**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：8209240203

姓 名：丁晨阳

**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 k = 1;

Int i = k + 1;

cout << ~~i++~~ ++i<< endl;

~~int~~ i = 1;

cout << ~~i++~~++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；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.

#include<iostream>

using namespace std;

int main()

{

//int i = k + 1; 没有初始化变量k 无法赋值

int k = 1;

int i = k + 1;

//cout << i++ << endl; i++是先输出i的原始值 再对i进行+1的操作 在此步中 i++没有意义

cout << ++i << endl;

//int i = 1; i重定义 报错

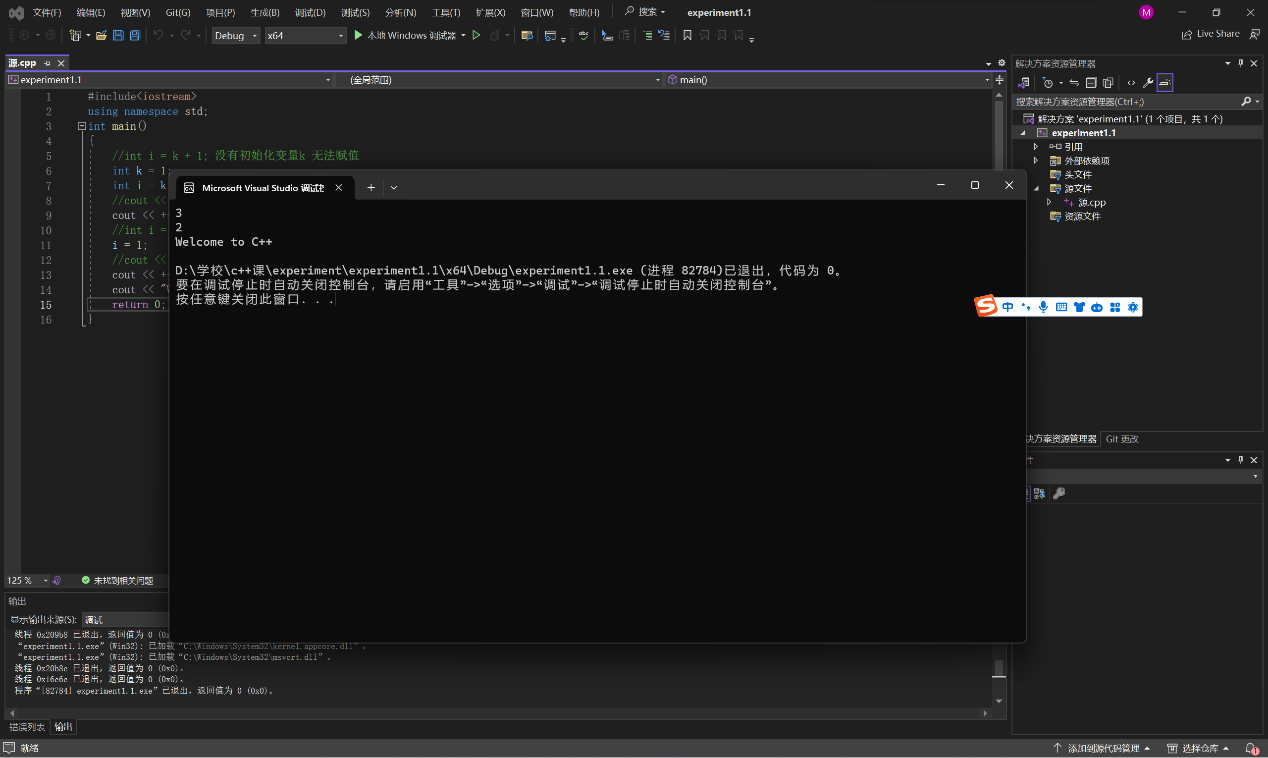
i = 1;

//cout << i++ << endl; i++是先输出i的原始值 再对i进行+1的操作 在此步中 i++没有意义

cout << ++i << endl;

cout << “Welcome to C++” << endl;

return 0;

}

2.

#include<iostream>

#define π 3.14;//定义宏常量 防止值被改

using namespace std;

int main()

{

double r;//定义半径

double h;//定义高度

cout << “请输入圆锥底面半径:”;//提示输入半径

cin >> r;//输入半径

cout << “请输入圆锥高:”;//提示输入高

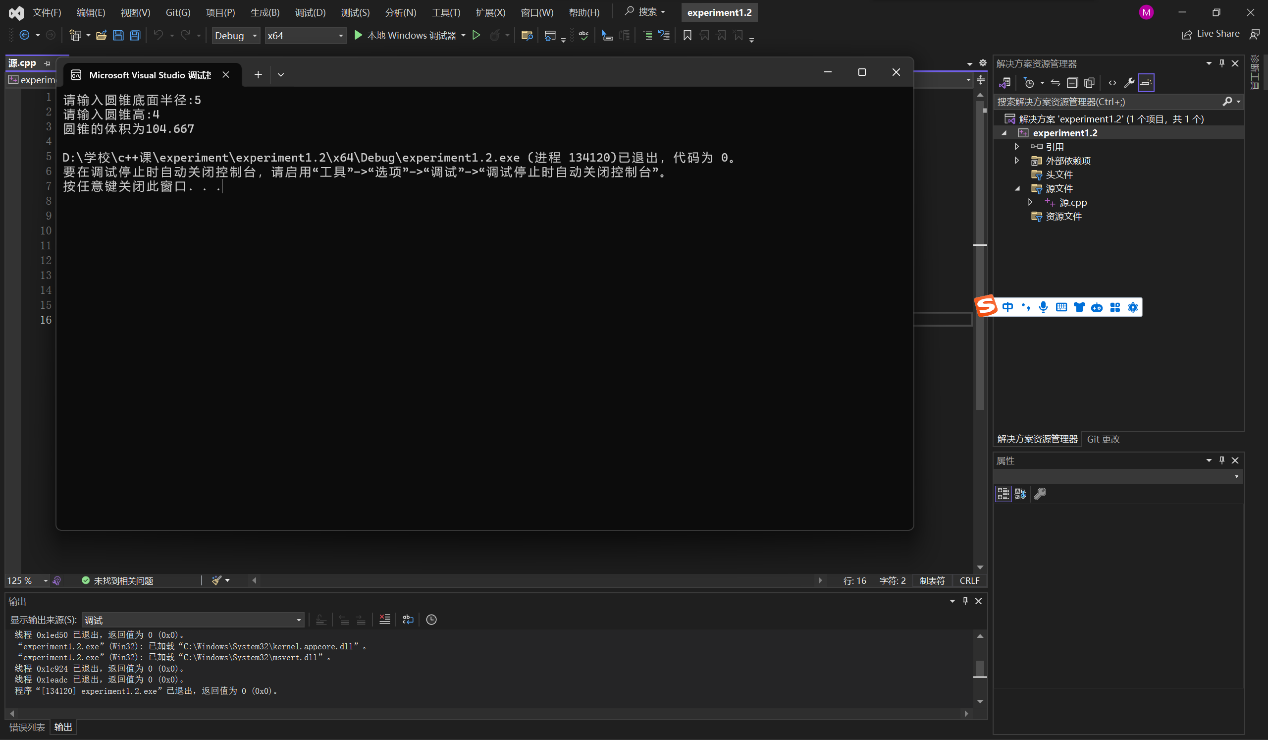
cin >> h;//输入高

double s = (r \* r) \* π;//s=πr^2 定义s并给其赋值

double v = s \* h \* (1.0 / 3);//v=1/3sh 定义v并给其赋值

cout << “圆锥的体积为” << v << endl;

return 0;

}

3.

#include<iostream>

using namespace std;

int main()

{

cout << “char length:” << sizeof(char) << endl;//输出char类型数据的长度

cout << “int length:” << sizeof(int) << endl;//输出int类型数据的长度

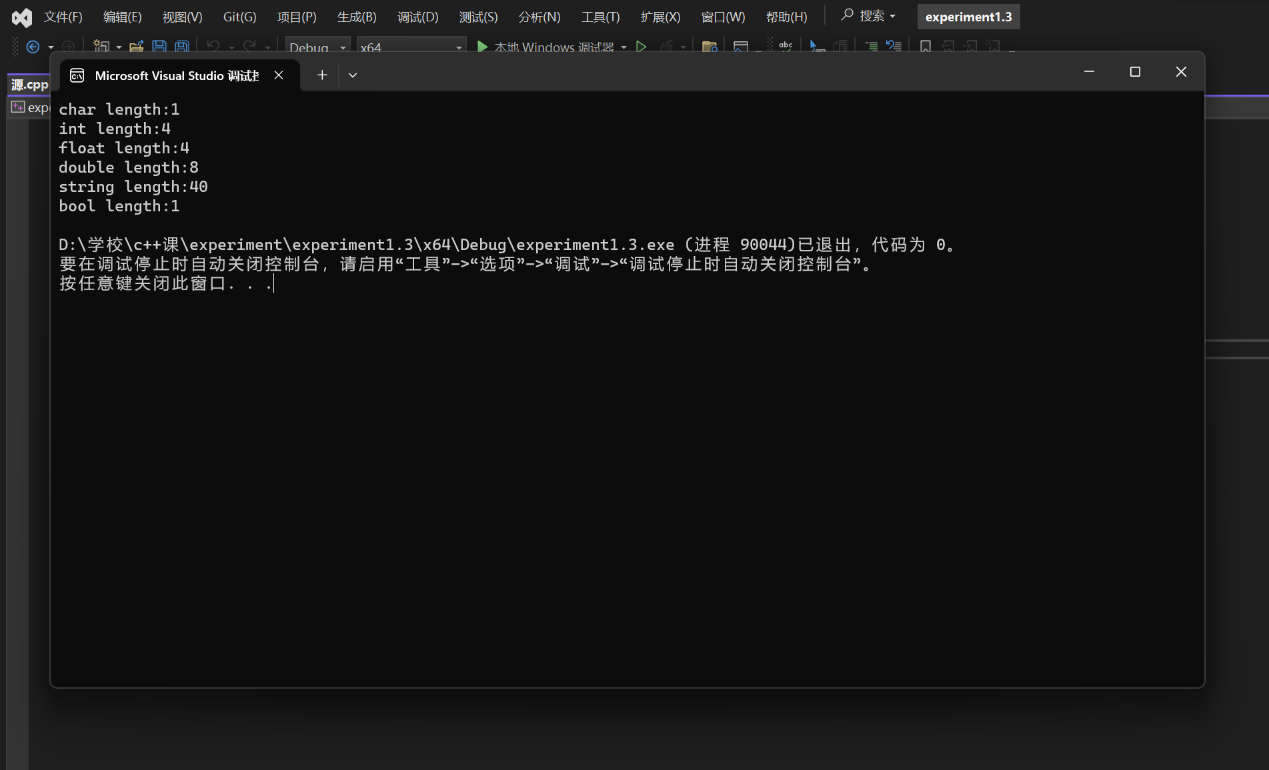
cout << “float length:” << sizeof(float) << endl;//输出float类型数据的长度

cout << “double length:” << sizeof(double) << endl;//输出double类型数据的长度

cout << “string length:” << sizeof(string) << endl;//输出string类型数据的长度

cout << “bool length:” << sizeof(bool) << endl;//输出bool类型数据的长度

return 0;

}

4.

#include <iostream>

#include <iomanip>

using namespace std;

int main()

{

unsigned int testUnint = 65534;//oxfffe

cout << “output in unsigned int 1 type:” << testUnint << endl;//<<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

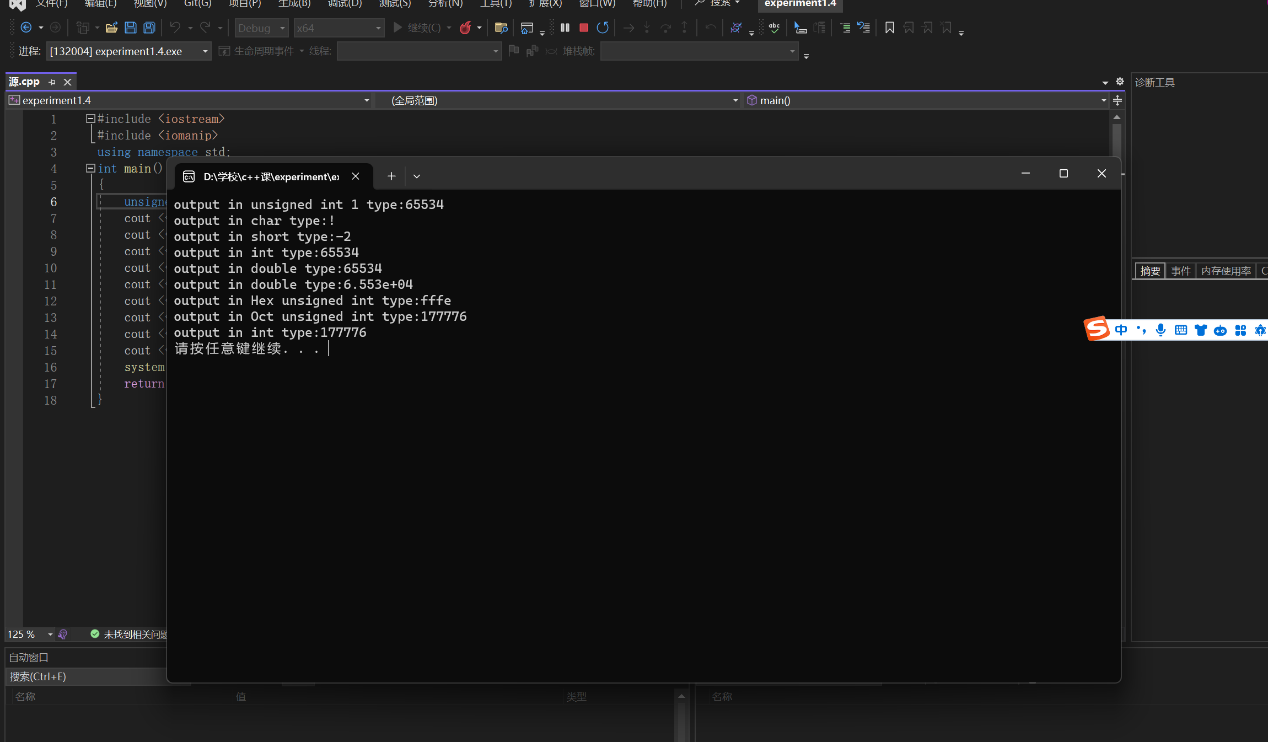
cout << “output in Oct unsigned int type:” << oct << testUnint << endl;//八进制输出

cout << “output in int type:” << static\_cast<int>(testUnint) << endl;//转换成int

system(“pause”);

return 0;

}



5.

#include<iostream>

#include<iomanip>//用setprecision要引用该头文件

using namespace std;

int main()

{

double F;//定于华氏温度变量

cout << “请输入华氏温度:”;//提示输入华氏温度

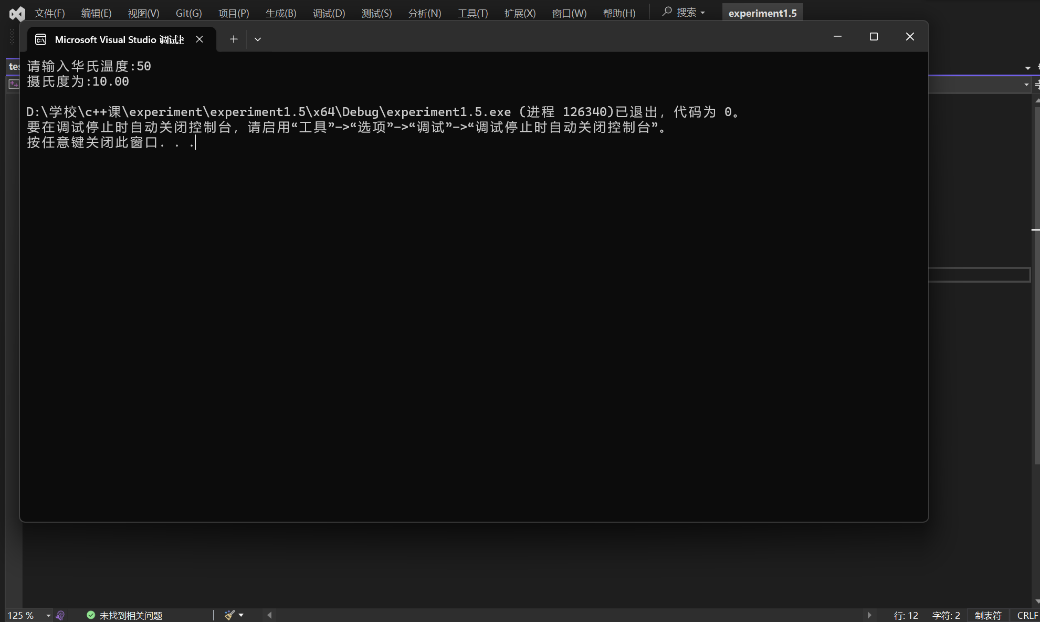
cin >> F;//输入华氏温度

double C = (F – 32) / 1.8;//定义摄氏温度 计算并赋值

cout << “摄氏度为:” <<fixed << setprecision(2) << C << endl;//输出摄氏度 使用fixed和setprecision保留两位小数

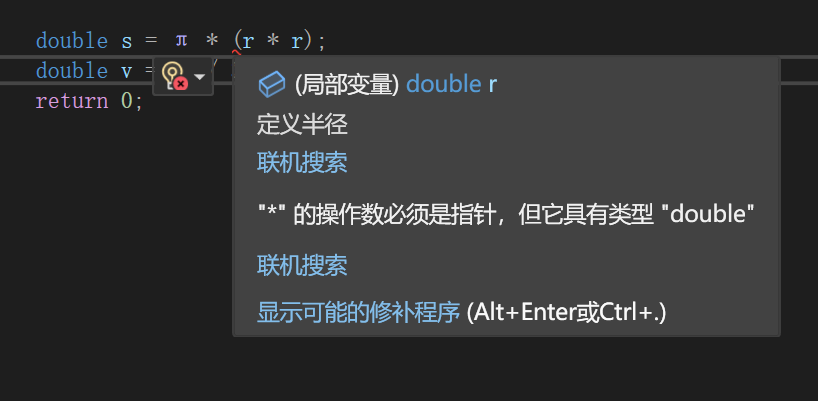
return 0;

}

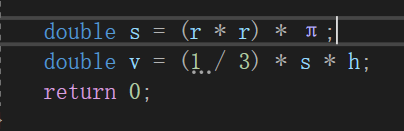


**四、遇到的问题与解决方法**

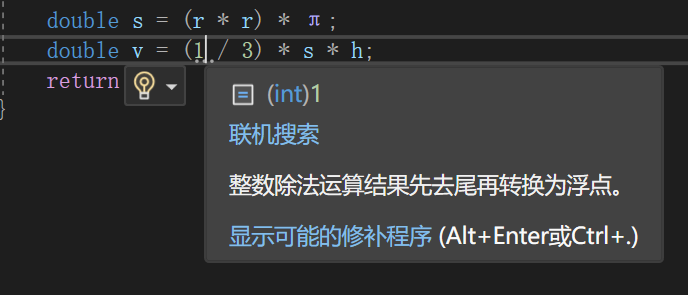
**1.在计算圆锥体积时，遇到报错及提示**

 （1）报错

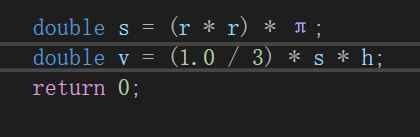
根据编译器的报错提示，推测是由于在r变量前加了\*号，使编译器识别成了指针变量。

 考虑到r是局部变量，将（r\*r）提前写，后面写“\*π”再次尝试。

报错消除，程序能够正常运行

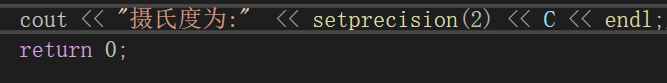
（2）提示

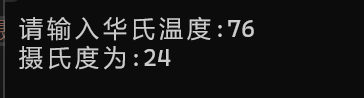
在计算1/3分数时获得编译器提示，想起整数除法运算会去尾，影响计算准确度。

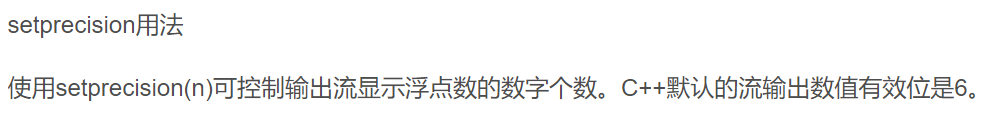
 考虑将整数转换为小数，改为1.0/3，提示消除，程序能够正常运行，计算准确度提高。

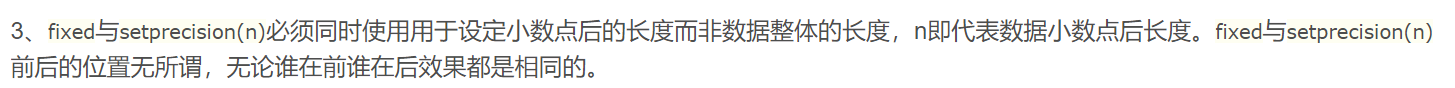
**2.不知道如何保留两位小数**

在编辑华氏度转摄氏度的程序时，体重要求结果保留两位小数。

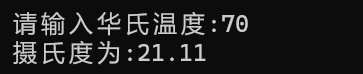
 首先考虑通过setprecision设置显示浮点型数据的位数。

 在设置显示位数后，运行程序，发现运行结果不佳。

 随后查阅相关资料，想起单使用setprecision控制的是总的显示位数，不控制小数点后的显示位数

 再次查询相关资料，发现要想保留小数点两位，需要fixed与setprecision结合使用。

 遂在代码中加入fixed。

 发现代码可以正常运行，两位小数可以正常保留，代码修改成功。

**五、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。

例如，在实验1.1当中，出现了两次对于变量i的定义。我在最开始还没有将这段代码录进编译器时，没有注意到这一点错误。直到将此段代码在编译器中打出，发现编译器在此处提示报错，在查看编译器的报错提示后，才发现在这段代码中存在着对于变量i的重复定义。可能在日常的编程中，自己比较少遇到这一类错误，所以无法立即发现题中的这一点问题。在看到编译器的提示后才对原代码中的这一个问题恍然大悟。希望这一次经历能够加深我对这一类编程语法的重视性。

**2.在学习了后续的内容后，容易遗忘前面学到的知识，需要多加复习**

实验一涉及到的c++编程知识都是很基础的内容。在进行编程的实际过程中，我发现，对于前期学到的、一些比较少用的知识点会出现遗忘的情况。需要借助相关资料，才能回想起某一个知识点的所有细节内容。

例如，在试图控制浮点型数据显示位数的时候，回想起有一种方式可以调整位数，也会记得使用该方法需要引入头文件。但是无法仔细想起如何拼写该函数，也无法想起要引入哪一个头文件。只能通过通过查找相关资料的方法，回想起此函数的使用方式。

再例如，在进行进制转换时，也对进制转换的方式有所遗忘。在老师讲述这方面内容时，对这块知识掌握程度比较高，但是时隔长时间后，已经全然忘记如何进行进制转换进制转换，需要参考原代码中给出的方式才能成功实现进制转换。

实验一的实验帮助我对一些知识点重新进行了复习，也让我认识到了复习旧知识的重要性。

**3.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。

在输入公式s=π\*(r\*r)时，发现编译器将\*（r\*r）理解为了指针。虽然我可以立即将次报错取消，但是在取消报错之后我也无法理解为何编译器会在此处报错。可能是因为当初在进行指针相关知识的学习过程中，我对知识的理解程度不是很到位，例如，我对于“int \*p=1；”定义后，有时会难以分清“\*p”以及“p”的区别，有时又可以分清。

所以在这一实验过程中，我明白了自学过程中质量的重要性。

**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。 可以

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.

#include<iostream>

#include<algorithm>

using namespace std;

int main()

{

    char ch;

    cin>>ch;

    if(ch>='a'&& ch<='z')

    cout<<(char)toupper(ch)<<endl;

    else

    cout<<(int)ch+1<<endl;

    system("pause");

    return 0;

}



2.

#include<iostream>

#include<cmath>

using namespace std;

int main()

{

    double x;

    cin>>x;

    if(x>0 && x<1)

    {

        cout<<3-2\*x<<endl;

    }

    else if(x>=1 && x<=5)

    {

        cout<<2/(4\*x)+1<<endl;

    }

    else if(x>=5 && x<10)

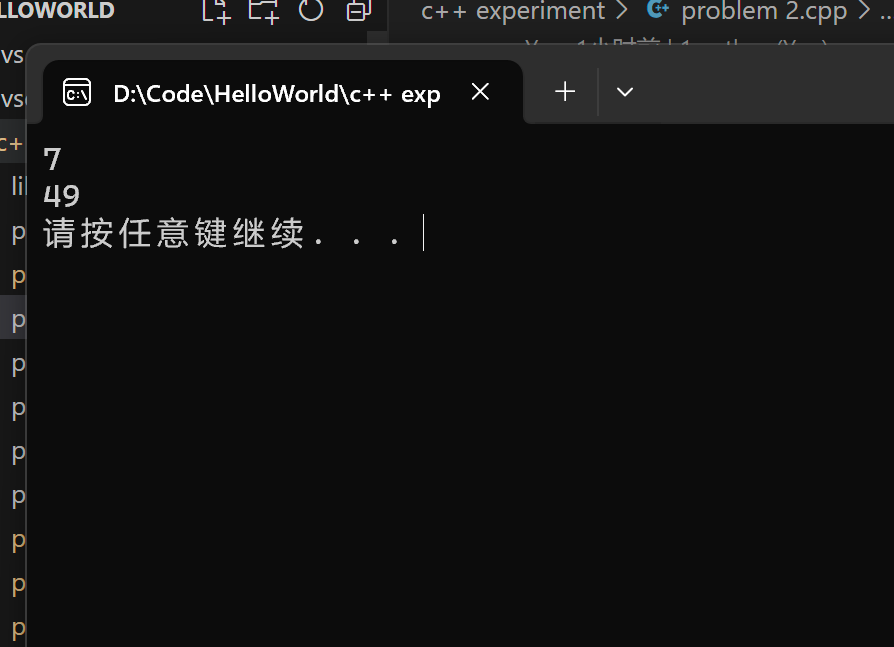
    {

        cout<< pow(x,2)<<endl;

    }

    system("pause");

    return 0;

} 

3.

#include<iostream>

using namespace std;

int main()

{

    int s1,s2,s3;

    cin>>s1>>s2>>s3;

    if (s1 + s2 > s3 && s1 + s3 > s2 && s2 + s3 > s1)

    {

        if (s1 == s2 || s2 == s3 || s1 == s3)

        {

            cout << "Success!" << endl;

        }

        cout << (s1 + s2 + s3) << endl;

    }

    else

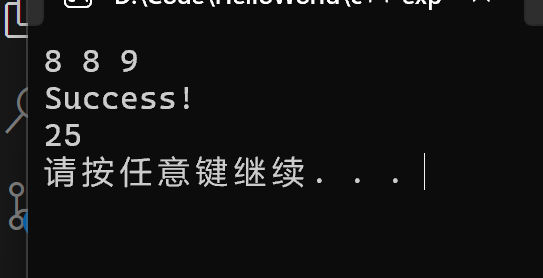
    {

        cout << "failed" << endl;

    }

    return 0;

}



4.

#include<iostream>

#include<string>

using namespace std;

int main()

{

    string s;

    cin>>s;

    switch (s[1])

    {

        case '+':

        {

            cout<<(s[0]-'0')+(s[2]-'0')<<endl;

            break;

        }

        case '-':

        {

            cout<<(s[0]-'0')-(s[2]-'0')<<endl;

            break;

        }

        case '\*':

        {

            cout<<(s[0]-'0')\*(s[2]-'0')<<endl;

            break;

        }

        case '/':

        {

            if(s[2]=='0')

            {

                cout<<"Error: Division by zero!"<<endl;

                system("pause");

                return 0;

            }

cout<<(s[0]-'0')/(s[2]-'0')<<endl;

            break;

        }

        case '%':

        {

      if(s[2]=='0')

            {

                cout<<"Error: Division by zero!"<<endl;

                system("pause");

                return 0;

            }

cout<<(s[0]-'0')%(s[2]-'0')<<endl;

            break;

        }

        default:

        {

            cout<<"Invalid operation!"<<endl;

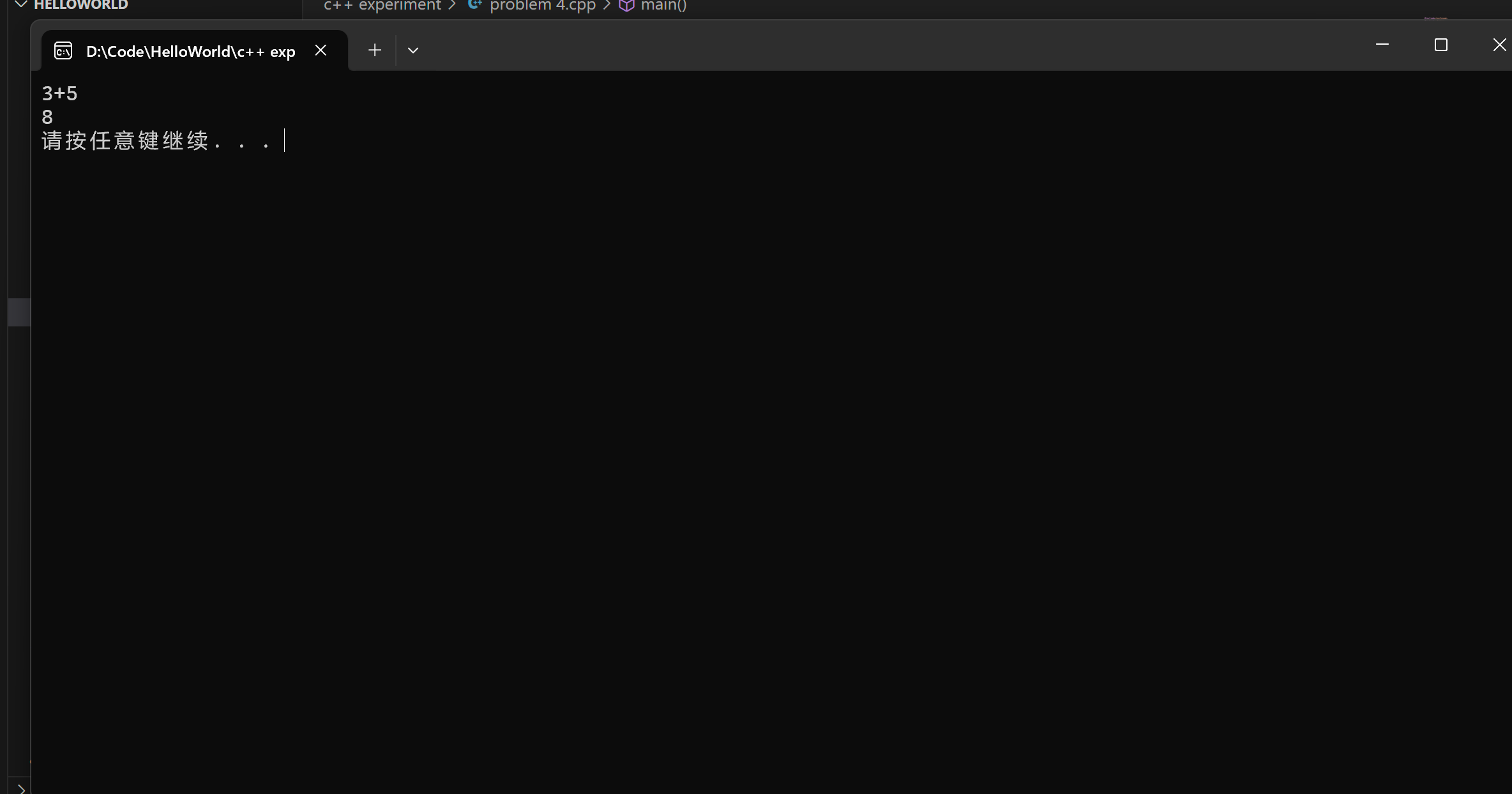
            break;

        }

    }

    system("pause");

    return 0;

} 

5.

#include <iostream>

#include<string>

#include<ctype.h>

using namespace std;

int cnt[4];

int main()

{

    string s;

    getline(cin,s);

    for(char ch:s)

    {

        if(isalpha(ch))

        cnt[0]++;

        else if(isdigit(ch))

        cnt[1]++;

        else if(ch==' ')

        cnt[2]++;

        else

        cnt[3]++;

    }

    cout<<"Alphabetic characters: "<<cnt[0]<<endl;

    cout<<"Numeric characters: "<<cnt[1]<<endl;

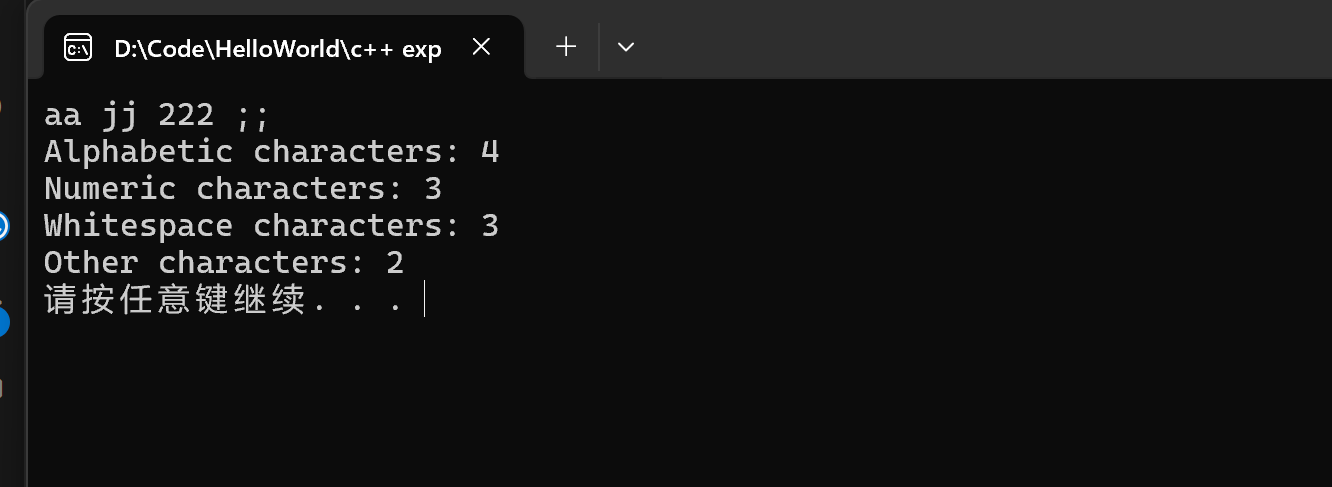
    cout<<"Whitespace characters: "<<cnt[2]<<endl;

    cout<<"Other characters: "<<cnt[3]<<endl;

    system("pause");

    return 0;

}



6.

#include<iostream>

using namespace std;

int gcd(int a, int b)

{

    int temp = b;

    while (a % b)

    {

        temp = a % b;

        a = b;

        b = temp;

    }

    return temp;

}

int main()

{

    int a,b;

    cin>>a>>b;

    //最大公约数

    cout<<gcd(a,b)<<endl;

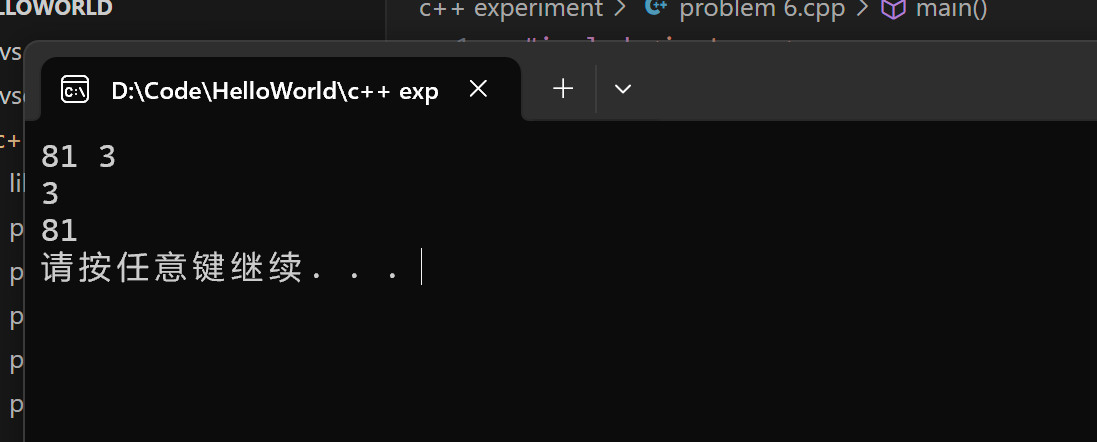
    //最小公倍数

    cout<<(a\*b)/gcd(a,b)<<endl;

    system("pause");

    return 0;

}



7.

#include<iostream>

using namespace std;

int main()

{

    for(int i=0;i<5;i++)

    {

        for(int j=0;j<i+1;j++)

        {

            cout<<"\*";

        }

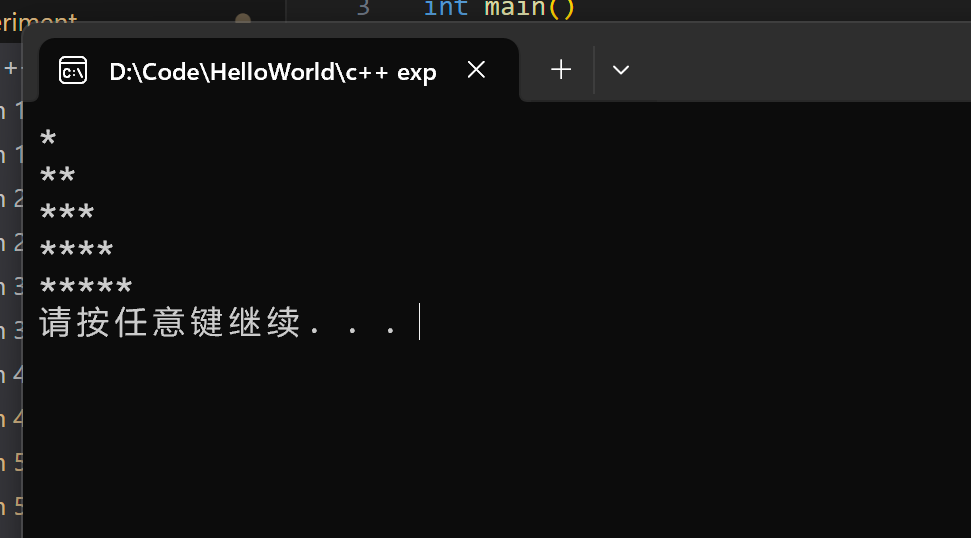
        cout<<endl;

    }

    system("pause");

    return 0;

}



8.

#include<iostream>

#include<cmath>

using namespace std;

int main()

{

    double a;

    cin>>a;

    if(a<0)//如果是负数直接报错并退出

    {

        cout<<"Error: Negative number input."<<endl;

        system("pause");

        return 0;

    }

    double x1=a;

    while(1)

    {

        double x2=(x1+(a/x1))\*(1.0/2);

        if(fabs(x2-x1)<0.00001)

        {

            cout<<x2<<endl;

            break;

        }

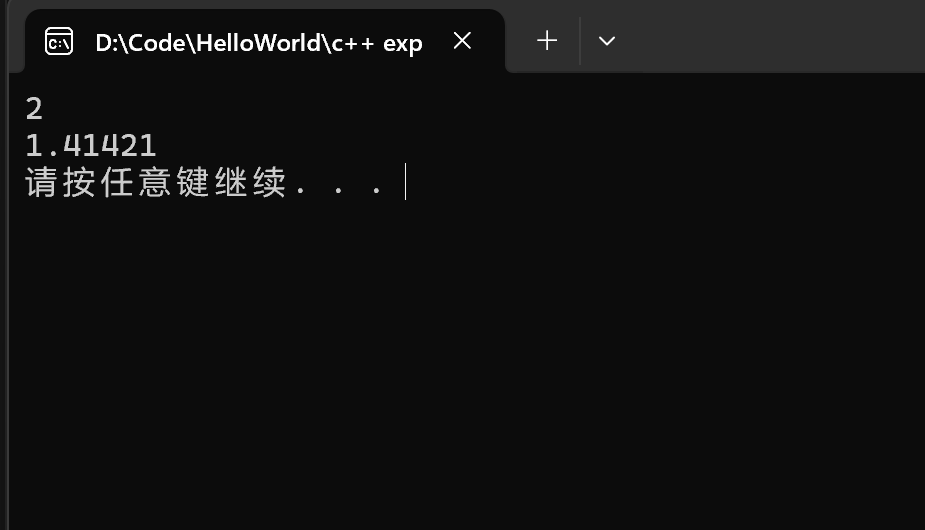
        x1=x2;

    }

    system("pause");

    return 0;

}



对于问题二，将if内代码段改为：

 if(fabs(x2-x1)<0.0000000001)

        {

            cout<<fixed<<setprecision(10)<<x2<<endl;

            break;

        }

即可实现

9.

#include<iostream>

using namespace std;

int main()

{

    int total=2;

    int buy=2;

    int day=1;

    while(total<=100)

    {

        buy\*=2;

        total+=buy;

        day++;

}

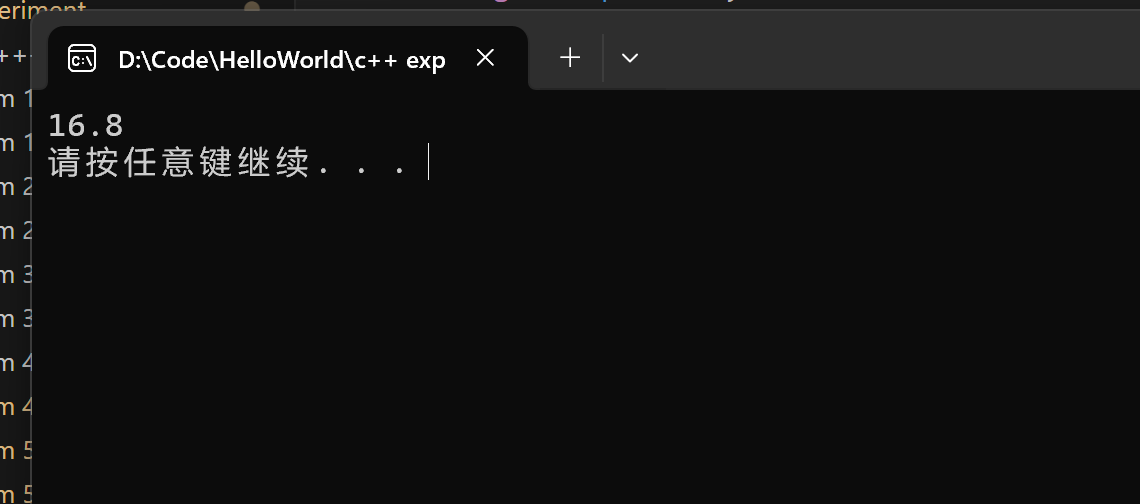
double ans=(total\*0.8)/(day\*1.0);

    cout<<ans<<endl;

    system("pause");

    return 0;

}



**四、遇到的问题与解决方法**

**1.不了解字符串中如何录入空格**

查找资料发现使用getline函数即可解决。

**五、体会**

**1.可以再对录入含空格的字符串进行更深的了解。**

对于getline函数可以进行进一步的了解

**2.在编写程序时要记得关注细节。**

在本次实验的过程中，我发现自己会对一些细节疏忽，从而造成一些小错误。例如在计算绝对值时忘记加上绝对值相关的判断逻辑或者忘记进行绝对值得运算，又或者说在计算类似a%i的循环时，将i的初始值设计为0导致程序报错。

其实这些细节的错误是可以避免的。在最后程序的运行阶段，如果发现程序存在细节问题，且编译器不报错，排查这些细节的过程将会变得十分麻烦且浪费时间。因此细节的关注十分有必要。