**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：8209240201

姓 名：徐昱隆

**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. #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;

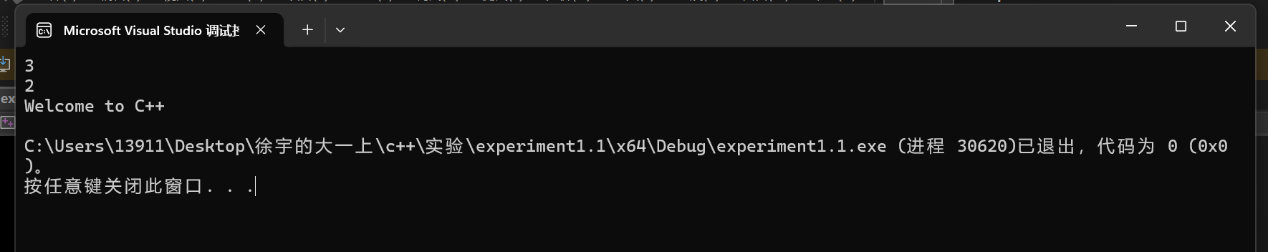
//int i = 1; i重定义 报错

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

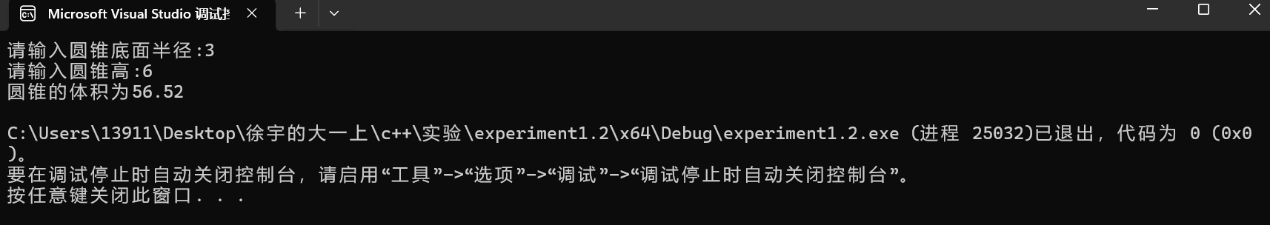
cin >> h;//输入高

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

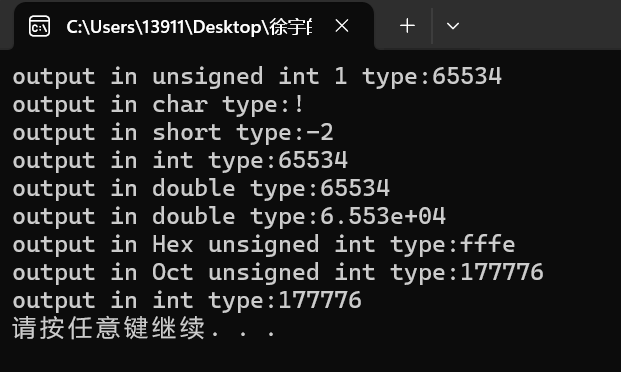
cout << "output in Oct unsigned int type:" << oct << testUnint << endl;//八进制输出

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

system("pause");

return 0;

}



5. #include <iostream>

using namespace std;

int main() {

double fahrenheit; // 定义双精度浮点数来存储输入的华氏温度

cout << "请输入华氏温度：";

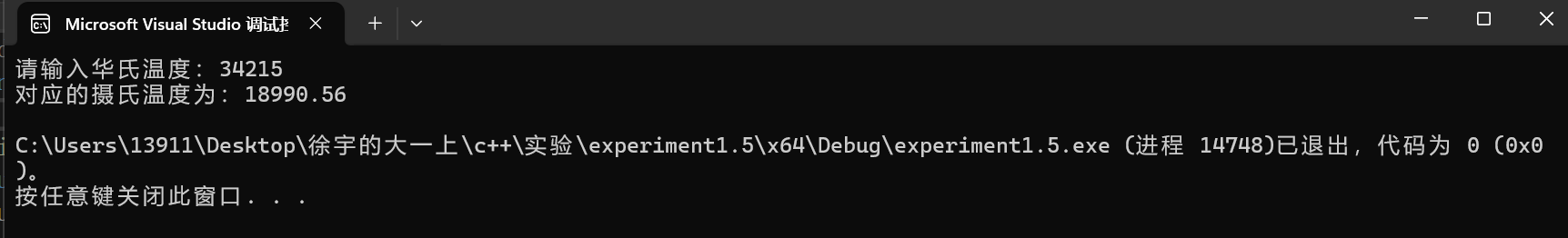
cin >> fahrenheit; // 输入华氏温度

double celsius = (fahrenheit - 32) \* 5.0 / 9.0; // 进行温度转换的计算

cout.precision(2); // 设置输出精度为 2 位小数

cout << "对应的摄氏温度为：" << std::fixed << celsius << std::endl; // 输出结果

return 0;

}

**四、遇到的问题与解决方法**

**1.不知道如何保留两位有效数字**

在编辑华氏度转摄氏度的程序时，要求结果保留两位小数。

在查阅相关资料后，学会运用将fixed与fahrenheit结合进行使用，从而将问题解决



**五、体会**

**通过C++上机实验，我收获颇丰。在实践中，我深刻体会到理论知识与实际操作的差异。敲代码的过程中，每一个标点符号、每一个变量定义都至关重要，一个小失误就可能导致程序无法运行，这让我明白严谨的重要性。**

**在解决编译错误时，从最初的迷茫到逐渐能分析出问题所在，这个过程极大地锻炼了我的问题解决能力。我学会了利用调试工具，逐步追踪程序的执行过程，去理解代码的逻辑和数据的流向。**

**同时，随着实验的深入，我对C++的语法和各种功能有了更清晰的认识。像是函数的调用、类的创建等，不再是纸上谈兵，而是真正理解了它们在程序中的作用。这次上机实验为我今后进一步学习和使用C++打下了坚实的基础。**

**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.** #include<iostream>

using namespace std;

int main()

{

char ch;//定义ch

cout << "请输入一个字符" << endl;//提示输入

cin >> ch;//输入

int a = int(ch);//定义a 并将ASCII码赋值给a

if (a >= 97 && a <= 122)//判断 如果ASCII码在小写字母之间

{

a -= 32;//ASCII码-32 变为大写

cout << char(a);//输出大写后的字符

}

else//没有大写 或者已经为大写

{

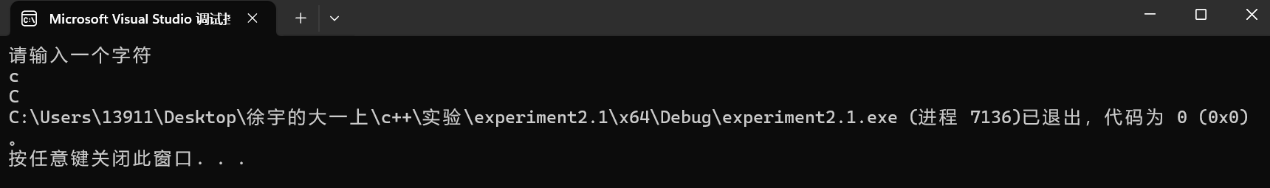
a++;//ASCII码+1

cout << a;//输出+1后的值

}

return 0;

}

****

**2.** #include<iostream>

using namespace std;

int main()

{

double x;//定义自变量x

double y;//定义因变量y

cout << "请输入x的值:";//提示输入x的值

cin >> x;//输入x的值

if (x > 0 && x < 1)//如果0<x<1

{

y = 3 - (2 \* x);//计算出y的值

cout << "y=" << y << endl;//输出y的值

}

else if (x >= 1 && x < 5)//如果1<=x<5

{

y = 2 / (4 \* x) + 1;//计算出y的值

cout << "y=" << y << endl;//输出y的值

}

else if (x >= 5 && x < 10)//如果5=<x<10

{

y = x \* x;//计算出y的值

cout << "y=" << y << endl;//输出y的值

}

else//都不满足

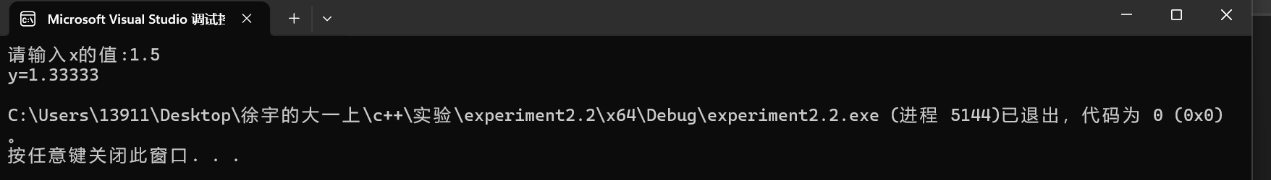
{

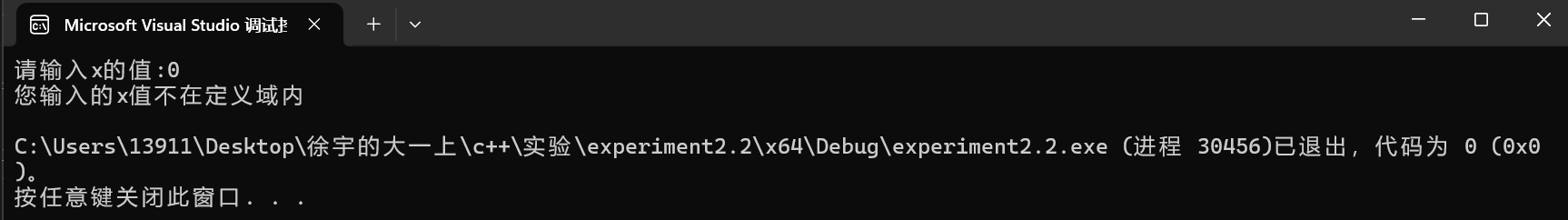
cout << "您输入的x值不在定义域内" << endl;//不在定义域里

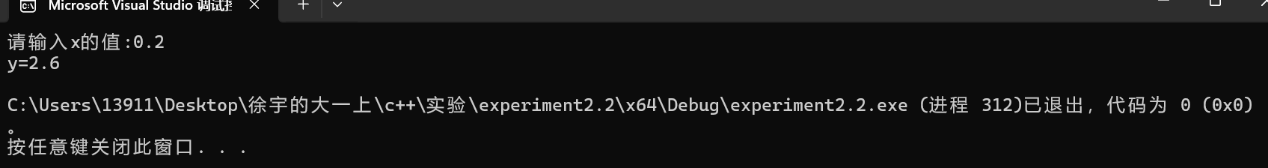
}

return 0;

}

****

****

****

**3.** #include<iostream>

using namespace std;

int main()

{

int s1;//定义边长1

int s2;//定义边长2

int s3;//定义边长3

cout << "请分别输入三边边长" << endl;//提示输入三条边边长

cout << "边长1:";//提示输入边长1

cin >> s1;//输入边长1

cout << "边长2:";//提示输入边长2

cin >> s2;//输入边长2

cout << "边长3:";//提示输入边长3

cin >> s3;//输入边长三

if (s1 + s2 > s3 && s1 + s3 > s2 && s2 + s3 > s1)//判断是否满足三角形构成条件(两边之和大于第三边)

{

//满足三角形构成条件

if (s1 == s2 || s2 == s3 || s1 == s3)//判断是否存在两边边长相等的情况

{

cout << "您创建的是等腰三角形" << endl;//输出等腰三角形结果

}

cout << "三角形周长为" << (s1 + s2 + s3) << endl;//输出三角形周长

}

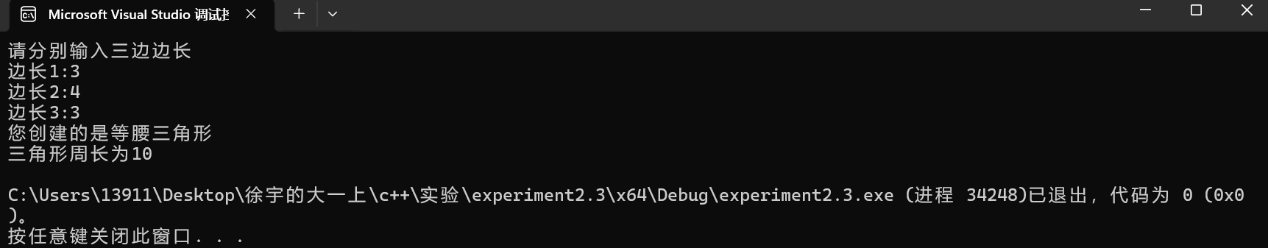
else//不满足三角形构成条件

{

cout << "您输入的三边长不能组成三角形" << endl;//输出提示三角形不能构成

}

return 0;

}****

**4.** #include<iostream>

using namespace std;

int main()

{

cout << "欢迎使用本计算器系统!" << endl;//提示进入计算器程序

cout << "请输入第一个值:" << endl;//提示输入值1

double num1;//定义值1

cin >> num1;//输入值1

cout << "请选择你要进行的运算" << endl;//提示选择进行的运算(避免了运算符判与选择)

cout << "1.加法运算" << endl;

cout << "2.减法运算" << endl;

cout << "3.乘法运算" << endl;

cout << "4.除法运算" << endl;

cout << "5.取余运算" << endl;

int select;//定义选择值

cin >> select;//输入选择值

cout << "请输入第二个值:" << endl;//提示输入值2

double num2;//定义值2

cin >> num2;//输入值2

switch (select)//判断选择值

{

case 1://1 执行加法运算

cout << "两数之和为" << (num1 + num2) << endl;//输出加法结果

break;

case 2://2 执行减法运算

cout << "两数之差为" << (num1 - num2) << endl;//输出减法结果

break;

case 3://3 执行乘法运算

cout << "两数之积为" << (num1 \* num2) << endl;//输出乘法结果

break;

case 4://4 执行除法运算

if (num2 == 0)//判断除数是否为0

{

//如果除数为0

cout << "除数不能为0" << endl;//提示除数不能为0

break;

}

else//除数不为0

{

cout << "两数之商为" << (num1 / num2) << endl;//输出除法结果

break;

}

case 5://5 执行取余运算

if (num2 == 0)//判断取余数是否为0

{

//如果取余数为0

cout << "除数不能为0" << endl;//提示取余数不能为0

break;

}

else//取余数不为0

{

cout << "两数取余结果为" << (int(num1) % int(num2)) << endl;//输出取余结果

break;

}

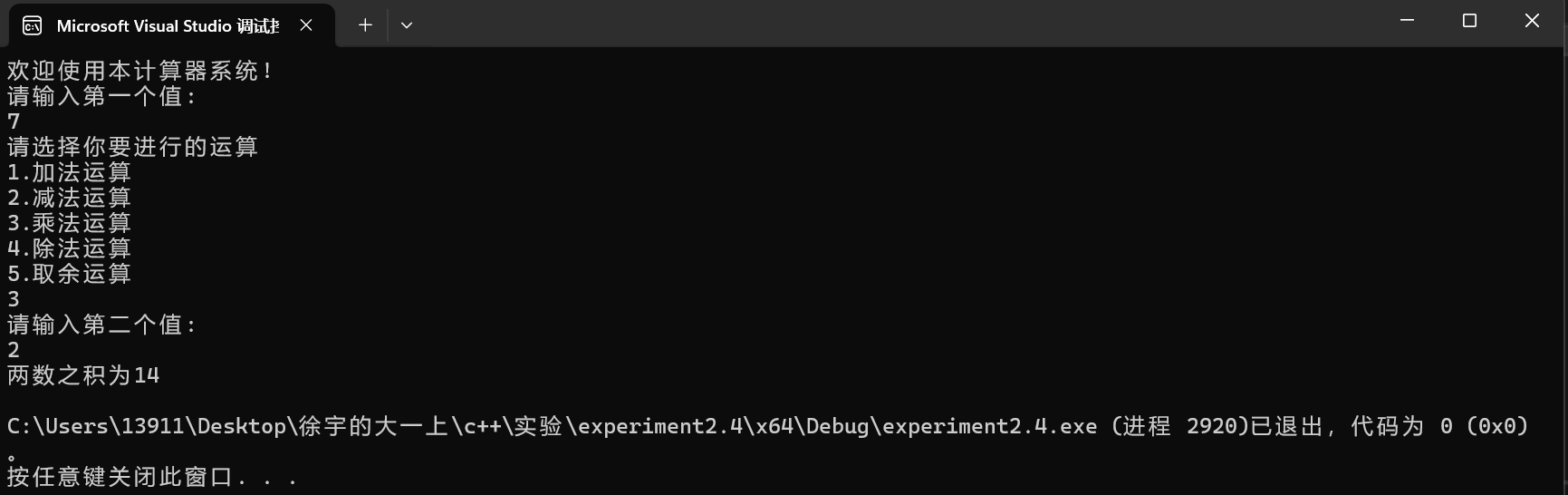
default:

cout << "请选择合法运算" << endl;//选择的运算不合法 提示选择合法运算

}

return 0;

}

****

**5.** #include<iostream>

using namespace std;

int main()

{

char c;//定义一个字符

int letter = 0, space = 0, number = 0, other = 0;//定义各种字符类型

cout << "请输入一串字符串" << endl;//提示输入字符串

while ((c=cin.get()) != '\n')//如果cin中不包含换行 循环继续

{

if (c>='a'&&c<='z' || c>='A'&&c<='Z')//判断字母

letter++;//字母字符+1

else if (c==' ')//判断空格

space++;//空格字符+1

else if (c>='0'&&c<='9')//判断数字

number++;//数字字符+1

else

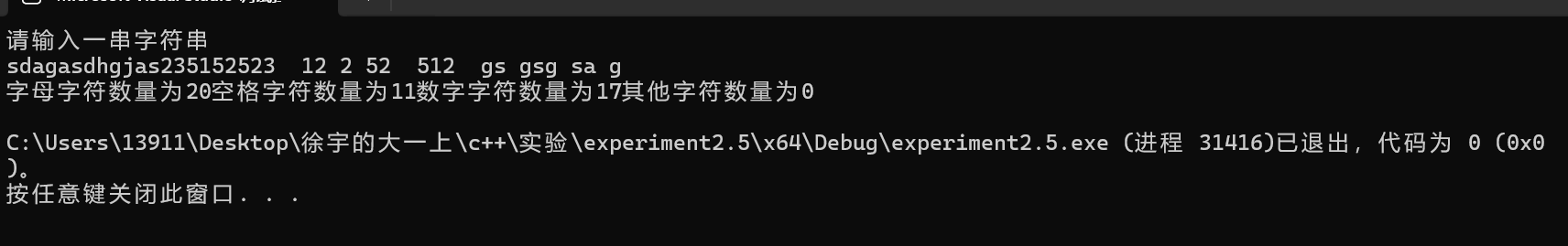
other++;//否则其他字符+1

}

cout << "字母字符数量为" << letter << "空格字符数量为" << space << "数字字符数量为" << number << "其他字符数量为" << other << endl;//输出各类字符数量

return 0;

}

****

**6、**#include<iostream>

using namespace std;

int Gcd(int num1, int num2)//定义最大公因数函数

{

int temp;//定义临时变量

for (int i = 1; i < num1 || i < num2; i++)//设置循环

{

if (num1 % i == 0 && num2 % i == 0)//如果i作为a,b的取余数均为0 就都是a,b的因数

{

temp = i;//将目前最大的i覆盖赋值到临时变量

}

}

return temp;//返回临时变量

}

int LCM(int num1, int num2)//定义最小公倍数函数

{

int temp;//定义临时变量

for (int i = num1; i <= (num1 \* num2); i++)//设置循环

{

if (i % num1 == 0 && i % num2 == 0)//如果i作为被取余数,分别取余a,b结果为0 就是a,b的公倍数

{

temp = i;//将公倍数赋值到临时变量

break;//找到的第一个公倍数就是最小公倍数 跳出循环

}

}

return temp;//返回临时变量

}

int main()

{

int a, b;//定义a,b两个变量

cout << "请输入数字a:";//提示输入a

cin >> a;//输入a

cout << "请输入数字b:";//提示输入b

cin >> b;//输入b

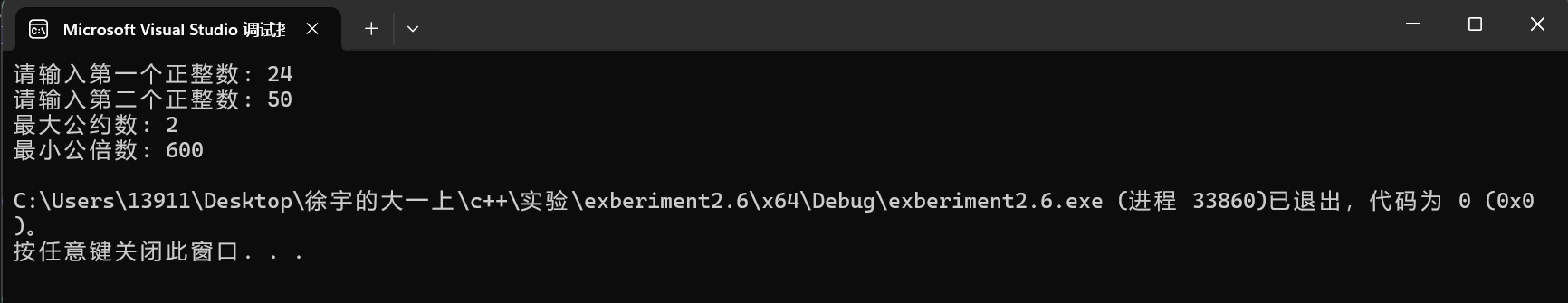
int gcd = Gcd(a, b);//调用最大公因数函数

int lcm = LCM(a, b);//调用最小公倍数函数

cout << "最大公因数为" << gcd << endl << "最小公倍数为" << lcm << endl;//输出结果

return 0;

}

****

**7.** #include<iostream>

using namespace std;

int main()

{

for (int i = 1; i < 6; i++)//定义一个循环 i为行数 一共五行 i<6

{

for (int j = 1; j <= i; j++)//定义嵌套循环 j为打印\*数

{

cout << '\*';//打印\*

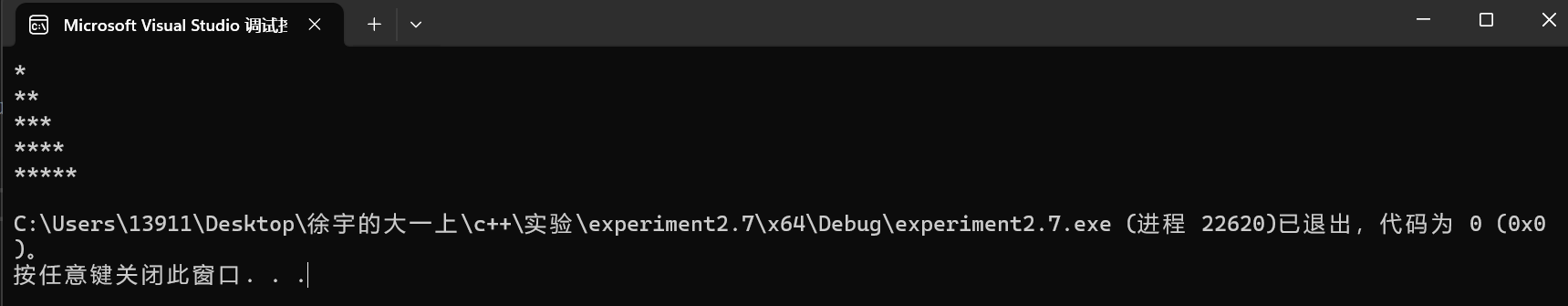
}

cout << endl;//本行打印完换行

}

return 0;

}

****

**8**#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

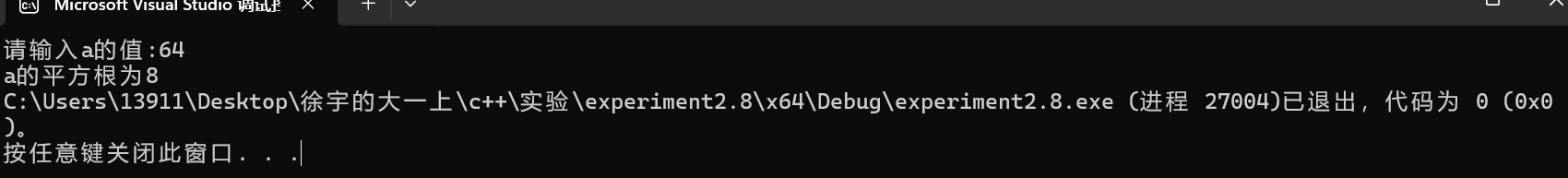
{

cout << "开平方需要a>=0" << endl;//提示不能开方

}

return 0;

}

****

**9.** #include<iostream>

using namespace std;

int main()

{

double price = 0.8, total\_p = 0;//定义单价及总价

int number = 2, day = 0;//定义购买数量及购买天数

double average\_p;//定义平均价格

for (int i = 1; number <= 100; i++)//循环 number超过100退出循环

{

day++;//天数+1

total\_p += price \* number;//总价=原先总价+每天新花的钱

number \*= 2;//数量\*2

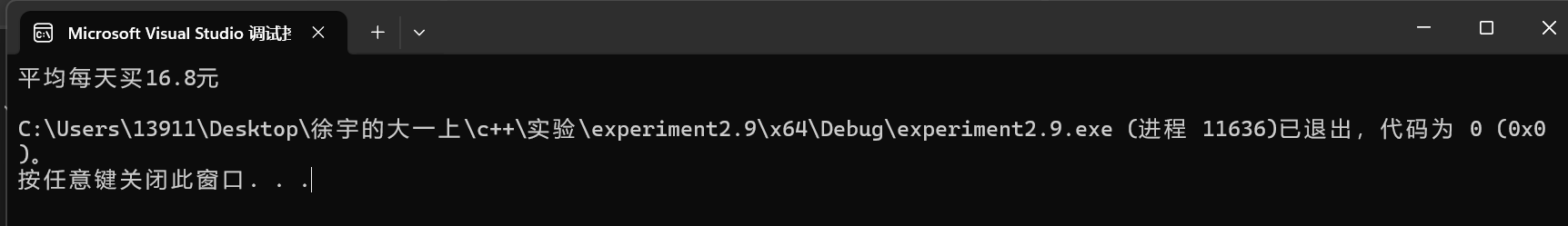
}

average\_p = total\_p / day;//平均价格=总价/天数

cout << "平均每天买" << average\_p << "元" << endl;//输出每天平均花费

return 0;

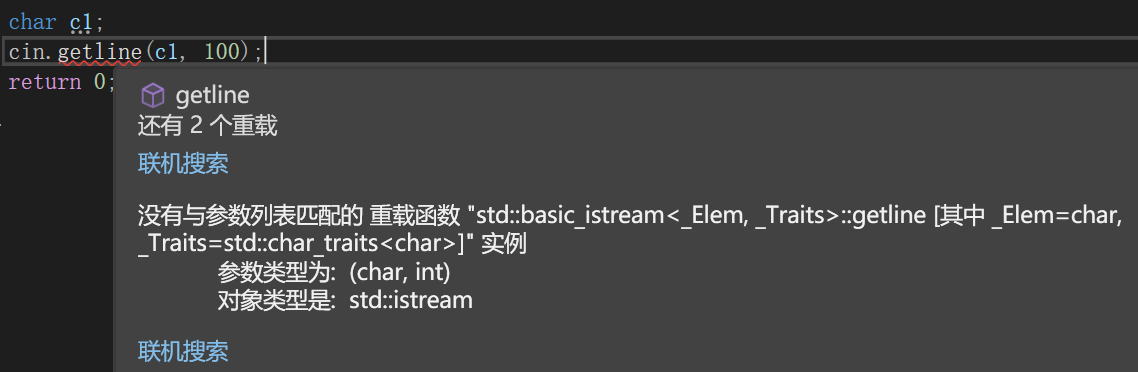
}

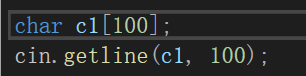
****

**四、遇到的问题与解决方法**

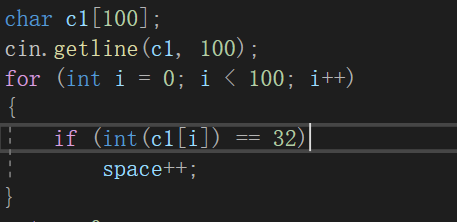
**1.不了解字符串中如何录入空格**

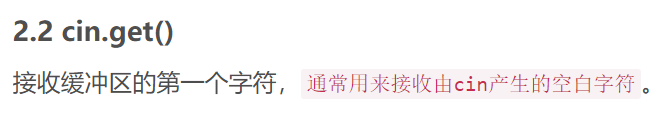
 最开始在互联网上查询相关资料，发现可以使用cin.getline()录入空格。

 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。

 考虑到在使用该函数式时有定义最大可接受字符量，尝试将单个字符变量改编为字符变量类型数组后再次尝试，报错消除。

考虑到定义的是数组，若想要检测每个字符，需要引入循环，故写出循环。

**** 但是考虑到在此前已经将c1写成了数组，如果在最开始录入的过程中，没有录满我们所设定的100个元素，后面的没有被录入的数组，可能会干扰到最后的运行结果，再加上我目前没有掌握设立动态数组的知识，故只能放弃这一方法。

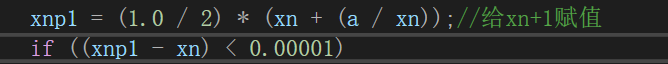
 再次在互联网上查找其他资料，发现cin.get（）也可以实现相关功能，故决定使用cin.get（）函数。

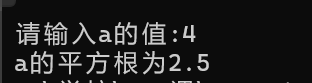
 但是在使用cin.get()函数时，因为不是很熟悉使用方法，发现在使用cin.get()录入后，再对结果进行输出，只能获得第一个字符。

} 后通过询问其他同学，了解到要想逐个对cin.get()所录入的字符分类,需要将其写入循环当中。

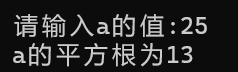
在这样写入后，程序能够对字符串中的每个字符进行判断，从而最终计算出各类型字符的数量。

**2.计算|xn+1-xn|时出现问题**

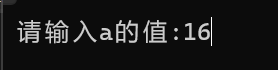
 在最开始计算|xn+1-xn|时，我的判断式写的是if（xnp1-xn<0.00001）。

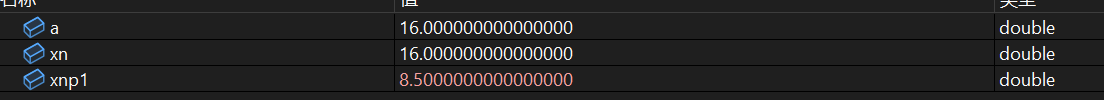
 但是在当时并没有注意到该错误，在直接运行程序后，发现结果不太准确。

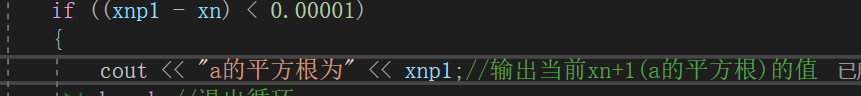
初步推测不准确有两种可能：（1）本算法本身带有一定误差；（2）程序某处出错。

 首先判断是否是由误差引起的不准确，分别计算25与16的平方根。

可见程序计算结果与其真实平方根相差极大，排除是误差引起的数据不准确。

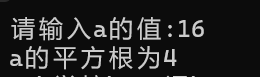
 接下来使用f11对程序进行逐步检查。在输入值时赋a=16。

 在进行if语句判断之前，各步骤计算均未出现错误。

 但是在各数据如上图所示时，if条件满足，直接执行了if条件里面的语句。

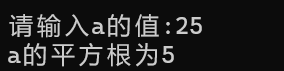
发现了是if的判断条件出错了，而要将if判断条件改正确，有两种方法：（1）(xnp1-xn<0.00001)&&(xn-xnp1<0.00001)（2）调用绝对值函数。

 首先尝试将判断条件修改为较为基础的逻辑表达式。

 程序可以正常运行。

随后尝试通过调用绝对值的函数，来修改if的判断条件。

 在互联网上查询了相关资料后，发现可以使用fabs（）函数，要引入相对应的头文件。

 在调用该函数后，发现可以正常计算，该方法也成立。

**五、体会**

C++上机实验心得体会

在经历了一系列C++上机实验后，我仿佛经历了一场奇妙的编程之旅，收获了满满的知识与宝贵的经验，内心充满了对这一过程的深刻感悟。

知识融合与深化理解

在理论学习C++时，各种概念像是散落在脑海中的拼图碎片。但在上机实验中，这些碎片开始逐渐拼接完整。从基本的数据类型，如整型、浮点型、字符型等，到复杂的指针和引用，每一个知识点都在实践中展现出了它们真实的面貌。例如，指针这个概念，在书本上理解起来总有些抽象，可当我在实验中利用指针操作数组、实现函数参数传递时，才真正领会到它的强大和精妙之处。它就像一把神奇的钥匙，打开了内存操作的新大门，让我对数据的存储和访问有了全新的认识。

类和对象的运用更是让我感受到了C++面向对象编程的魅力。通过创建类来模拟现实世界中的实体，比如设计一个银行账户类，其中包含账户余额、存款、取款等成员函数，我明白了如何将数据和对数据的操作封装在一起，提高代码的可维护性和可读性。继承和多态性则进一步拓展了这种编程模式的灵活性，让代码能够更好地适应复杂多变的需求。

调试技能的成长

调试是C++上机实验中不可或缺的一部分，也是最磨练人的环节之一。每一次程序报错，都是一次挑战。刚开始，面对满屏的编译错误或者运行时的逻辑错误，我常常感到不知所措。但随着实验次数的增加，我逐渐掌握了一些调试技巧。我学会了仔细阅读编译器给出的错误提示信息，这些信息就像是隐藏在代码迷宫中的线索，指引着我找到问题所在。有时，一个简单的语法错误，如括号不匹配或者分号遗漏，都可能引发一系列看似复杂的报错。通过逐步排查，我明白了不能被表面的错误现象所迷惑，要深入分析每一个可能的原因。

在处理运行时错误时，我掌握了使用调试工具设置断点、单步执行程序的方法。通过观察变量的值在程序运行过程中的变化，我能够追踪到逻辑错误发生的位置。这种追踪错误的过程就像是在解一道复杂的谜题，每一次找到问题的根源都让我充满成就感。而且，在这个过程中，我也培养了耐心和细心，不再因为错误而轻易烦躁，而是冷静地分析和解决问题。

编程风格与规范的重要性

良好的编程风格和规范在C++编程中至关重要，而上机实验让我深刻体会到了这一点。清晰、易读的代码不仅方便自己在后续的修改和维护，也有利于与其他同学或开发者进行交流。在给变量和函数命名时，我开始遵循有意义的命名规则，让名字能够直观地反映出其代表的含义。例如，用“studentScore”表示学生成绩，而不是随意使用“s”之类模糊的名称。

代码的缩进和排版也不再被我忽视。合理的缩进能够清晰地展示代码的逻辑结构，使程序的流程一目了然。在编写复杂的函数或者嵌套结构时，良好的排版就像一张清晰的地图，能够帮助我快速定位和理解代码的执行路径。同时，适当的注释也是代码的重要组成部分。在关键的代码段添加注释，解释其功能和目的，能够让自己和他人在阅读代码时迅速抓住重点，减少理解代码所需的时间。

团队协作与知识共享

在一些综合性的C++上机实验项目中，团队协作成为了实验成功的关键因素之一。与同学们一起讨论项目需求、设计算法和实现代码的过程中，我领略到了不同思维方式碰撞产生的火花。每个人都有自己独特的想法和见解，有的同学在算法优化方面有独到的思路，有的同学则擅长发现代码中的潜在问题。通过相互交流和学习，我们能够取长补短，共同提高。

在团队协作中，我们还学会了如何分工合作，发挥各自的优势。比如，有的同学负责编写核心算法部分，有的同学负责用户界面的设计，还有的同学负责测试和调试。这种分工方式使得整个项目的开发过程更加高效有序。同时，我们也会定期进行代码审查，互相检查对方的代码，提出改进建议。这不仅提高了代码的质量，也让我们从他人的代码中学到了新的编程技巧和方法。

总之，C++上机实验是一次极具挑战性和收获性的经历。它让我在知识、技能、编程习惯以及团队协作等方面都得到了全方位的提升。这些经验将成为我编程道路上的宝贵财富，激励我在未来的学习和实践中不断探索和进步。