**计算机程序设计基础（C++)**

**实验报告**

专业班级： 软工2405

学 号： 8209240518

姓 名： 曹雨佳

**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

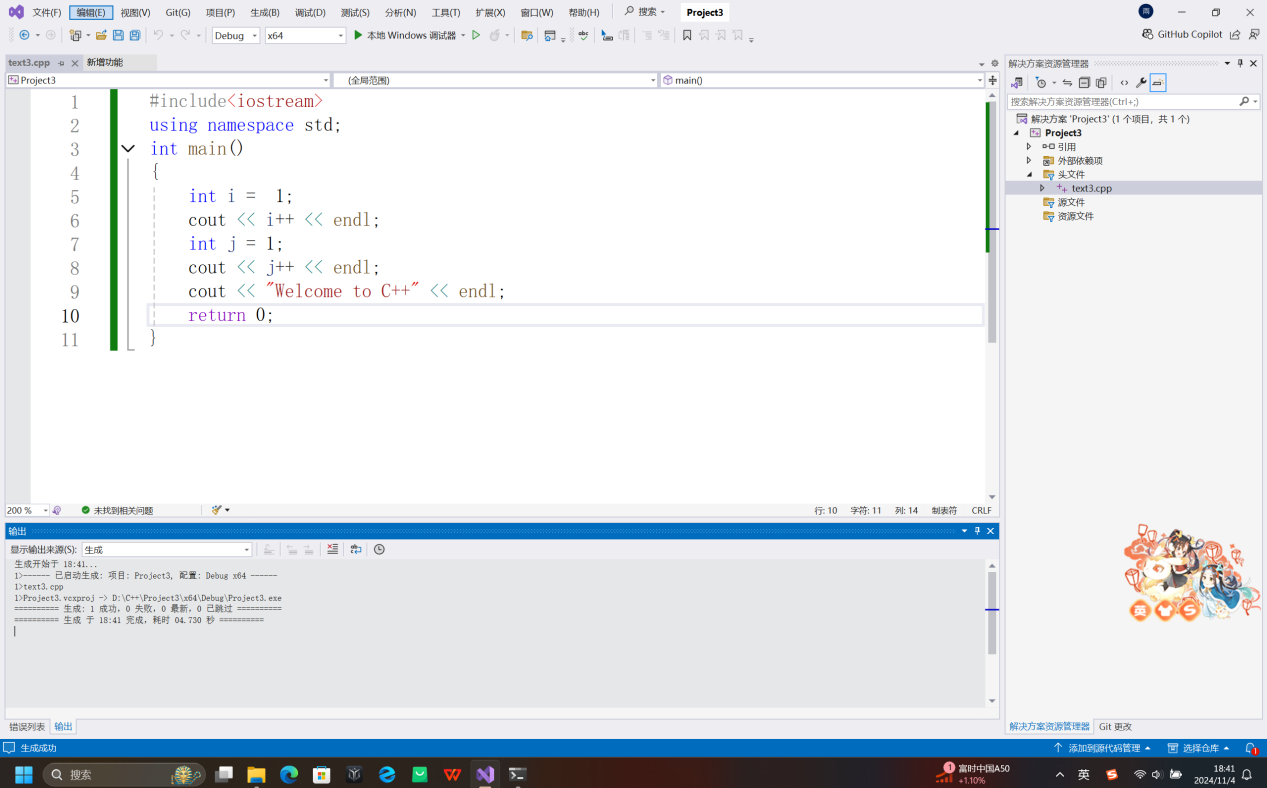
system("pause");

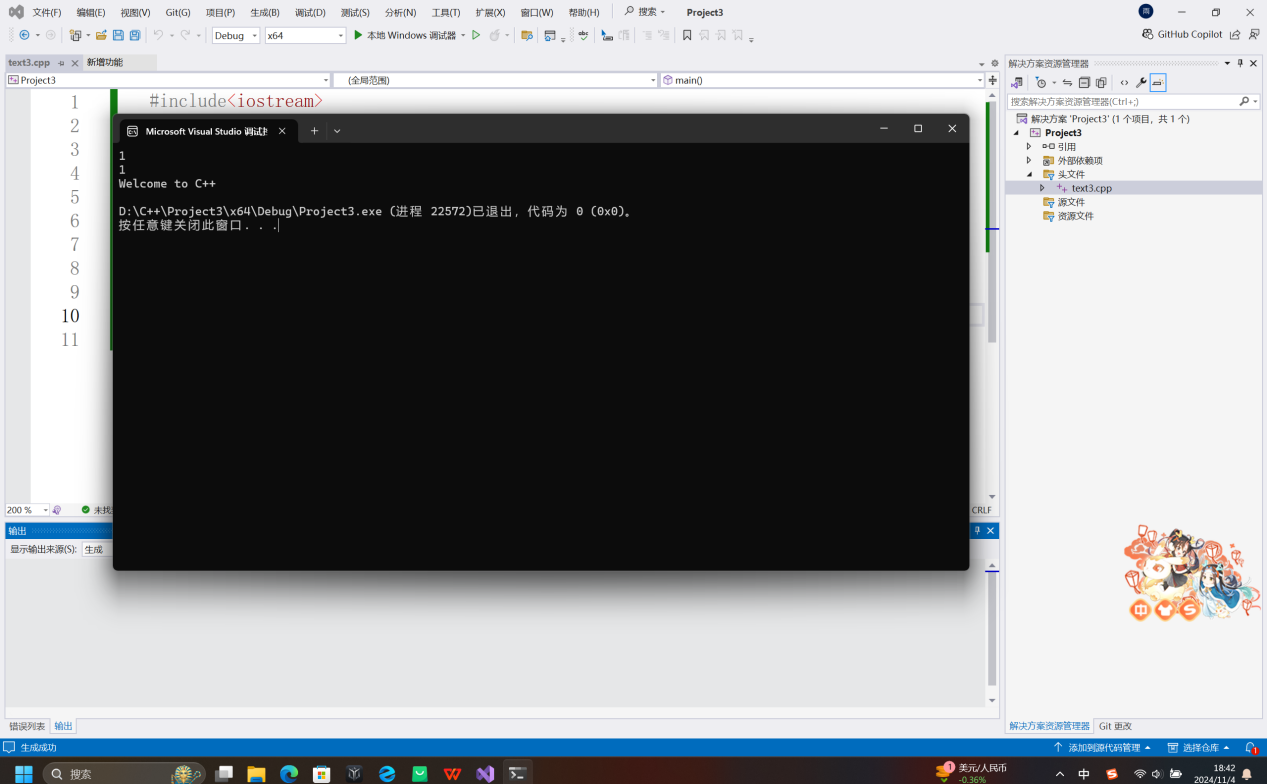
return 0;

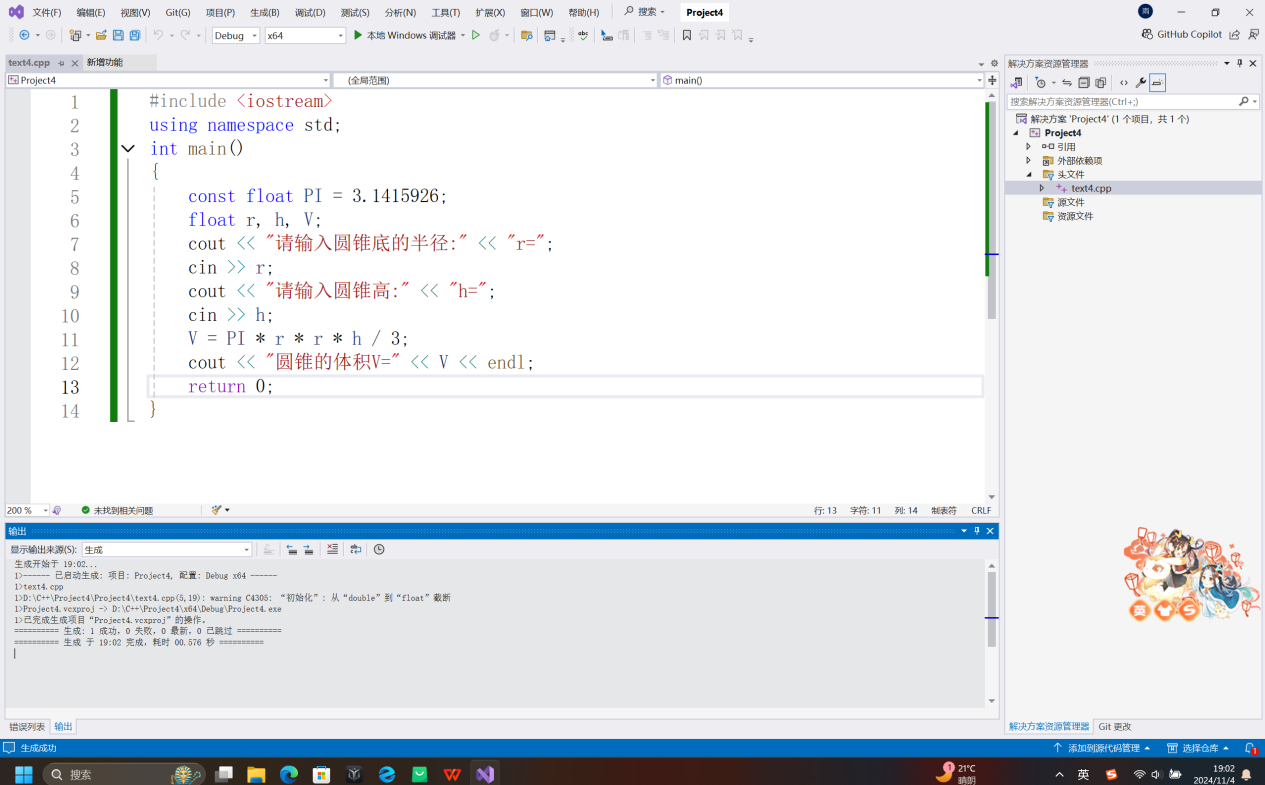
**自己编程测试一下将testUnint按8进制输出<<oct;je\_将一个实数转换成int,观察结果。**

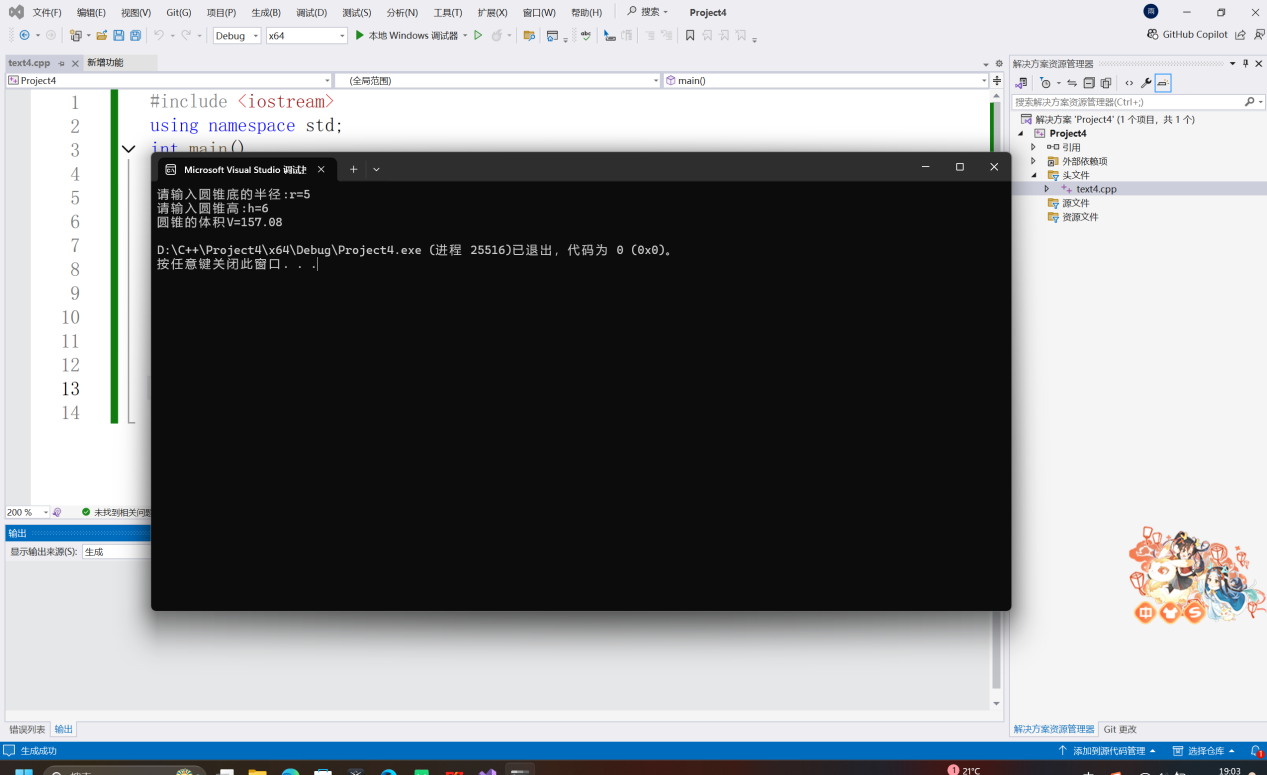
**5.编程，输入华氏温度，将其转换为摄氏温度后输出（保留两位小数）。**

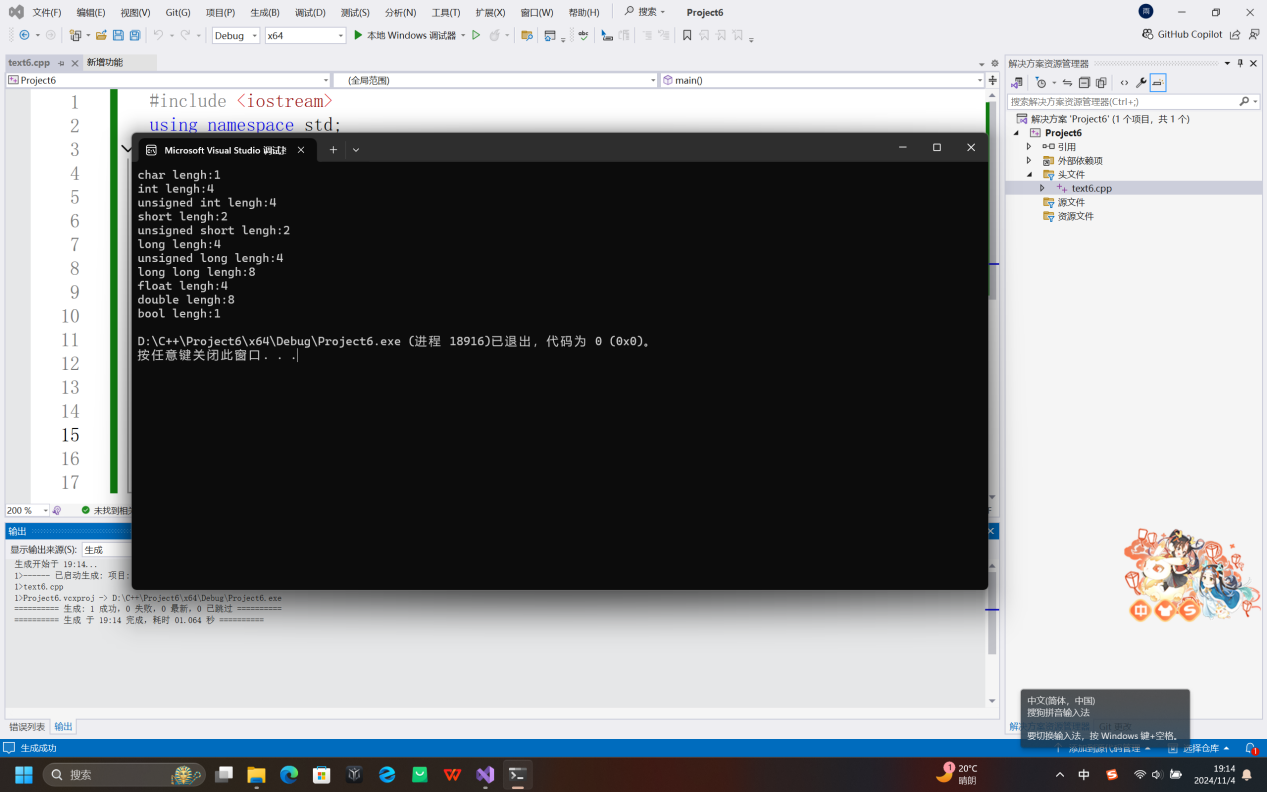
**三、算法分析，程序结果**

1.

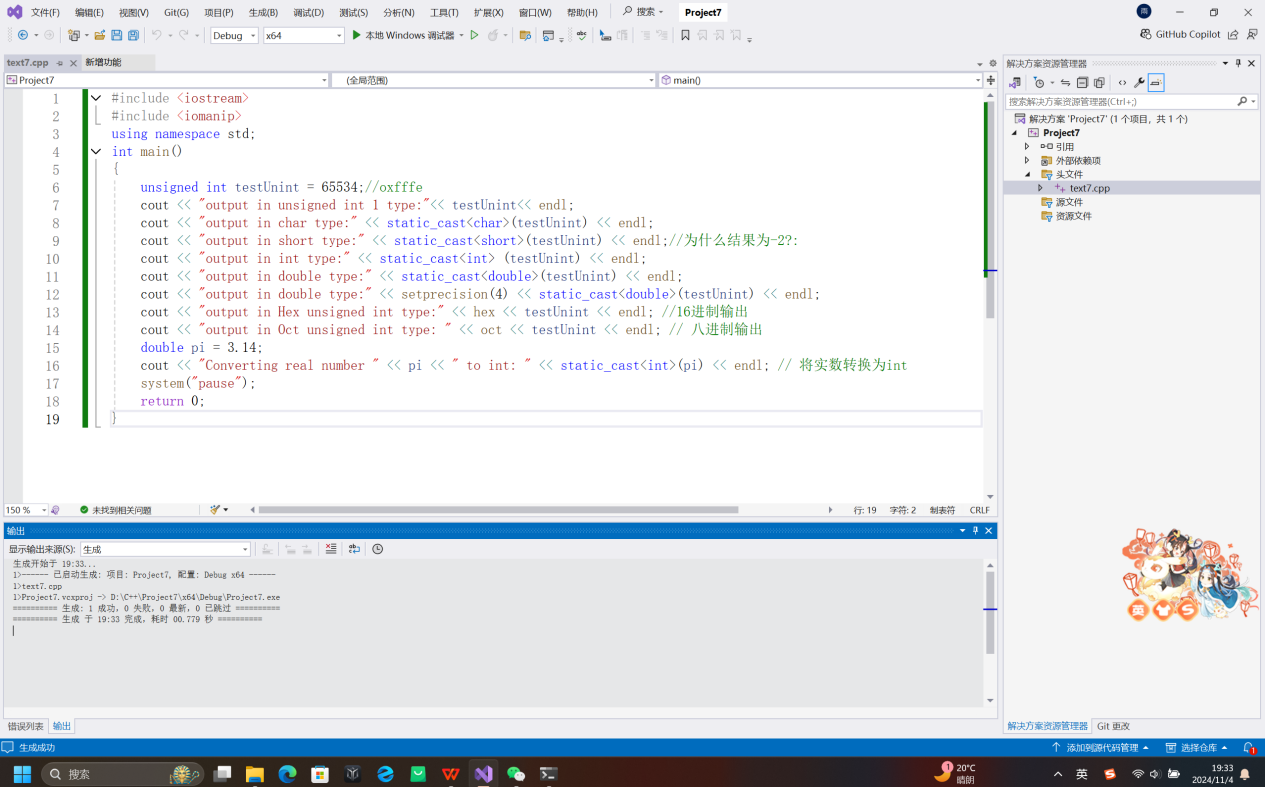


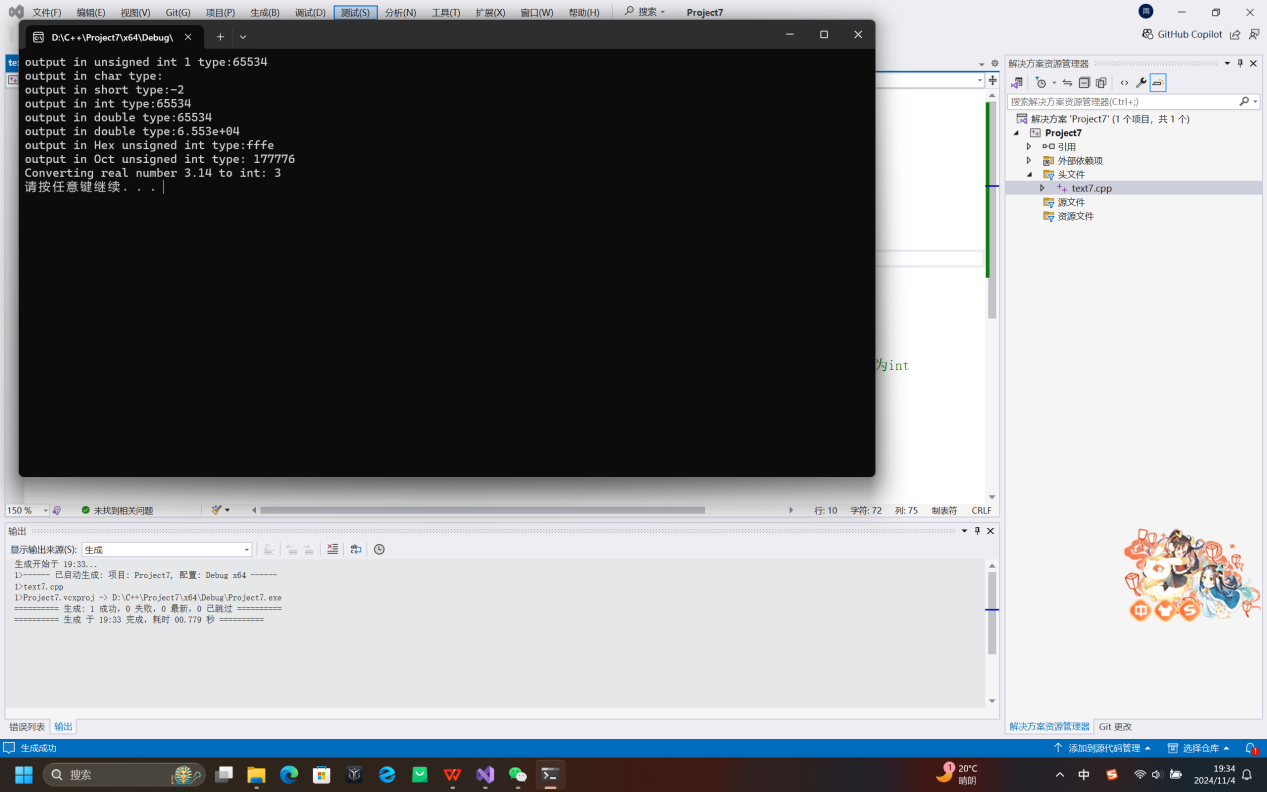
2.

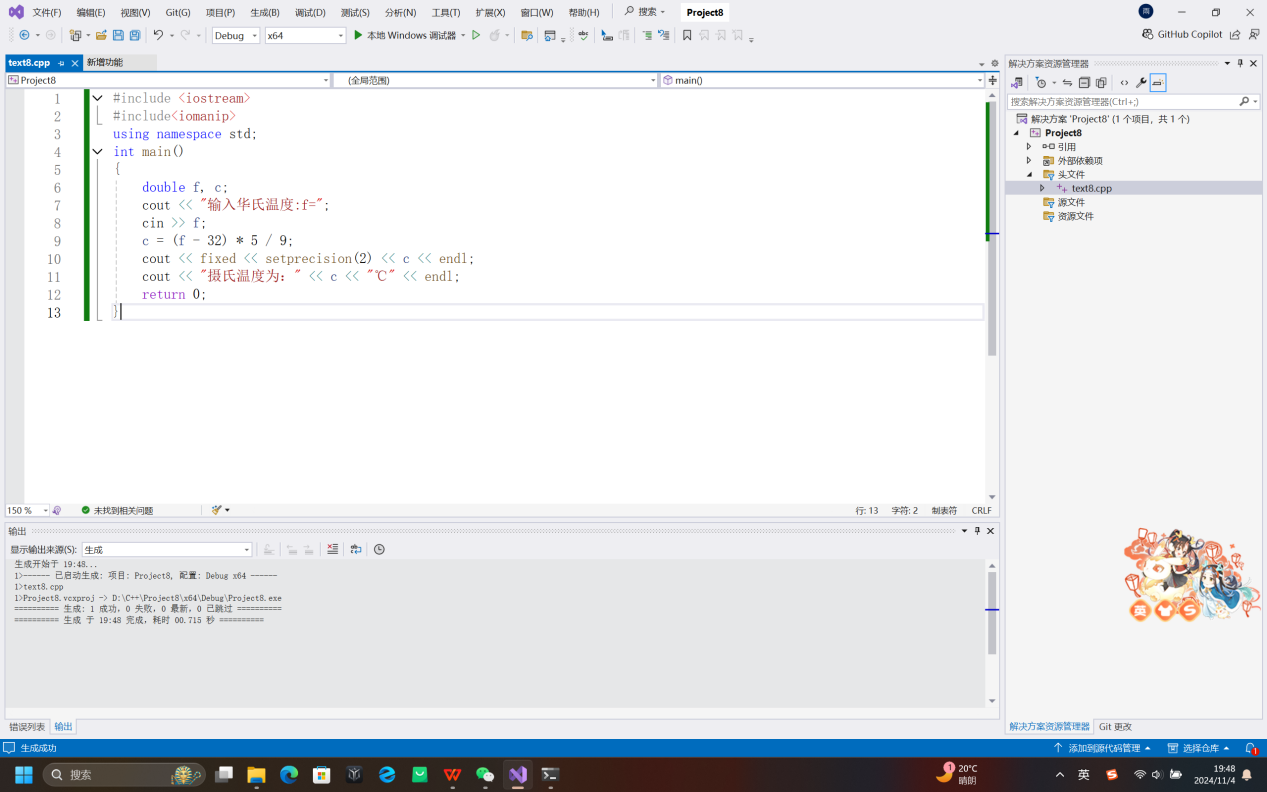


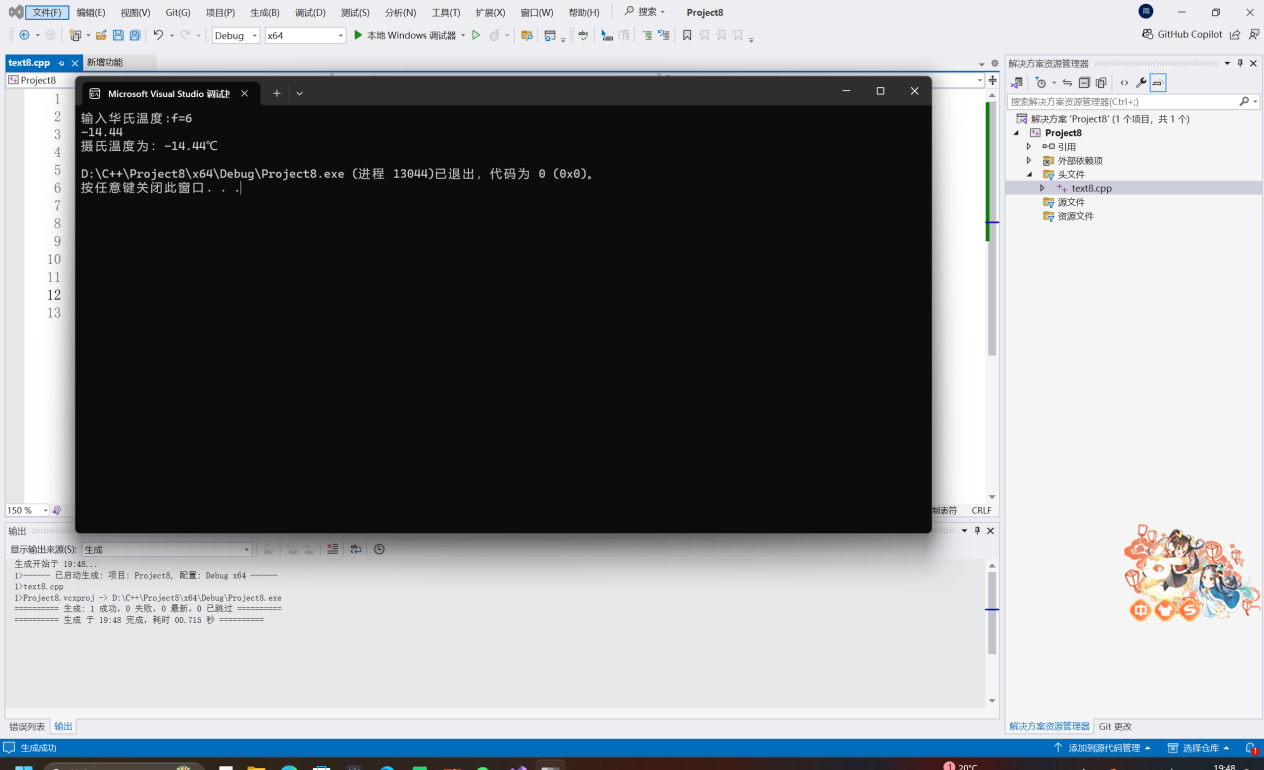
3.

4.





5.

****

1. **遇到的问题与解决方法**

1. 变量重复定义错误：在代码 Int i = k + 1; 和 int i = 1; 中，出现变量 i 重复定义的情况，并且 k 未定义。解决方法是删除 Int i = k + 1; 这一行，因为在后面已经重新定义了 i ，并且确保使用的变量都有正确的定义和初始化。

2. 数据类型范围理解不足：在涉及数据类型转换和运算时，对于数据超过类型表示范围的溢出情况理解不够深入。通过手动计算和查阅资料，深入学习了不同数据类型的取值范围，如 int 类型在不同编译器下的范围，以及当数据溢出时会出现的错误结果，从而在编写代码时更加注意数据范围的限制，必要时进行边界检查和处理。

3. 编译信息解读困难：初次面对编译错误信息时，难以快速准确地定位问题。通过仔细阅读错误提示信息中的行号、错误描述等内容，逐步学会理解常见的编译错误，如语法错误、变量未定义、类型不匹配等，并根据这些提示去修改代码中的错误。同时，积累了一些常见编译错误的解决经验，以便在后续编程中能够更快地解决类似问题。

1. **体会**

通过本次实验，我对 C++编程有了更深入的理解与掌握。在熟悉集成开发环境时，深刻体会到一个良好的开发工具对编程效率的提升作用。如在 VS 中，代码的编辑、编译、调试等功能集成度高，方便快捷。

对于 C++程序基本要素的学习，认识到变量定义、常量使用以及数据类型的重要性。不同数据类型的长度和取值范围决定了数据的存储和运算方式。在处理数据类型转换时，明白了隐式转换可能带来的风险以及强制转换的必要性，特别是在混合算术运算中，数据类型转换规则的理解有助于避免错误结果。

输入输出操作让程序与用户有了交互性，而对编译信息的解读和错误修改能力在编程过程中不可或缺。每一个错误提示都是一次学习和提升的机会，通过仔细分析错误信息，能够逐步排查问题所在，从而提高代码质量。

简单程序设计部分则锻炼了逻辑思维和代码实现能力，从编写代码到调试运行成功，这个过程培养了耐心和解决问题的能力，为后续更复杂的程序开发奠定了坚实基础。

**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。

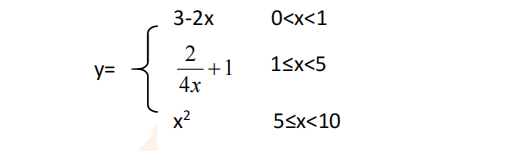
2、熟练掌握if、switch、while、do-while，for语句的语法结构与执行过程。

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：

\*

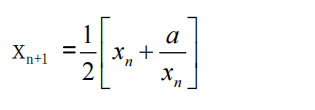
\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

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。

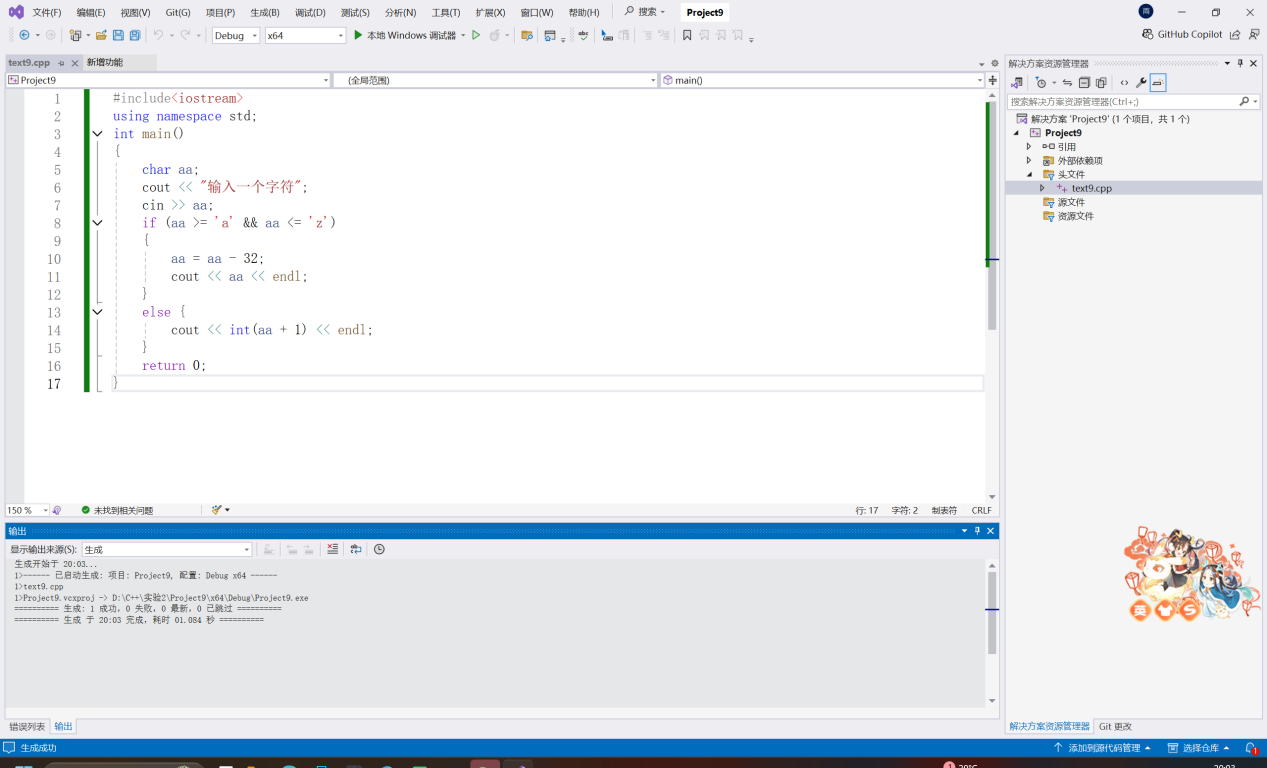
提示：迭代法是把 xn代入迭代公式右边，计算出 xn+1来，然后把 xn+1 作为新的 xn ，计算出新的 xn+1，如此重复，直到|xn+1 - xn|<10 -5 时，xn+1 为所求的平方根。可以把 a 作为 xn 的初始值。

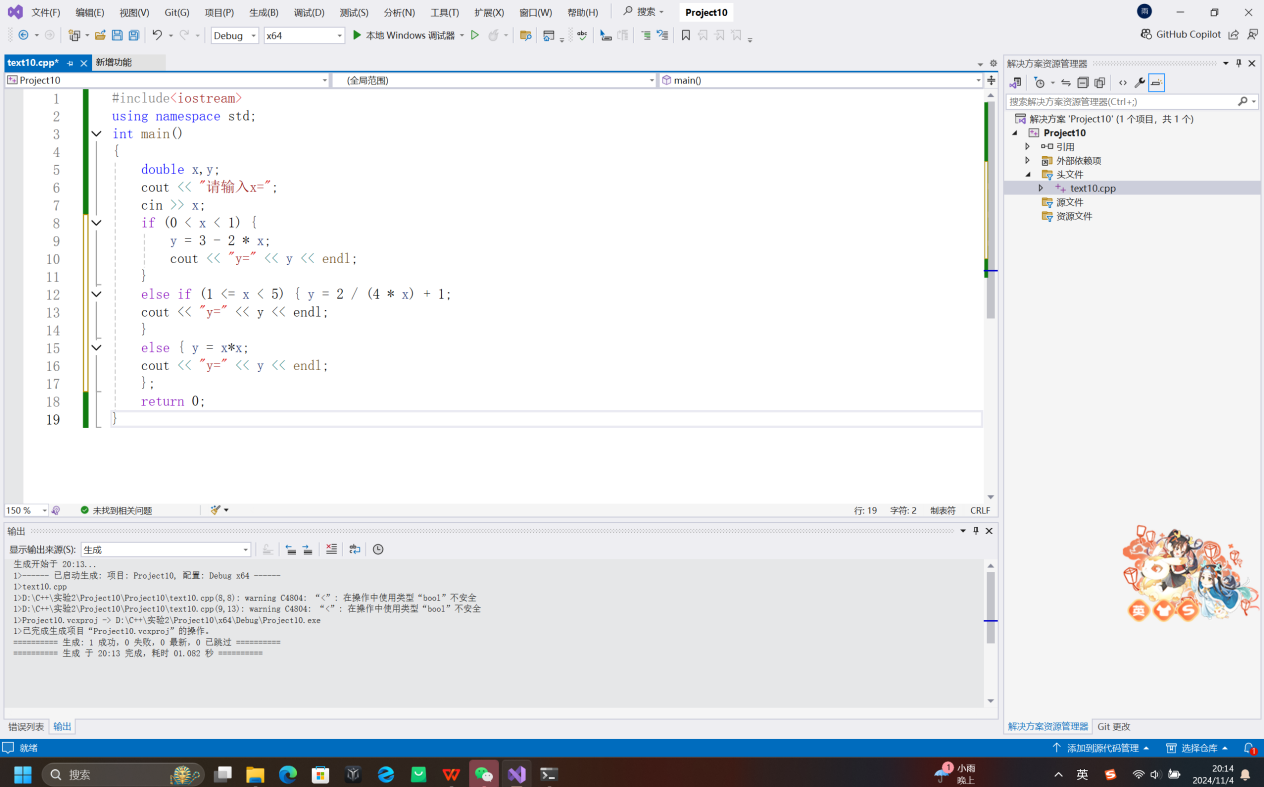
思考：（1）如果输入 a 为负，在运行时会出现什么情况? 修改程序使之能处理任何的 a 值。

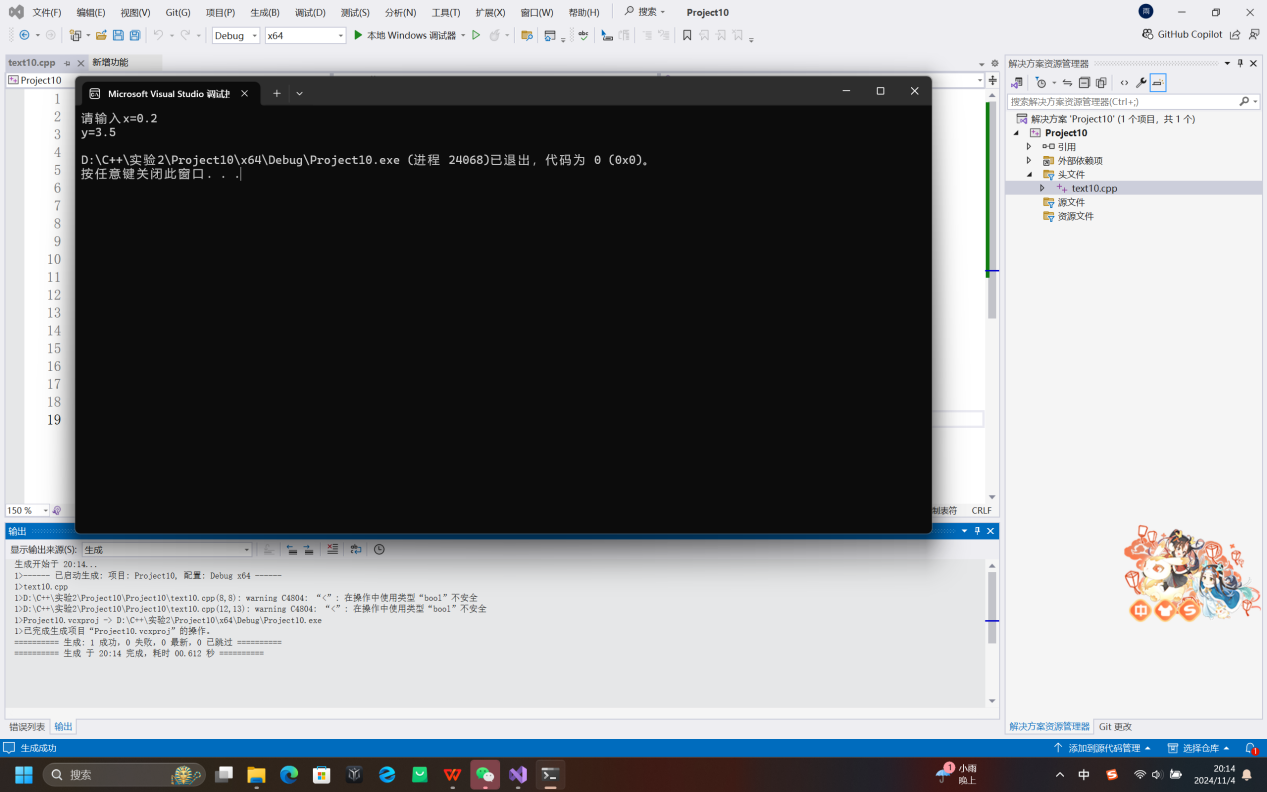
(2）能否|xn+1 -xn|<10 -10或更小? 为什么? 请试一下。

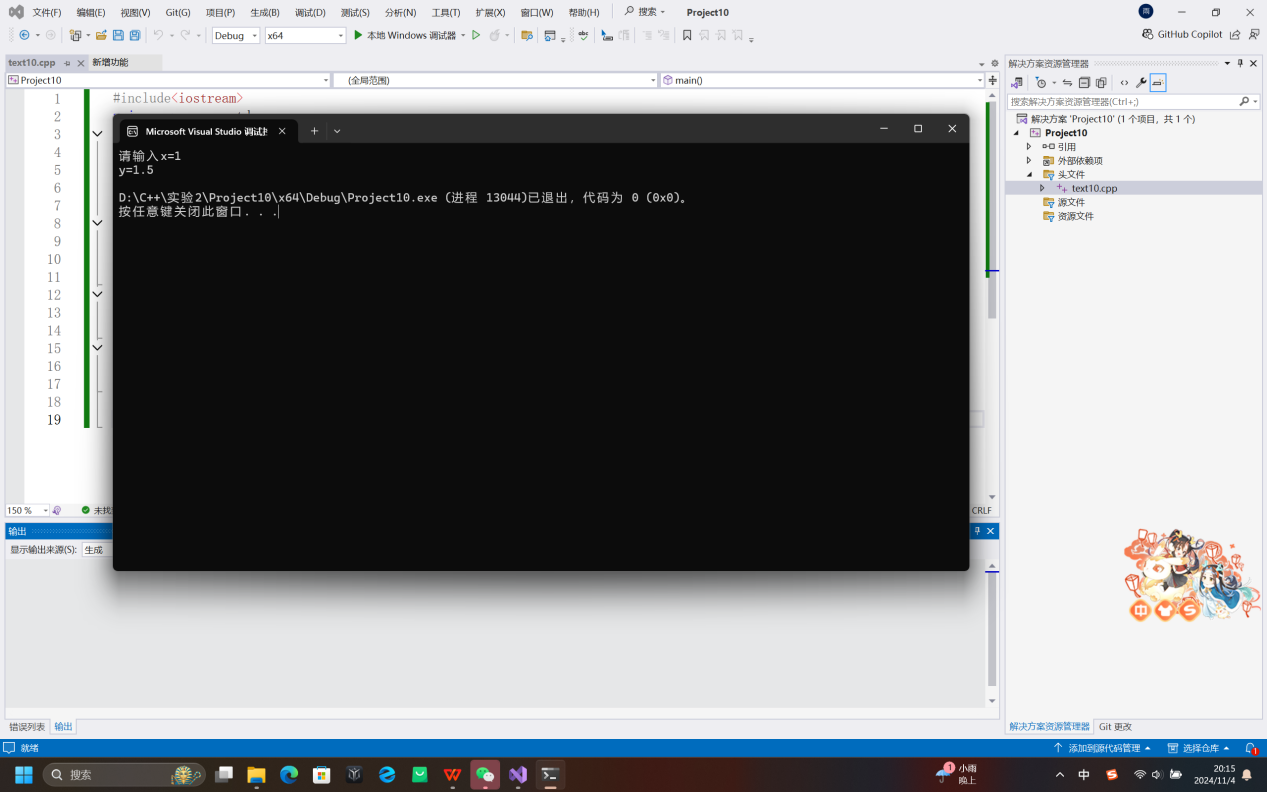
9、苹果每个 0.8 元，第一天买 2 个，第二天开始，每天买前天的 2 倍，直到购买的苹果数不超过100的最大值，求每天平均花多少钱。

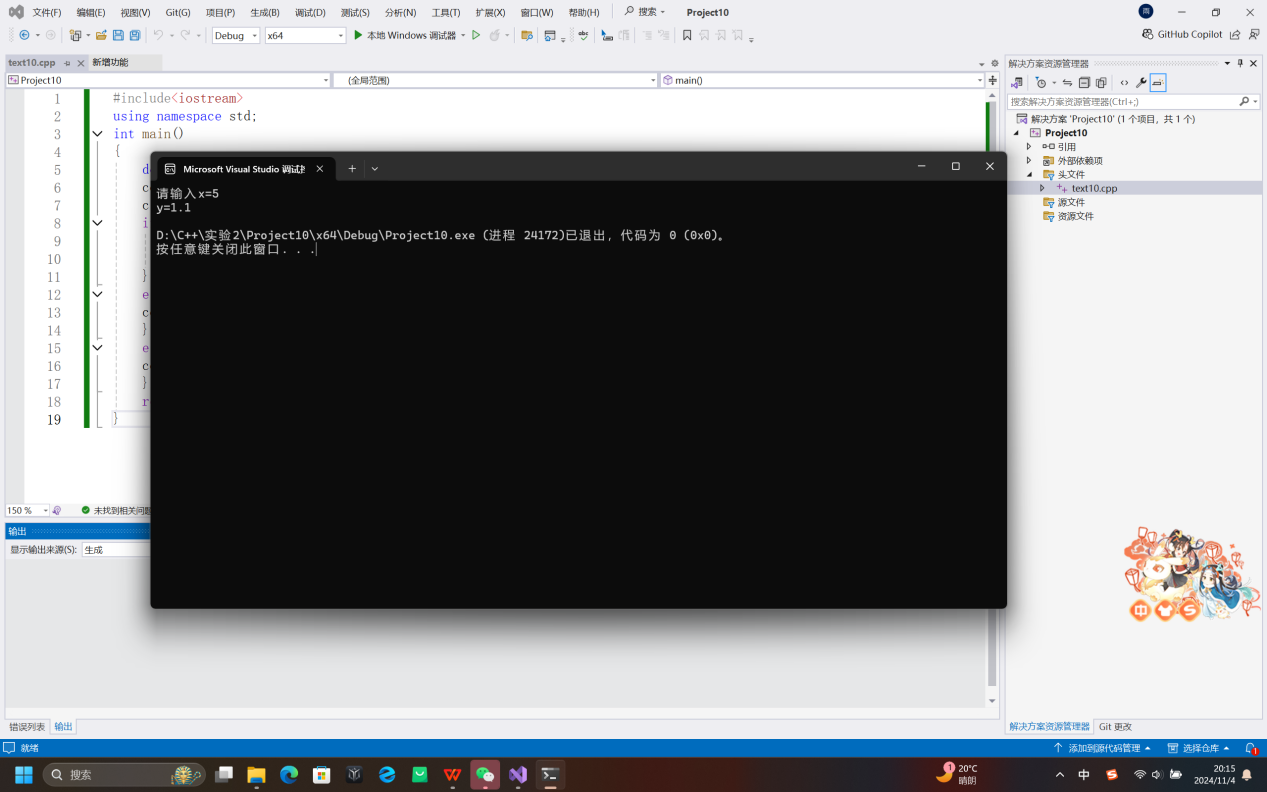
**三、算法分析，程序结果**

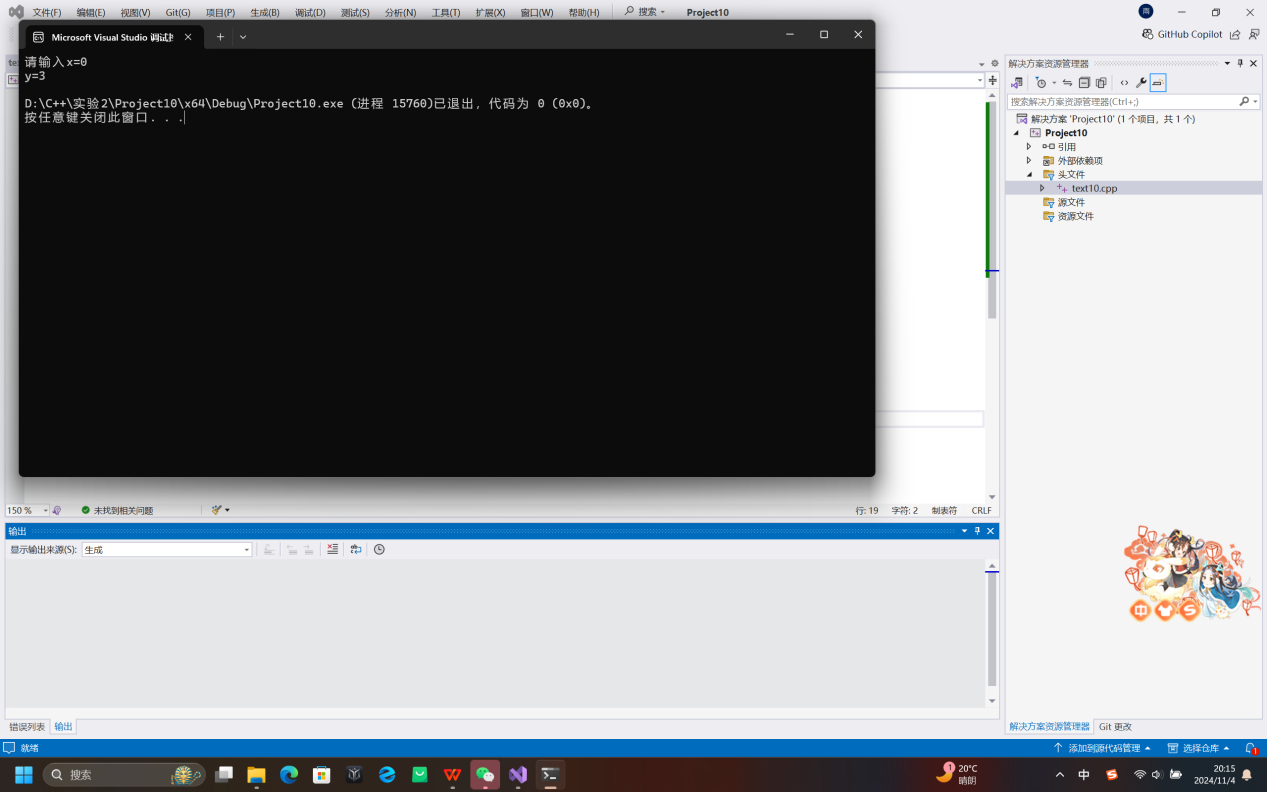
1.

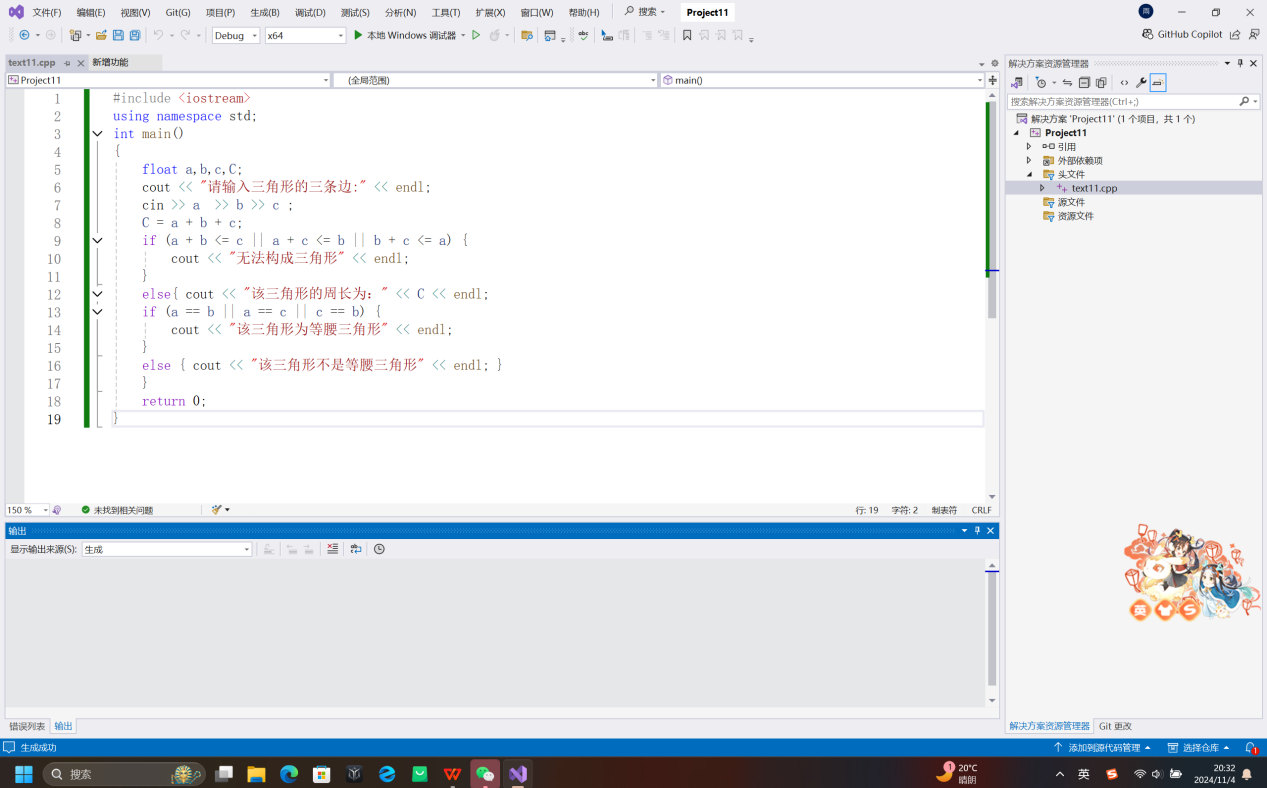
2.

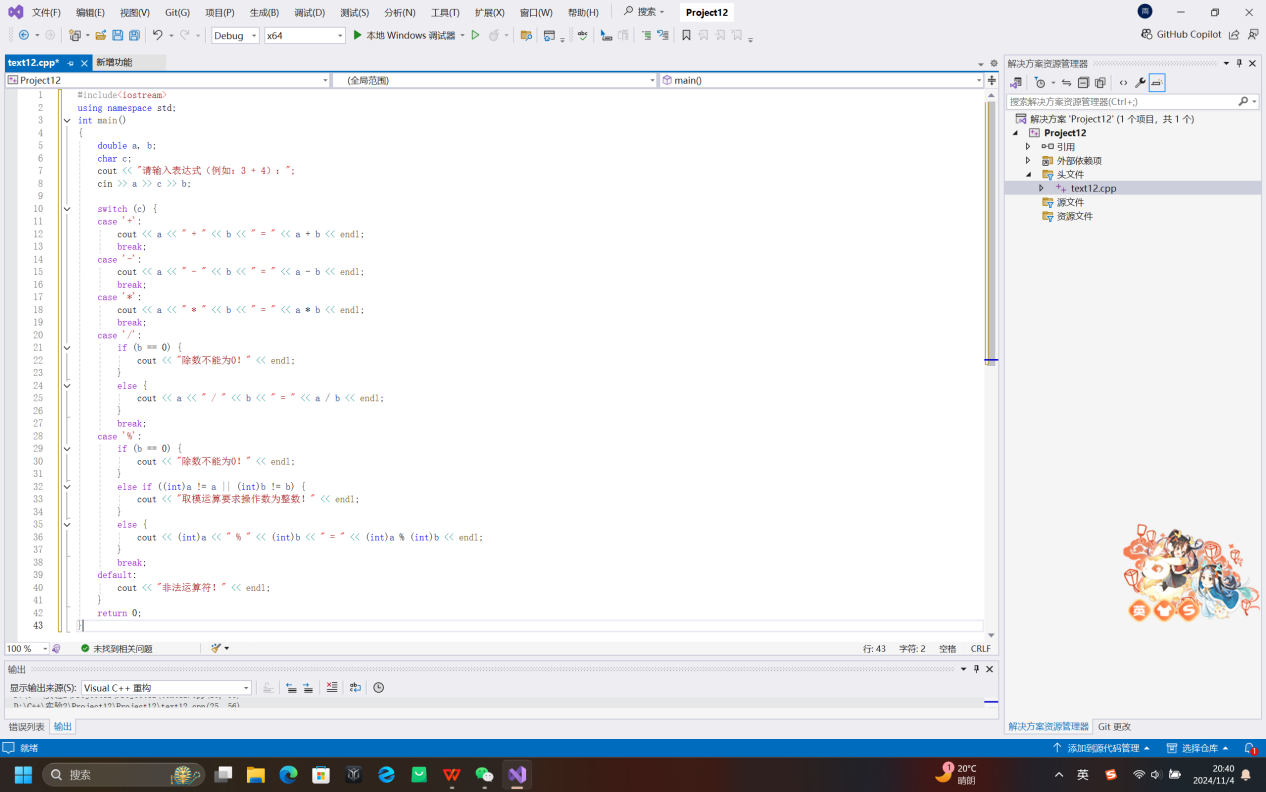


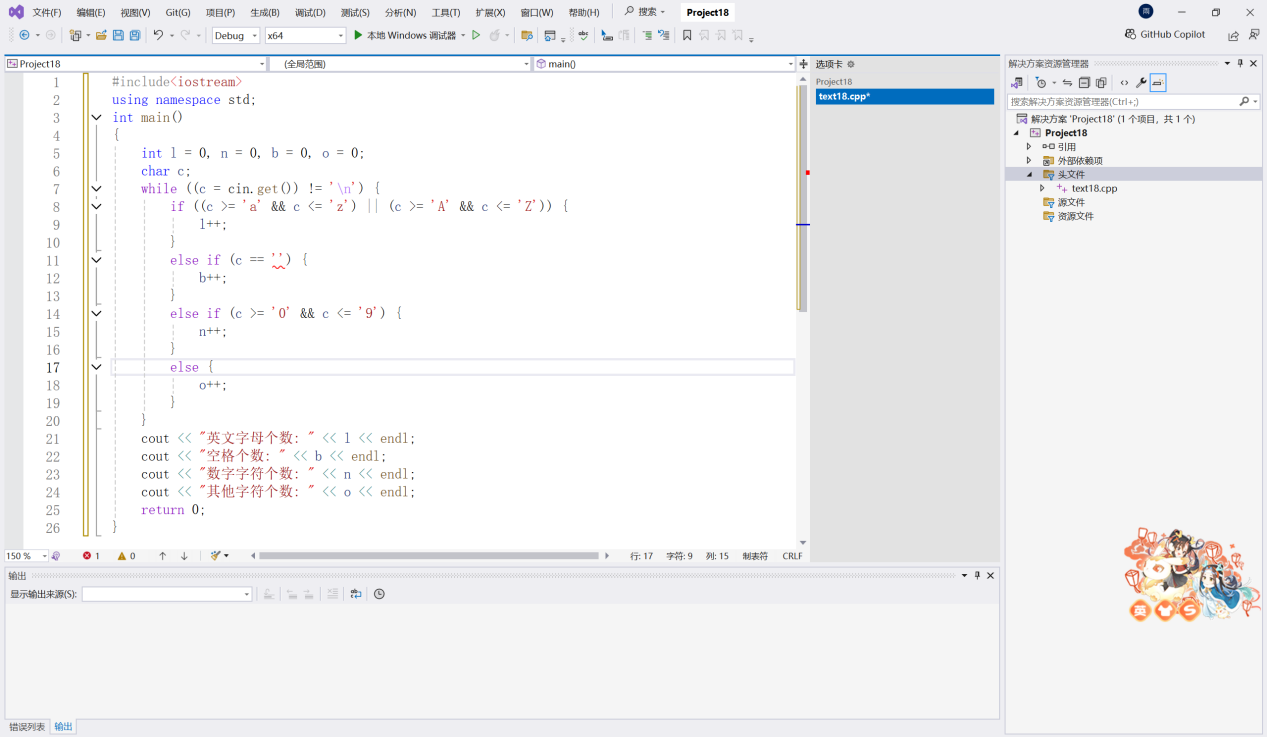


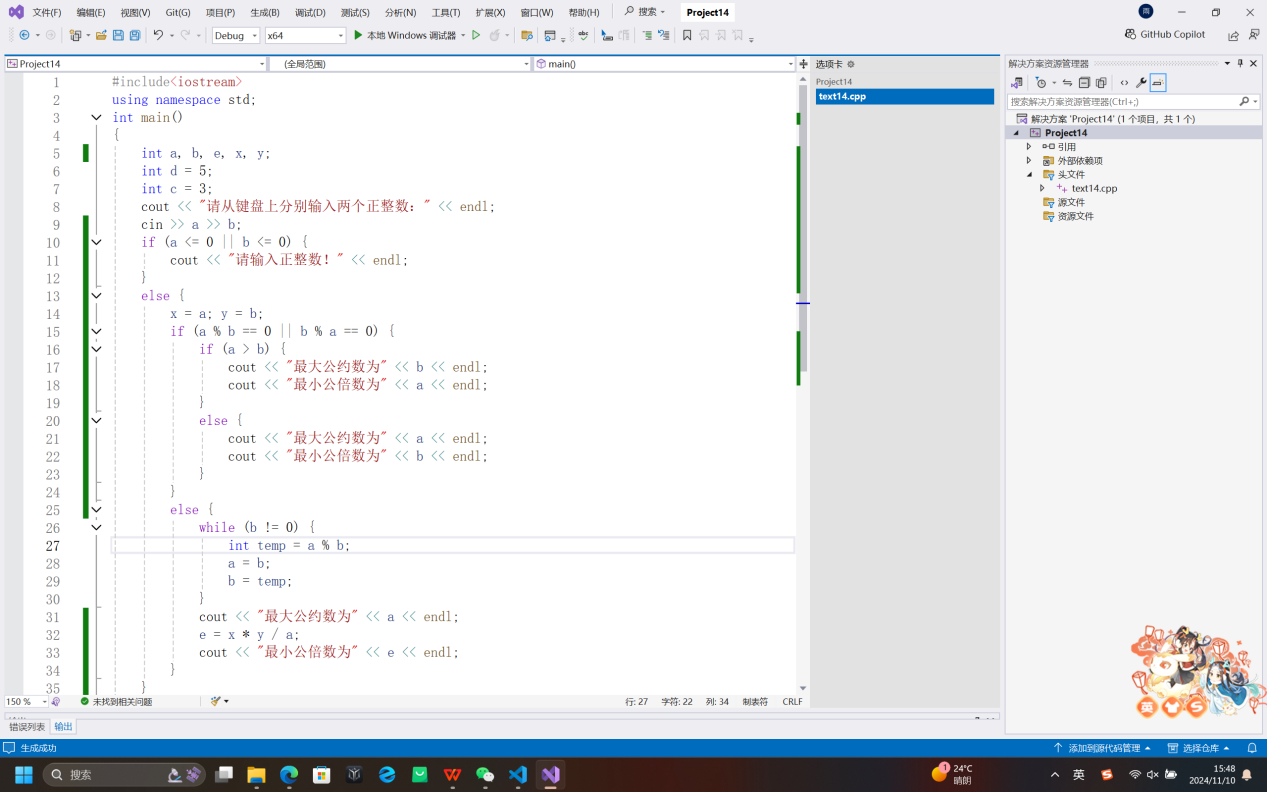




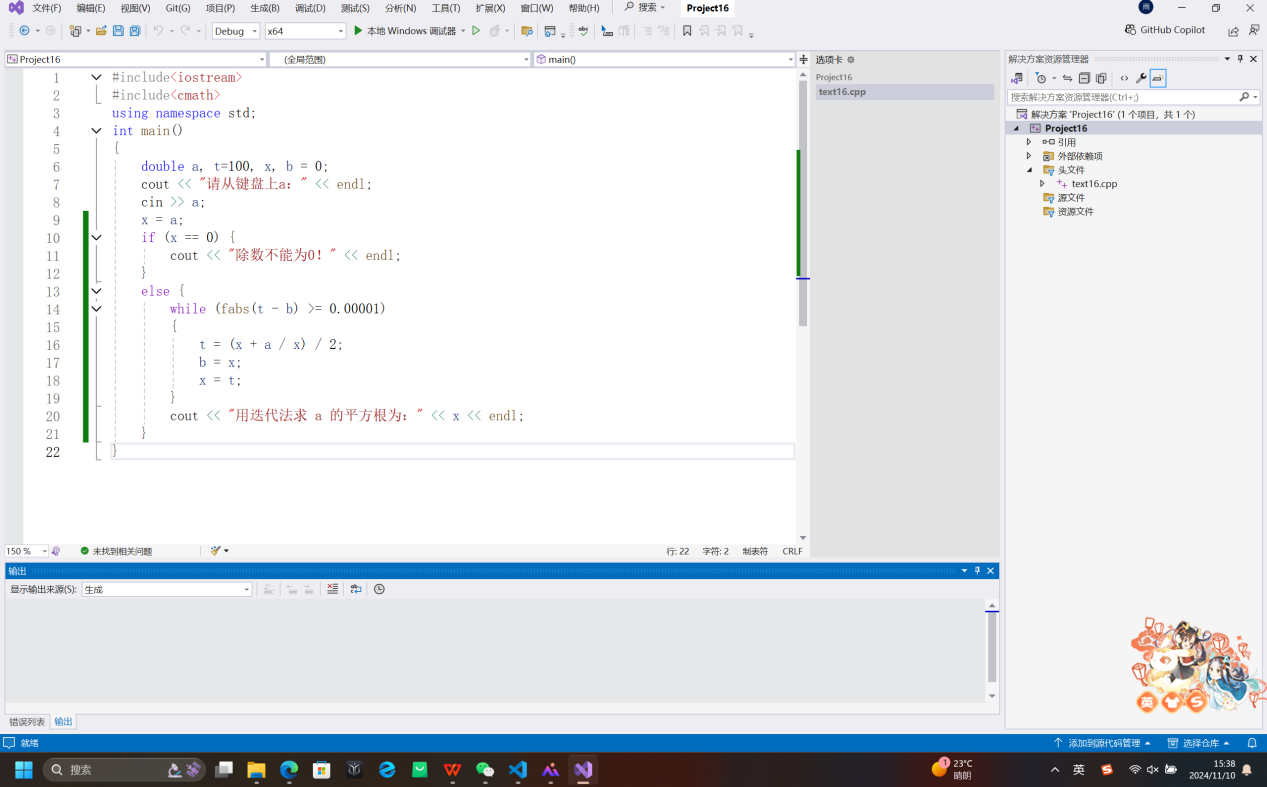
3.

4.

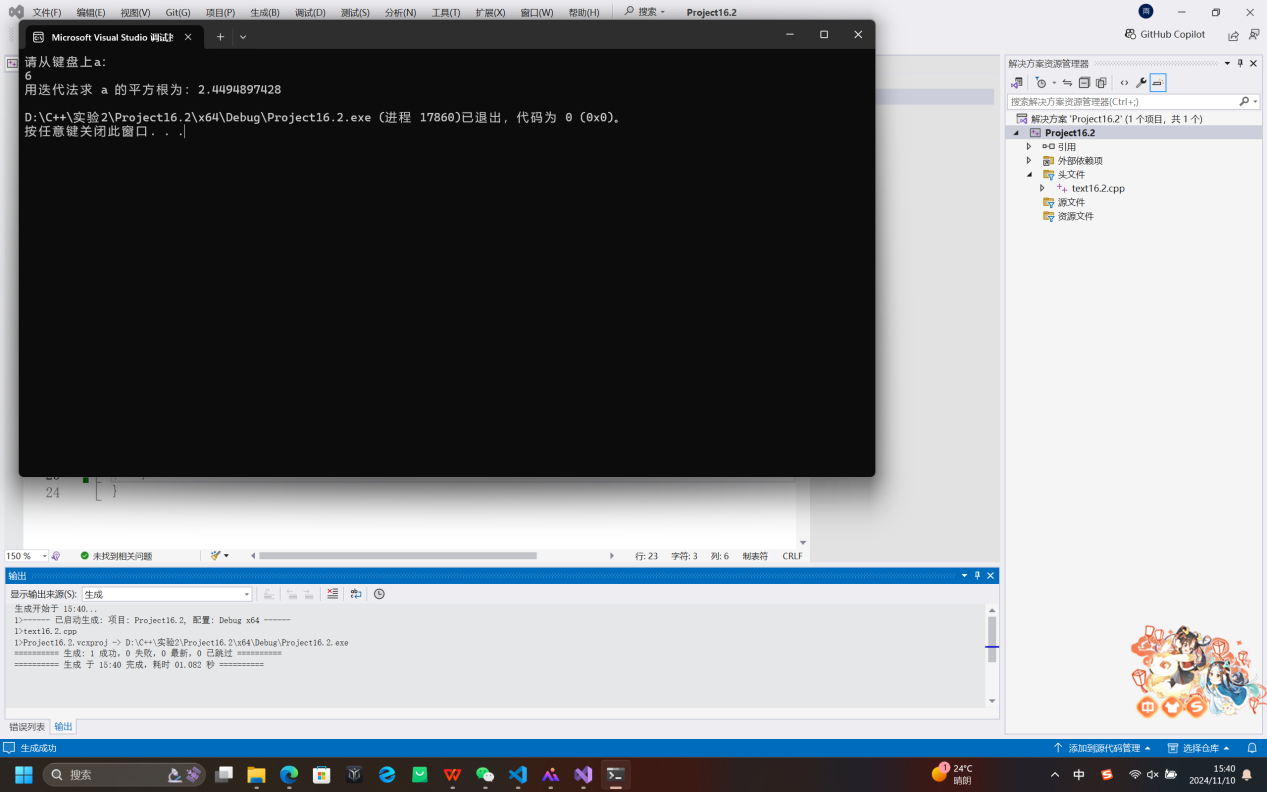
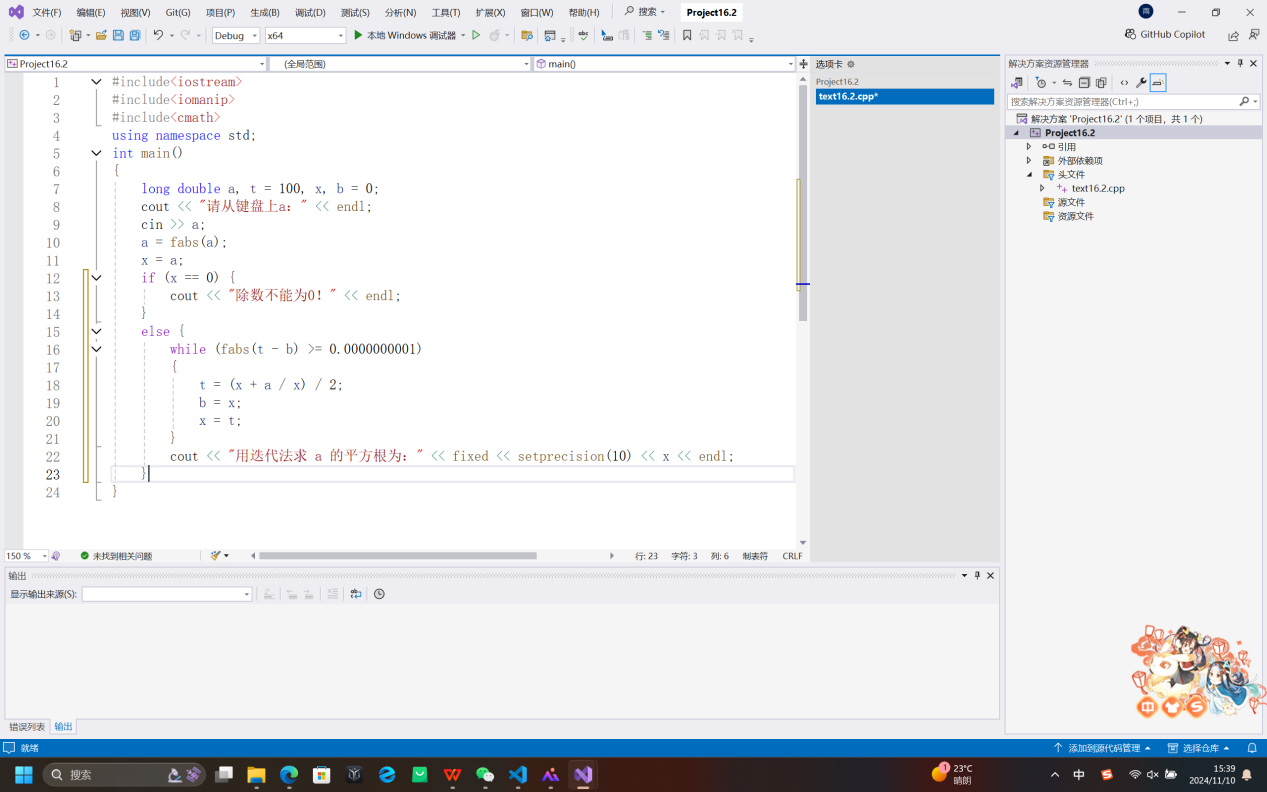
5.

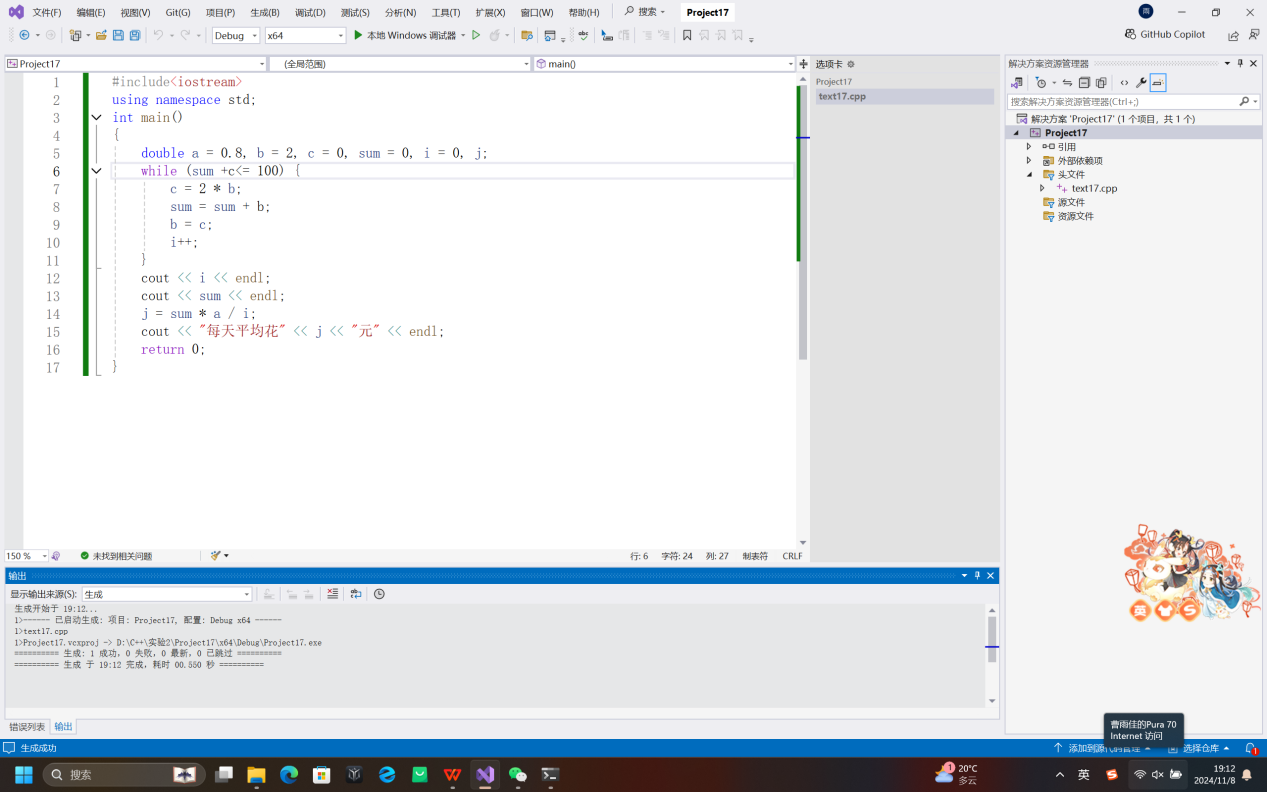
6.

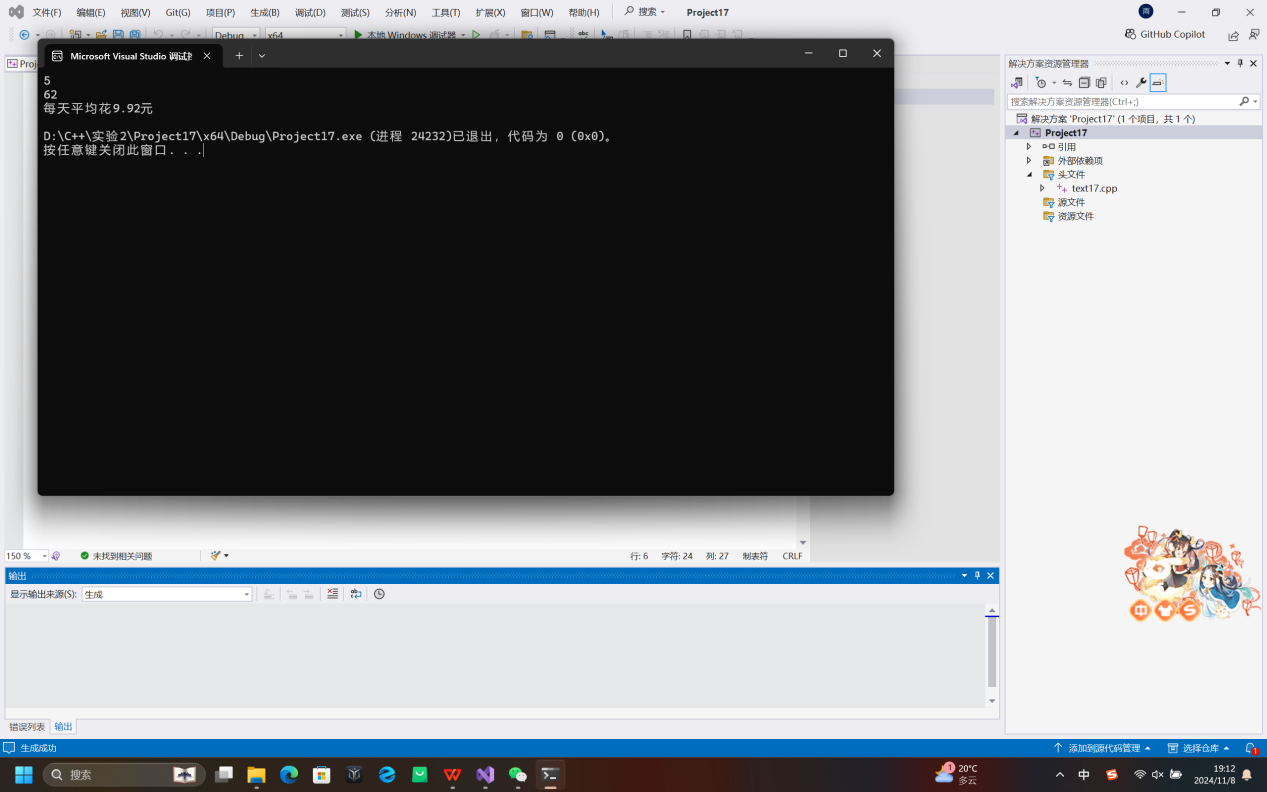
7.

8.





9.

****

**四、遇到的问题与解决方法**

1. 字符输入问题：在统计字符个数的程序中，发现 cin 和 scanf() 不能读入空格以及 \n 字符。通过查阅资料，了解到可以使用 getchar() 函数或者 cin.get() 函数来读取单个字符，包括空格和换行符。在循环读取字符时，通过合适的条件判断来控制循环的结束，比如当读取到 \n 时停止循环，从而解决了字符输入不完整的问题。

2. 迭代法精度控制问题：在使用迭代法求平方根时，对于精度 |xn+1 - xn|<10 -5 的控制，一开始不太清楚如何准确判断是否达到精度要求。通过在循环中添加判断条件，比较相邻两次迭代结果的差值与设定精度的大小关系，来决定是否继续迭代。同时，尝试修改精度为 10 -10 时，发现迭代次数大幅增加且可能由于浮点数精度限制出现死循环或不准确的结果。经过分析，明白了浮点数在计算机中的表示精度有限，不能无限制地提高迭代精度要求，需要根据实际情况合理设置精度值，以平衡计算结果的准确性和计算效率。

3. 除数为 0 及非法运算符处理：在计算器程序中，对于除数为 0 和非法运算符的情况考虑不够周全。通过添加条件判断语句，在进行除法运算前检查除数是否为 0，如果为 0 则输出错误提示信息。对于非法运算符，使用if判断语句或 switch 语句中的 default 分支来捕获并提示用户输入了非法运算符。

**五、体会**

本次数据结构实验让我在编程思维和逻辑构建方面有了很大提升。逻辑运算与表达式的运用使程序能够根据不同条件做出正确判断。例如在字符大小写转换和判断三角形类型的程序中，通过巧妙运用逻辑表达式，能够准确地执行相应操作。

掌握各种控制语句（if、switch、while、do - while、for）的语法结构和执行过程后，我在程序设计时能够更加灵活地选择合适的语句来实现功能。如在计算器程序中，根据不同的运算符选择相应的计算分支，这需要对 if 和 switch 语句的熟练运用；而循环语句在处理重复性操作时发挥了巨大作用，像计算数列求和、输出图形等任务，通过循环能够简洁高效地完成。

在解决问题过程中，遇到了一些如字符输入、特殊情况处理等挑战。通过自主查阅资料和探索，不仅找到了解决方法，还拓宽了知识面。例如在处理输入空格和换行字符时，学习到了新的函数和方法来实现。

最大公约数和最小公倍数的计算让我对数学算法在编程中的应用有了更深入的理解。同时，迭代法求平方根的实验让我体会到了近似算法的巧妙之处以及精度控制的重要性。

总体而言，实验二让我在编程实践中不断成长，提高了自己的编程技能和解决实际问题的能力，也让我更加深刻地认识到数据结构和算法在程序设计中的核心地位。