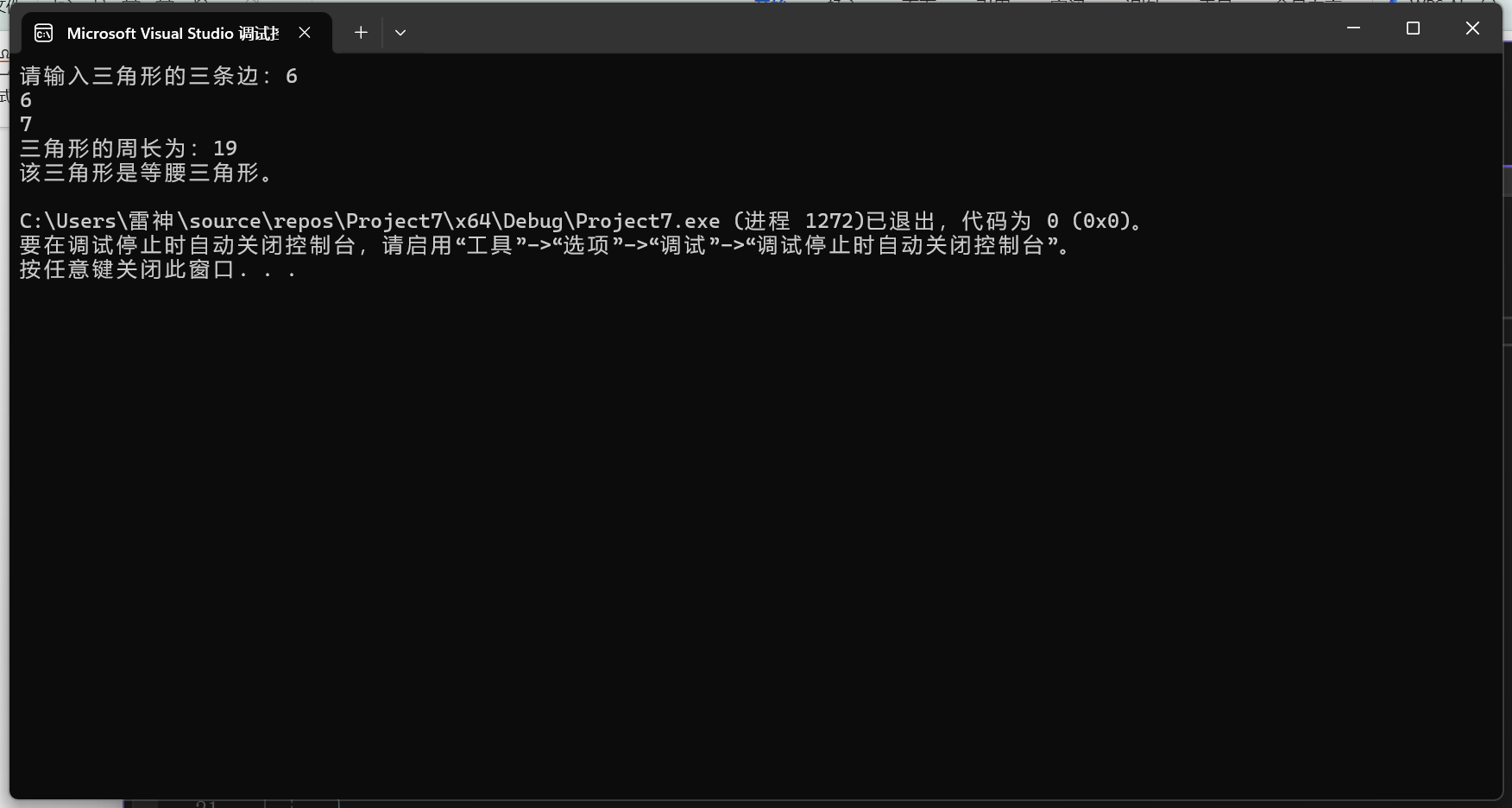
****

****

**3.代码如下：**

****

**运行结果：**

****

**4.代码如下：**

#include <iostream>

using namespace std;

int main() {

double num1, num2;

char op;

cout << "请输入第一个数：";

cin >> num1;

cout << "请输入运算符（+、-、\*、/、%）：";

cin >> op;

cout << "请输入第二个数：";

cin >> num2;

double result;

bool validOp = true;

switch (op) {

case '+':

result = num1 + num2;

break;

case '-':

result = num1 - num2;

break;

case '\*':

result = num1 \* num2;

break;

case '/':

if (num2 == 0) {

cout << "除数不能为 0。" << endl;

validOp = false;

}

else {

result = num1 / num2;

}

break;

case '%':

if (static\_cast<int>(num2) == 0) {

cout << "除数不能为 0。" << endl;

validOp = false;

}

else {

result = static\_cast<int>(num1) % static\_cast<int>(num2);

}

break;

default:

cout << "非法运算符。" << endl;

validOp = false;

}

if (validOp) {

cout << "结果为：" << result << endl;

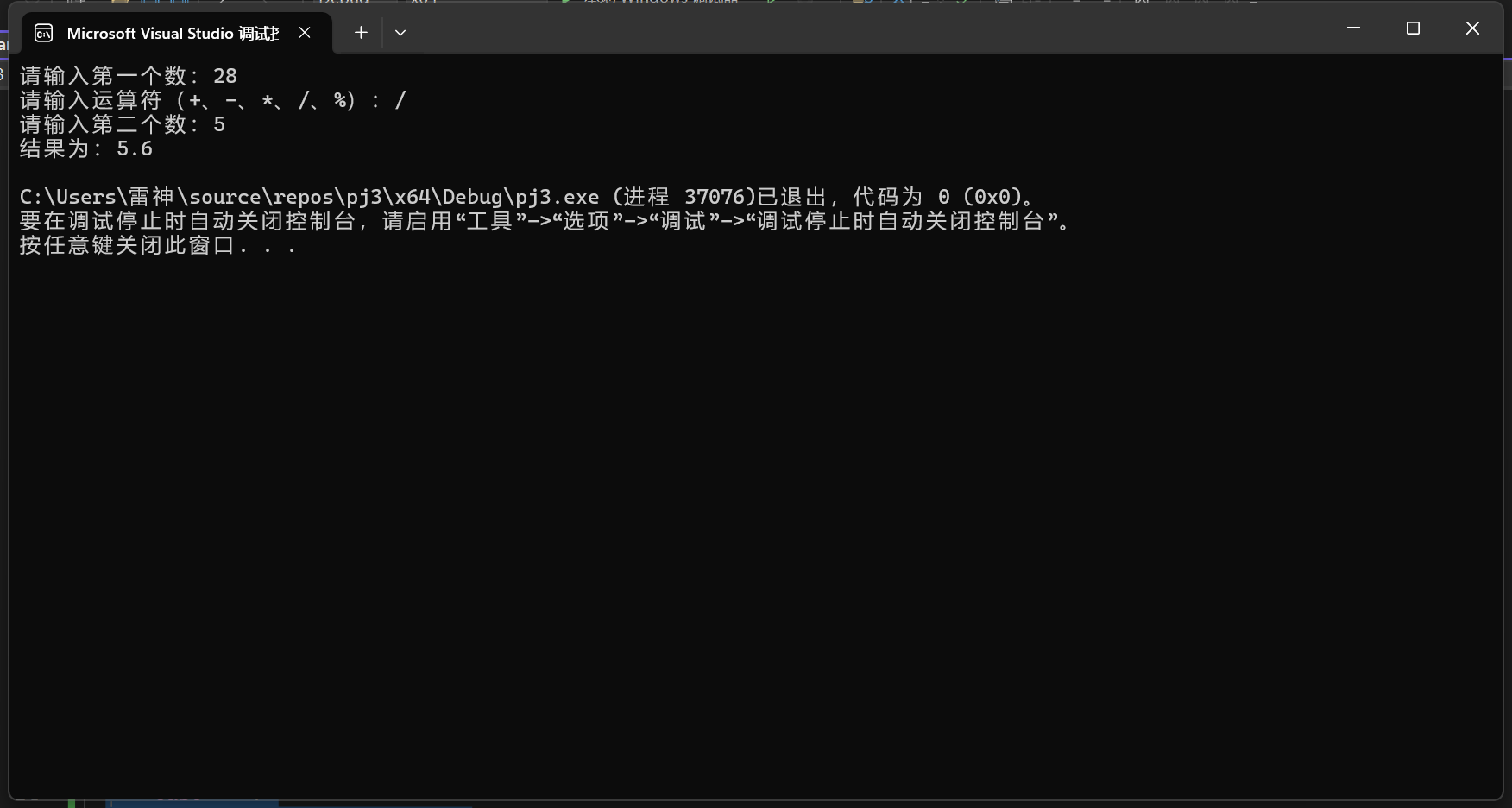
}

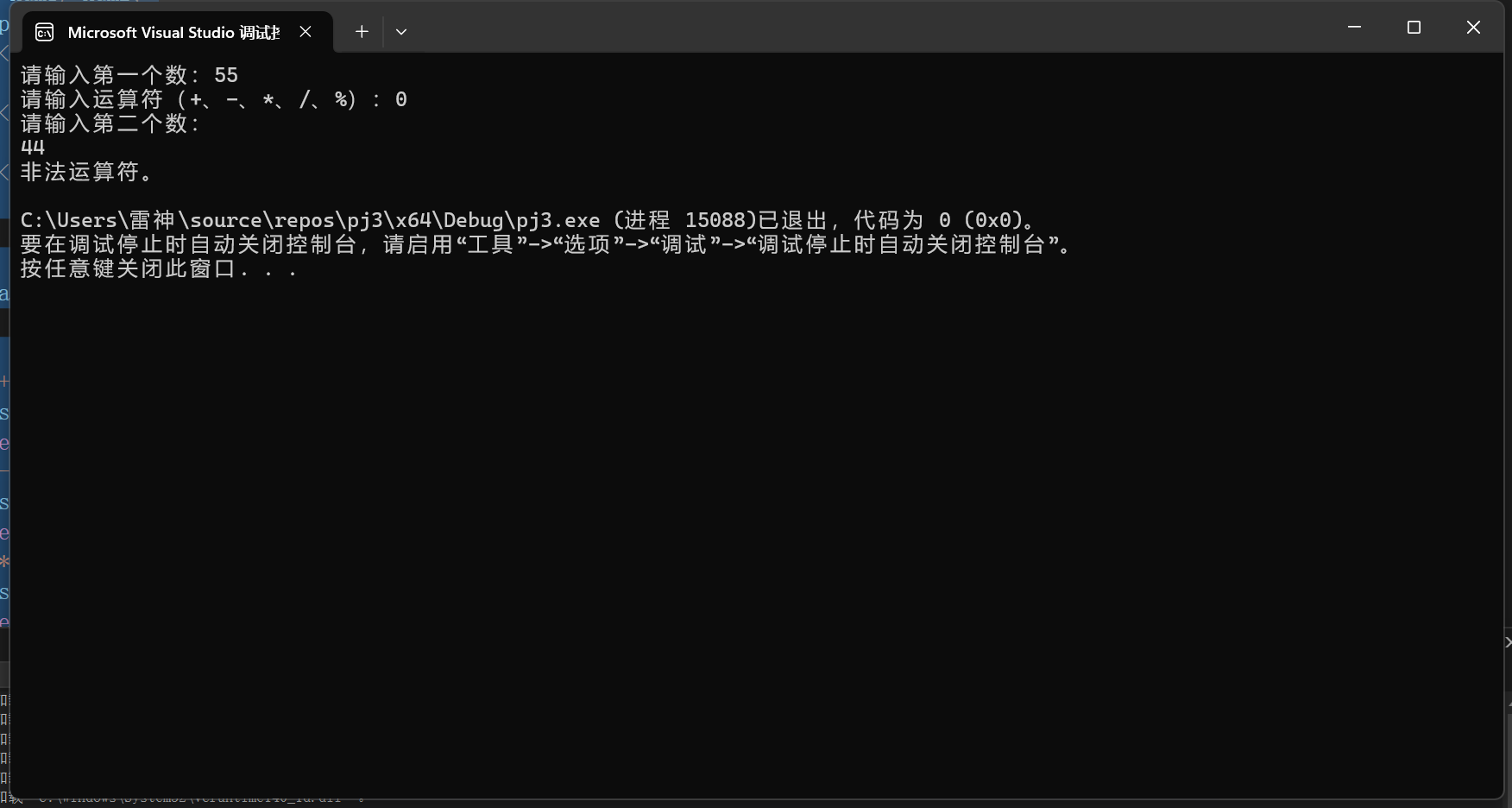
return 0;

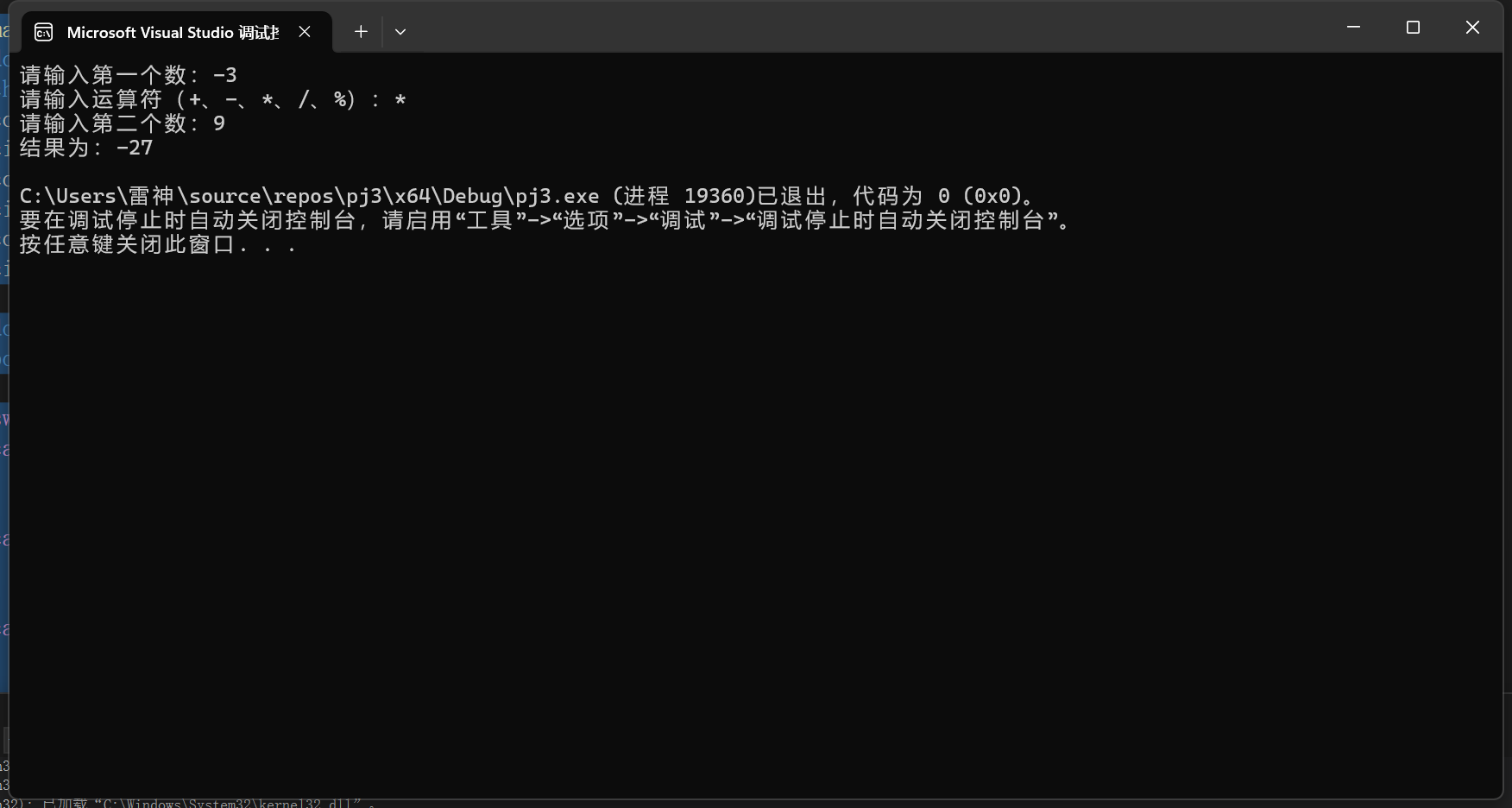
}

**（这里代码太长了一张图截不下来）**

**运行结果如下：**

****

****

****

1. **代码如下：**

#include <iostream>

using namespace std;

int main() {

char c;

int letters = 0, spaces = 0, digits = 0, others = 0;

cout << "请输入一行字符：";

while ((c = cin.get()) != '\n') { // 读取字符直到回车换行

if ((c >= 'a' && c <= 'z') || (c >= 'A' && c <= 'Z')) {

letters++; // 统计英文字母

}

else if (c == ' ') {

spaces++; // 统计空格

}

else if (c >= '0' && c <= '9') {

digits++; // 统计数字字符

}

else {

others++; // 统计其他字符

}

}

cout << "英文字母个数：" << letters << endl;

cout << "空格个数：" << spaces << endl;

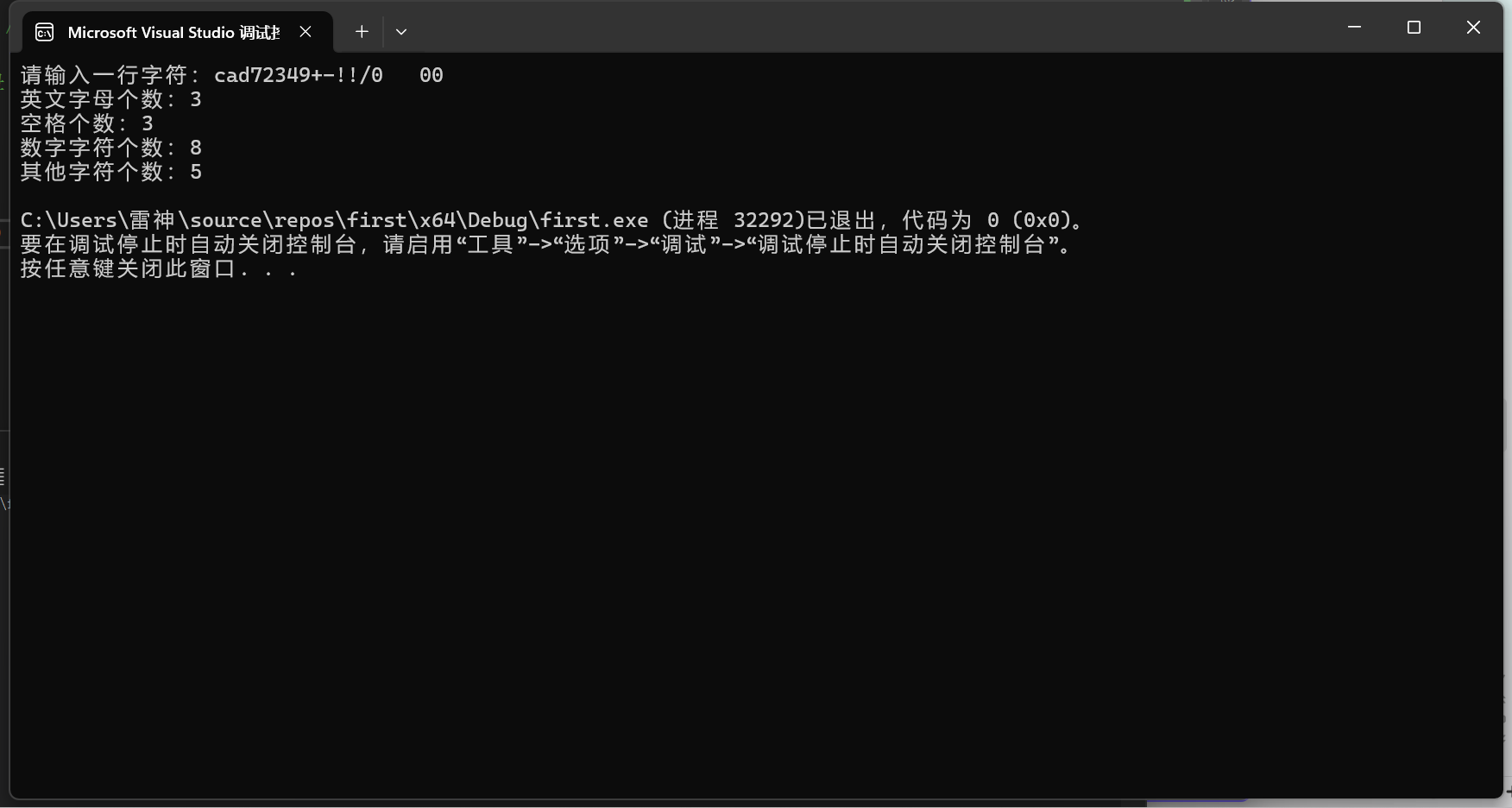
cout << "数字字符个数：" << digits << endl;

cout << "其他字符个数：" << others << endl;

return 0;

}

运行结果：



1. **代码如下：**

#include <iostream>

using namespace std;

int gcd(int a, int b) {

while (b != 0) {

int temp = b;

b = a % b;

a = temp;

}

return a;

}

int lcm(int a, int b) {

return a \* b / gcd(a, b);

}

int main() {

int a, b;

cout << "请输入两个正整数：";

cin >> a >> b;

int greatestCommonDivisor = gcd(a, b);

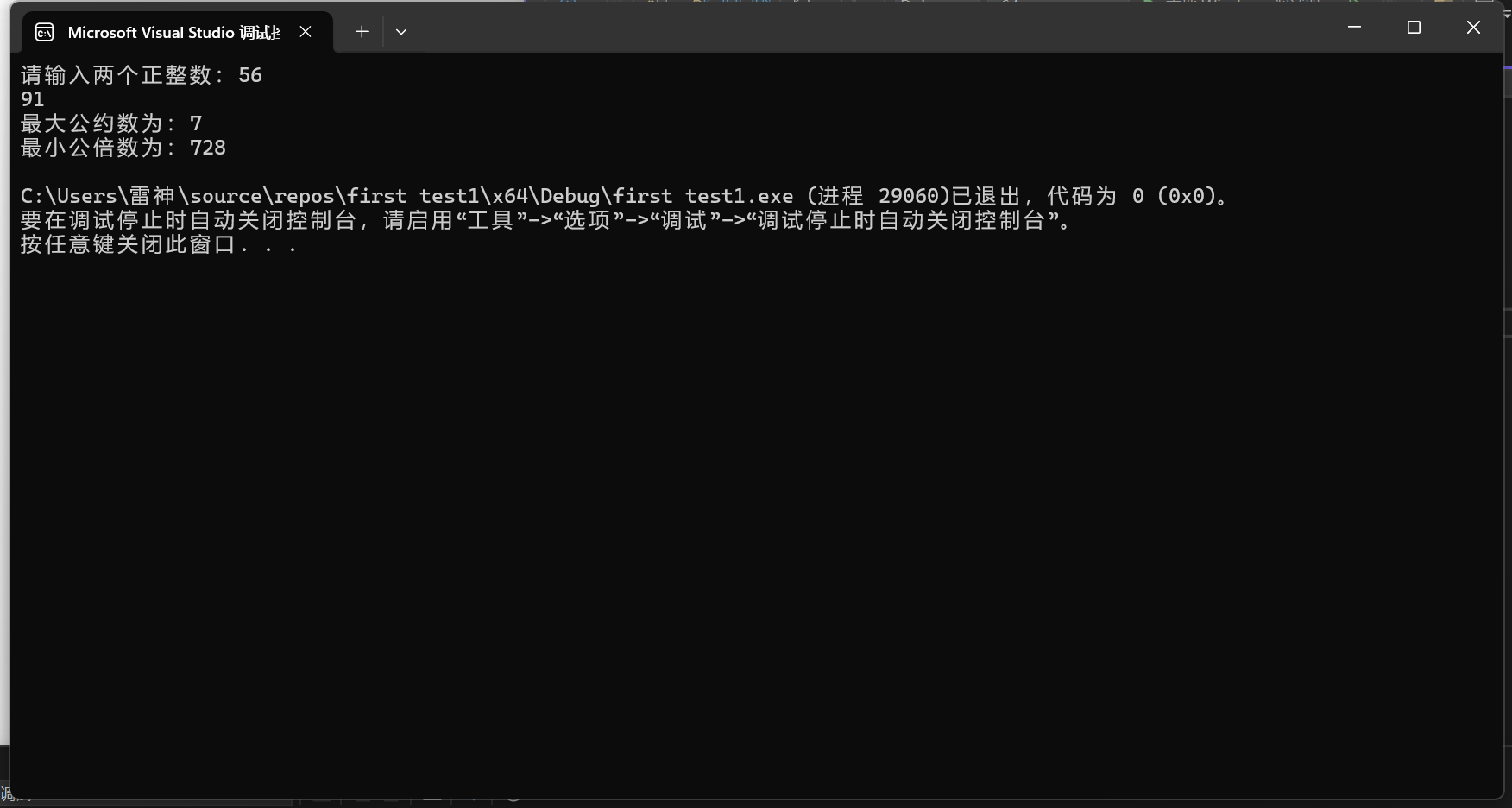
int leastCommonMultiple = lcm(a, b);

cout << "最大公约数为：" << greatestCommonDivisor << endl;

cout << "最小公倍数为：" << leastCommonMultiple << endl;

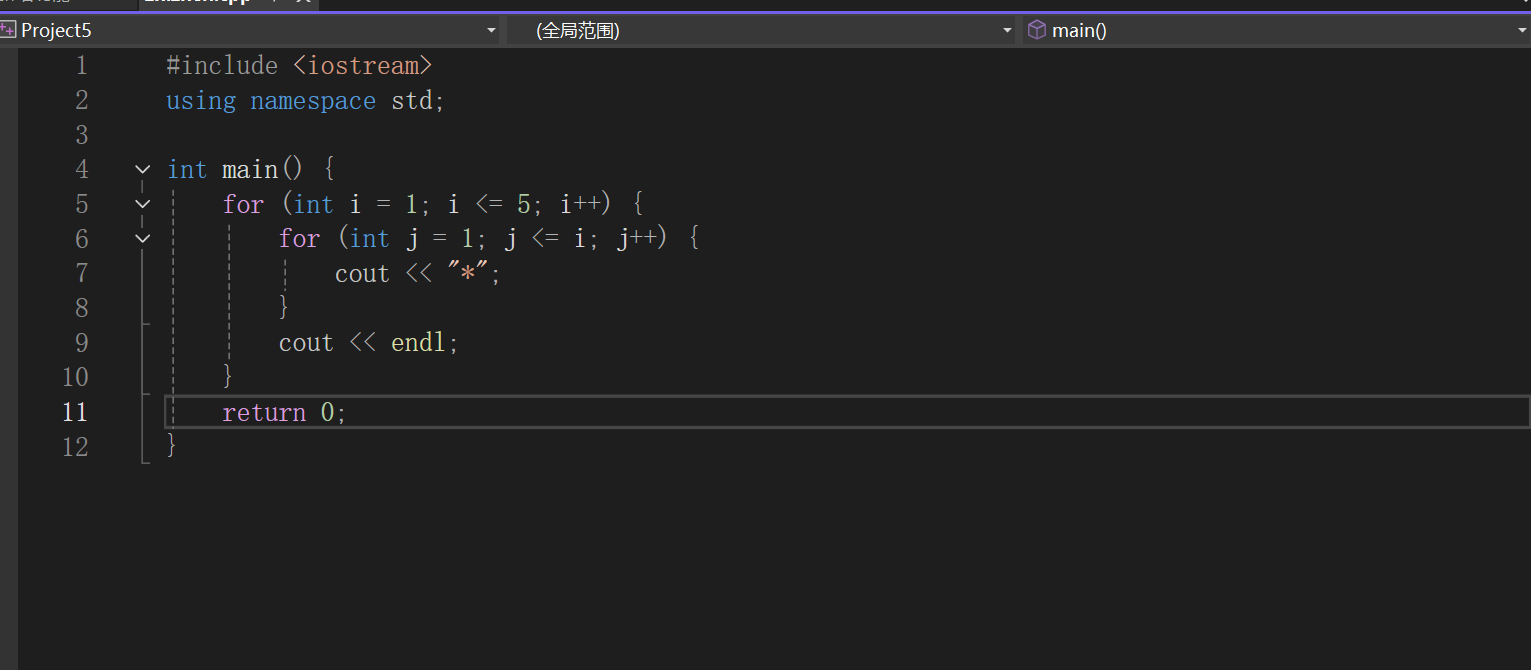
return 0;

}

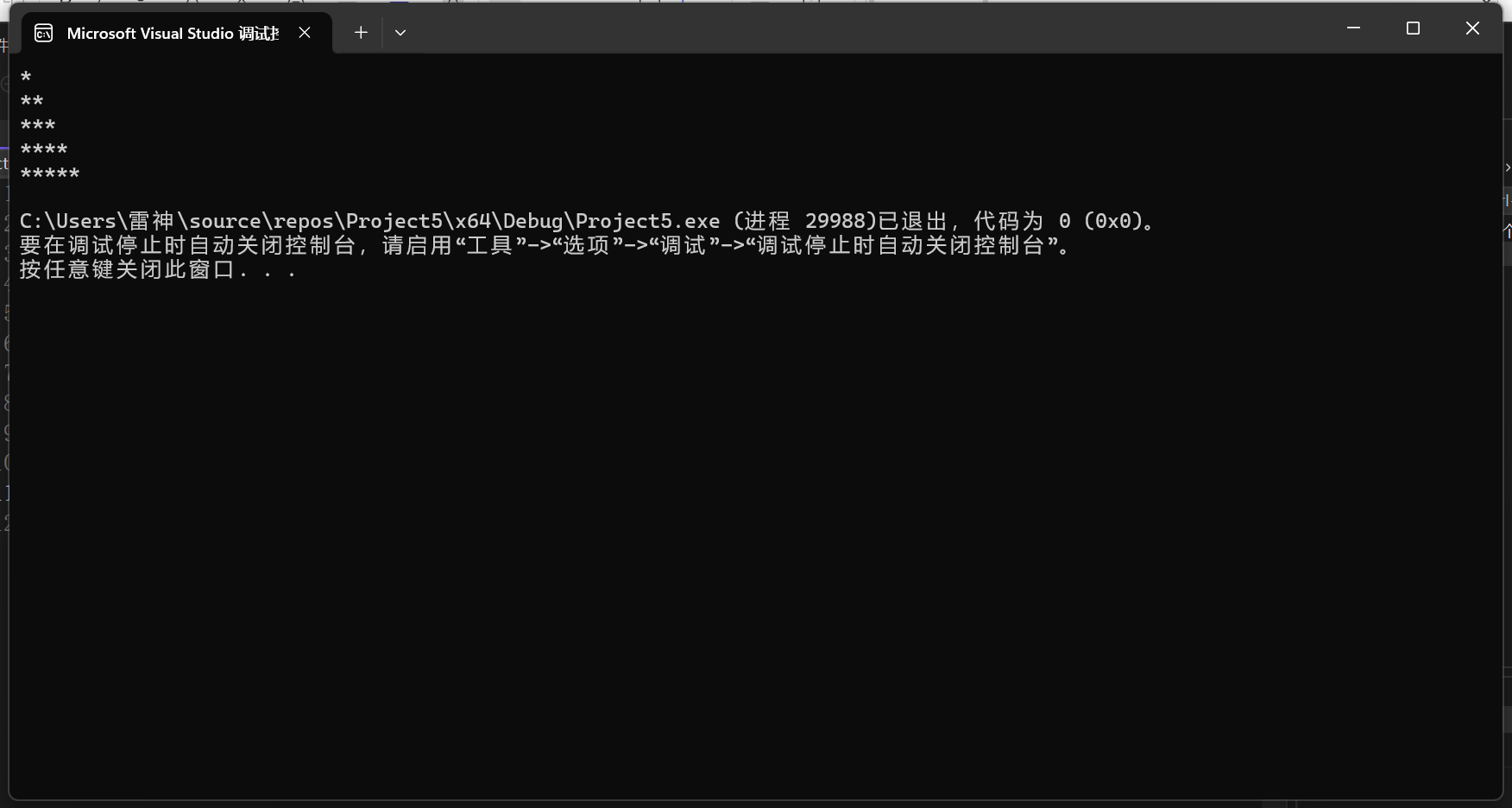
运行结果：

**7.**

**代码如下：**

****

**运行结果：**

****

**8.代码如下：**

#include<iostream>

#include<cmath>//引入fabs所需头文件

using namespace std;

int main()

{

double a, xn, xnp1;//定义a xn xn+1

cout << "请输入a的值:";//提示输入a的值

cin >> a;//输入a的值

xn = a;//将a的值赋给x

if (a >= 0)//如果a大于等于0 开方

{

for (; 1;)//死循环

{

xnp1 = (1.0 / 2) \* (xn + (a / xn));//给xn+1赋值

if (fabs(xnp1 - xn) < 0.00001)//fabs为绝对值 如果满足条件

{

cout << "a的平方根为" << xnp1;//输出当前xn+1(a的平方根)的值

break;//退出循环

}

xn = xnp1;//将此次xn+1的值赋给下次的xn

}

}

else//如果a小于0

{

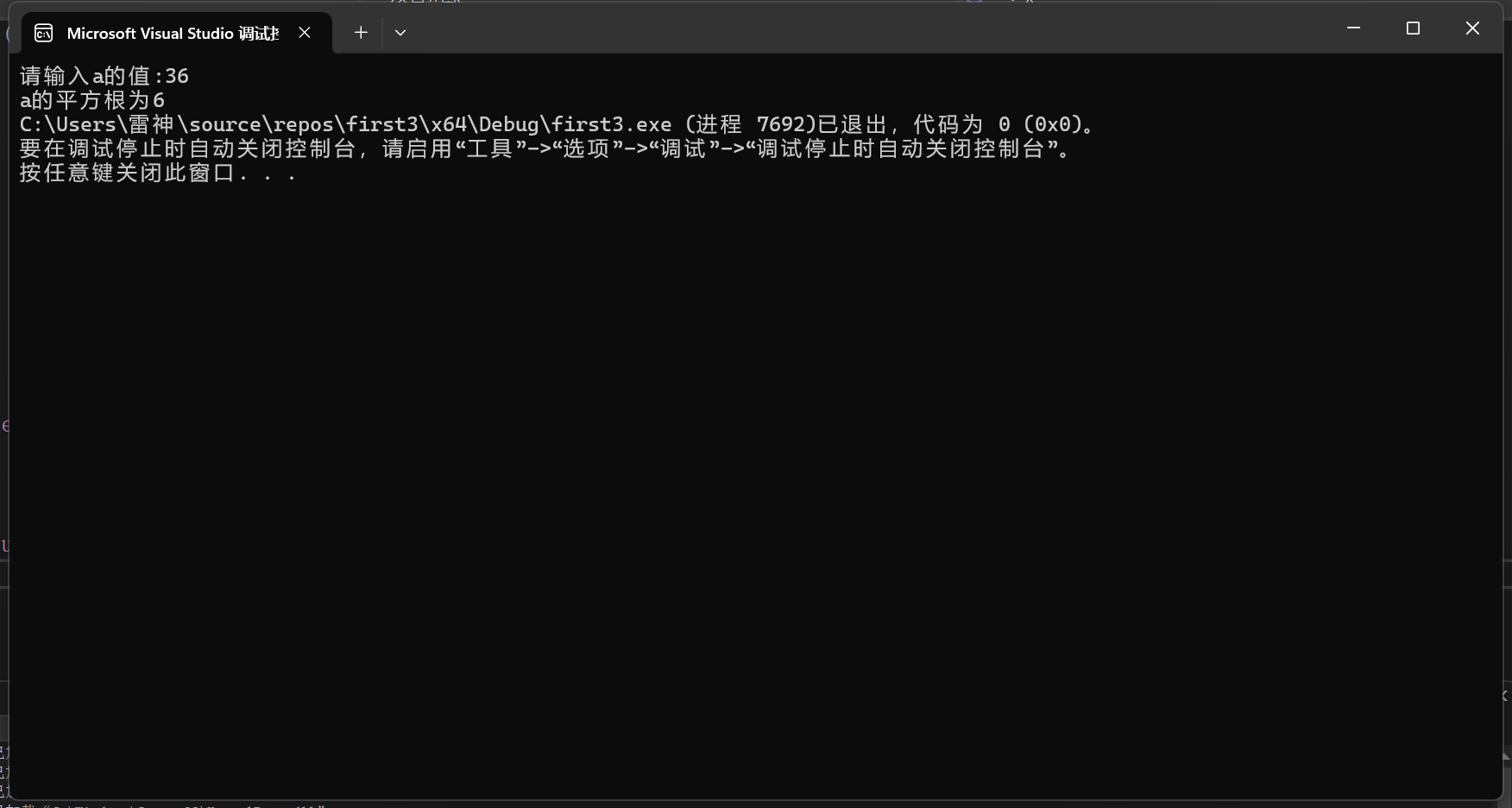
cout << "开平方需要a>=0" << endl;//提示不能开方

}

return 0;

}

**运行结果：**

****

**9.代码如下：**

#include <iostream>

using namespace std;

int main() {

const double price\_per\_apple = 0.8; // 每个苹果的价格

int day = 1; // 从第一天开始

int apples\_today = 2; // 第一天买2个苹果

int total\_apples = 0; // 总苹果数

double total\_cost = 0; // 总花费

// 计算每天购买的苹果数并更新总数和总花费

while (total\_apples + apples\_today <= 100) {

total\_apples += apples\_today; // 更新总苹果数

total\_cost += apples\_today \* price\_per\_apple; // 更新总花费

// 输出每天的信息

cout << "Day " << day << ": bought " << apples\_today << " apples, total apples: "

<< total\_apples << ", total cost: " << total\_cost << endl;

// 更新为前天的两倍，下一天的苹果数

apples\_today \*= 2;

day++;

}

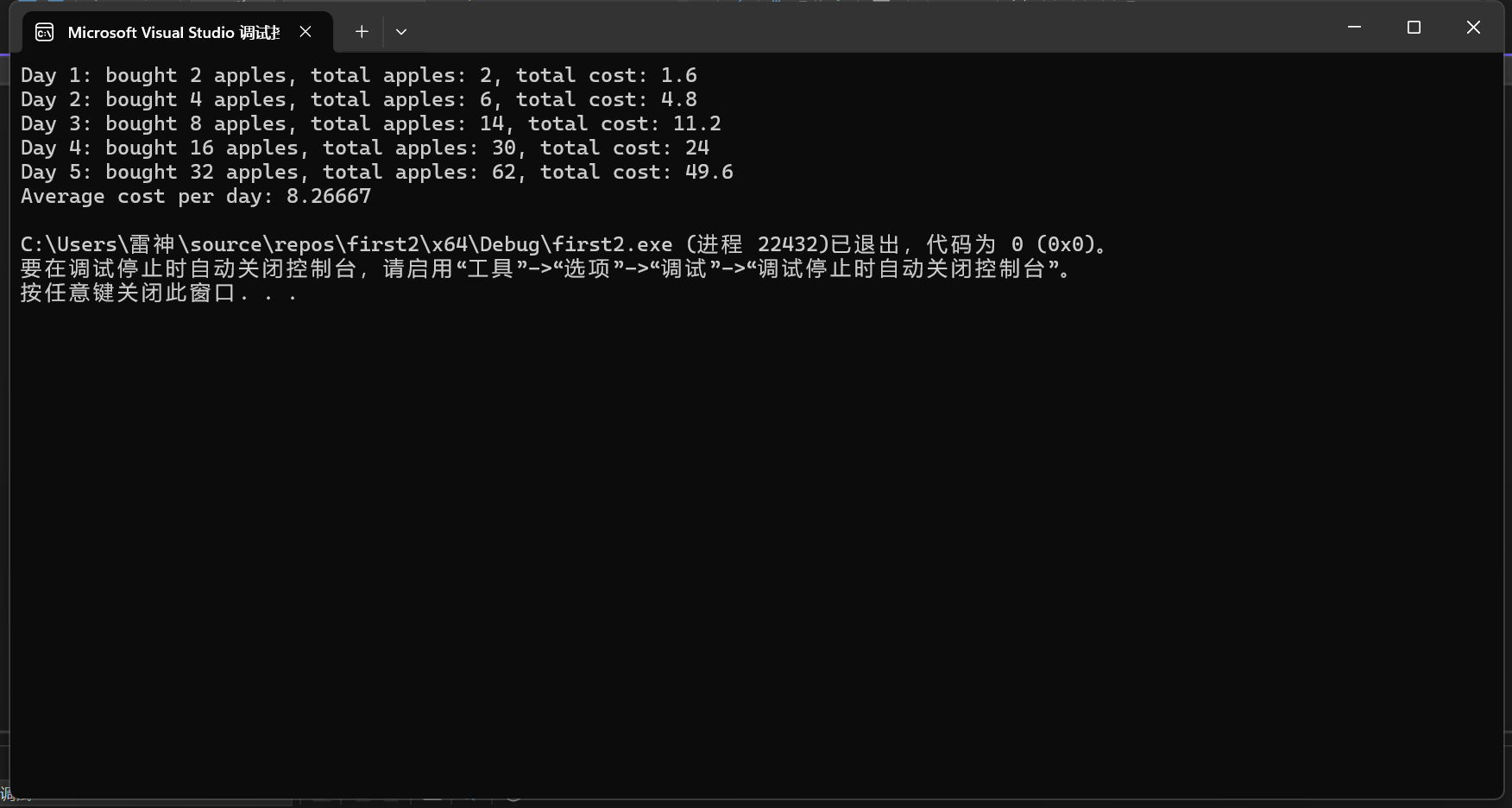
// 计算平均花费

double average\_cost = total\_cost / day;

cout << "Average cost per day: " << average\_cost << endl;

return 0;

}

运行结果：

1. **遇到的问题与解决方法**

**1.在完成第二题时，if循环在实际时出现了报错**

**原因在于未定义其因变量以及else的使用出现了问题，导致程序运行时出现了问题。**

**2.**

**不了解字符串中如何录入空格**

 最开始在互联网上查询相关资料，发现可以使用cin.getline()录入空格。

在实际编程过程中，发现若想使用该函数，需要设置最大可接受字符量。在设置后，编译器报错。

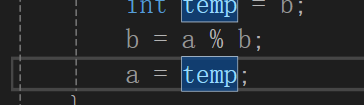
3.在编写计算器程序时遇到了逻辑错误的问题

- 运算顺序错误：没有正确处理运算符优先级。比如在计算“1 + 2 \* 3”时，如果程序先计算加法后计算乘法，结果就会出错。

- 边界情况处理不当：例如除数为0的情况。在做除法运算时，如果没有考虑除数为0，程序就会崩溃。像 a = 5 / 0 这样的代码在大多数编程语言中都会导致程序异常终止。

1. 在计算第六题时，最大公约数和最小公倍数的计算方法不是特别清楚

通过查询以及询问同学发现我们可以通过设置一个局部变量temp 将其先设置为其中一个变量的值，在进行计算，最终将这个局部变量的值转化为另一个变量的值。作为中间值的变量，可以在求解最大公约数时进项操作，将前一个运算的余数赋值给下一循环当中。



**五、体会**

**在学习C++的循环结构过程中，我深刻体会到了循环的强大和灵活性。循环是编程中的基础概念之一，它可以高效地执行重复性操作，极大地提高代码的简洁性和可维护性。在C++中，主要有三种循环结构：`for` 循环、`while` 循环和 `do-while` 循环，每种循环都适用于不同的场景。**

**首先，`for` 循环通常用于已知循环次数的场景，比如遍历数组或计算固定次数的任务。它结构紧凑、可读性强，非常适合控制较严格的循环。其次，`while` 循环更灵活，可以在循环条件不确定的情况下使用，比如读取输入直到特定条件出现为止。而 `do-while` 循环则确保循环体至少执行一次，适用于需要先执行后判断的情形。**

**通过练习循环，我逐渐理解了它们在实际问题解决中的重要性。例如，在数据统计、搜索、排序等操作中，循环使得我们能够灵活高效地处理大量数据。不过，使用循环时也要注意避免死循环和优化效率，否则可能会导致性能问题。总的来说，循环结构是C++编程的核心技能之一，它为解决复杂问题提供了很大的便利。**