**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： 8209240228

姓 名： 曾栋湘

**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。**

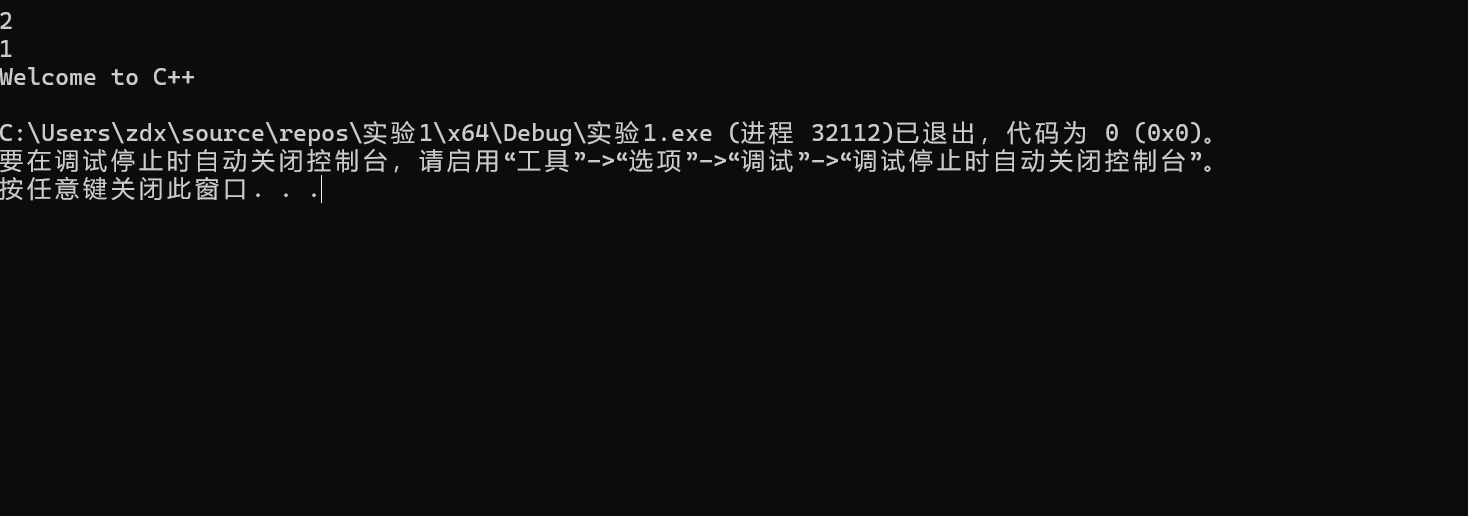
**5.编程，输入华氏温度，将其转换为摄氏温度后输出（保留两位小数）。**

**三、算法分析，程序结果**

1.一.Int错误应为int；

二.k未定义，应先定义k=0；

三.i重复定义，删去第二个定义。

****

2.#include<iostream>

#include <iomanip>

using namespace std;

#define pi 3.14159265358979323846;

int main()

{

double h, r;

cout << "请输入圆锥高度" << endl;

cin >> h;

cout << "请输入圆锥半径" << endl;

cin >> r;

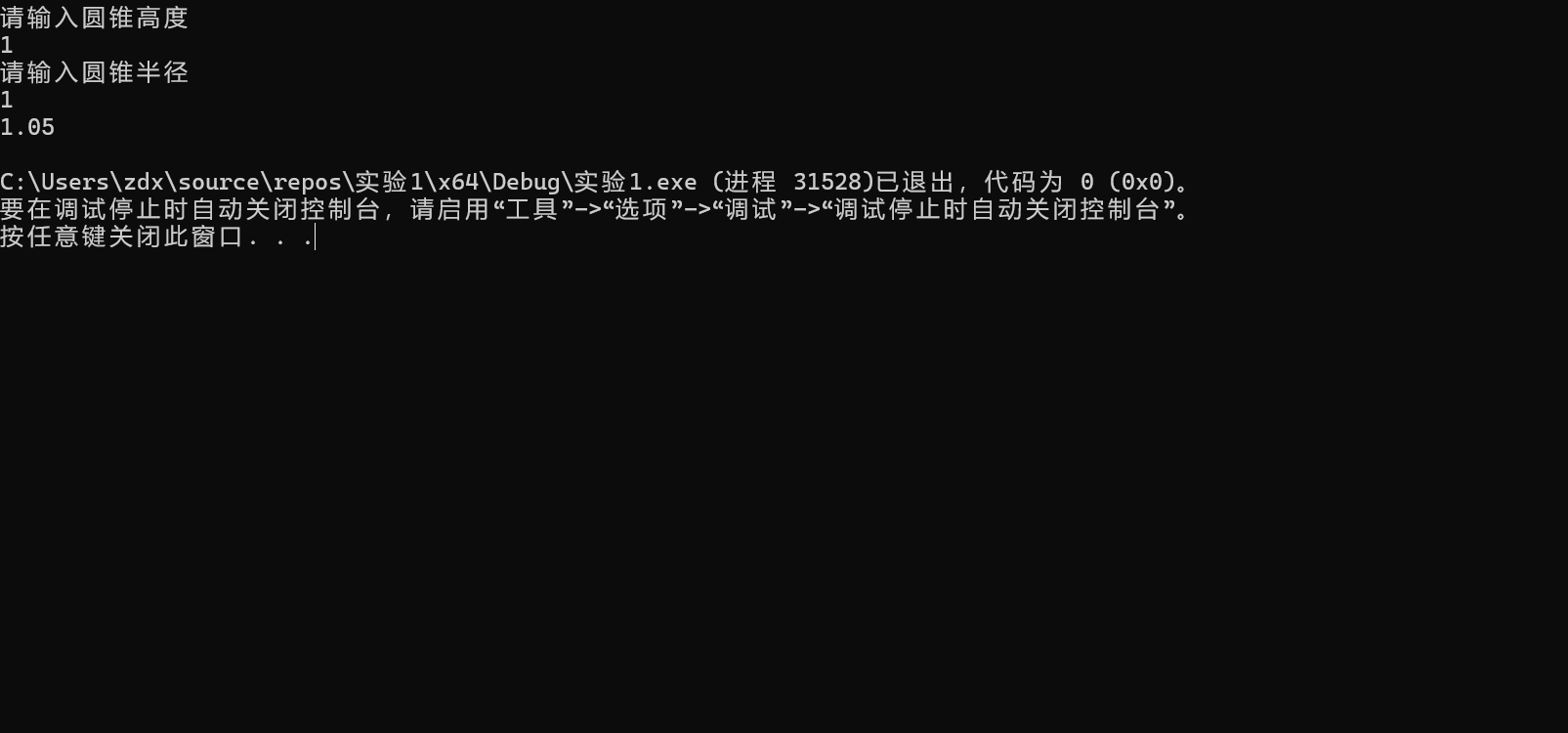
double V = 1.0 / 3 \* r \* r \* h \* pi;

cout << fixed << setprecision(2);

cout << V << endl;

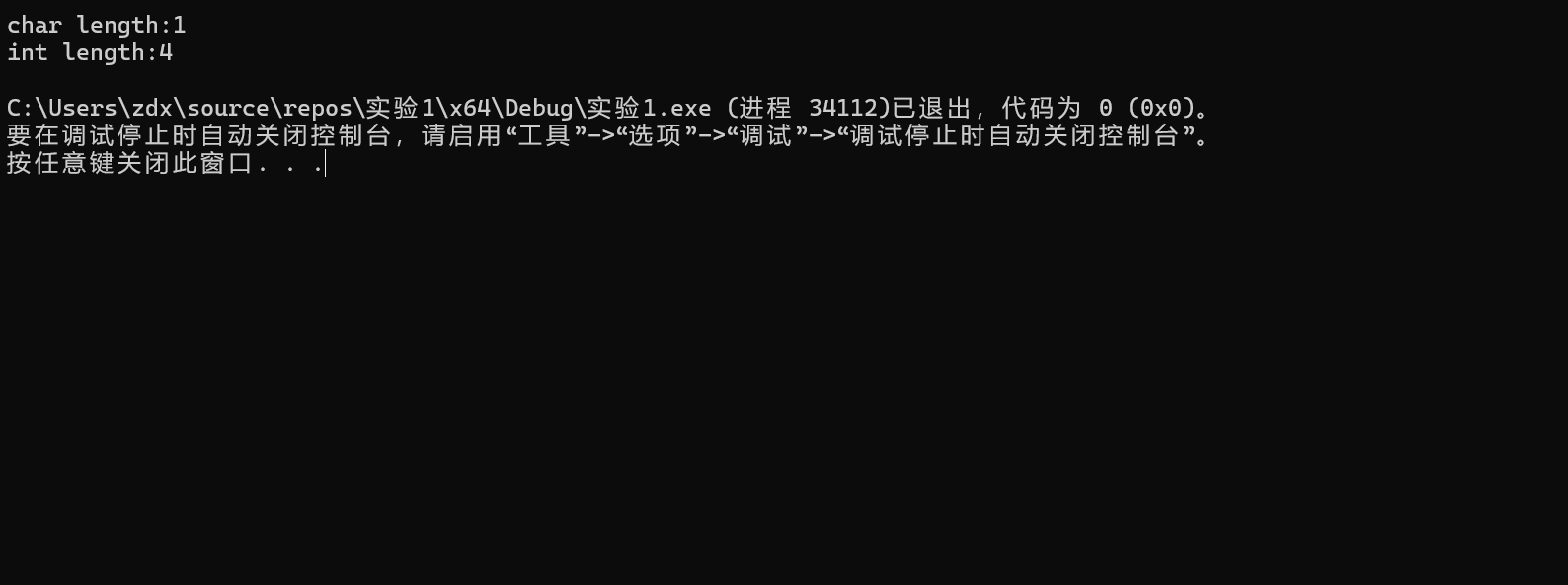
return 0;

}



3.char length:1

int length:4



4.output in unsigned int 1 type:65534

output in char type:!

output in short type:177776

output in int type:177776

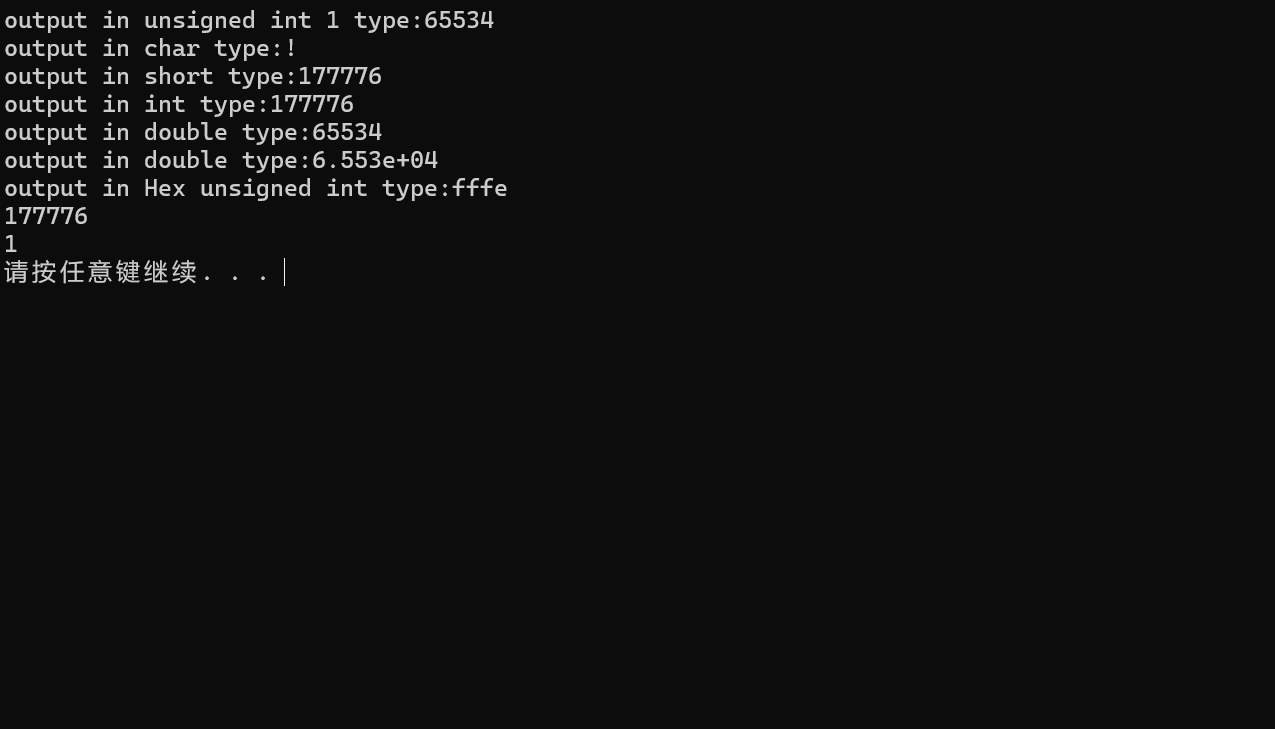
output in double type:65534

output in double type:6.553e+04

output in Hex unsigned int type:fffe

177776

1



5.#include<iostream>

#include <iomanip>

using namespace std;

int main()

{

double C, H;

cout << "请输入华氏温度：" << endl;

cin >> H;

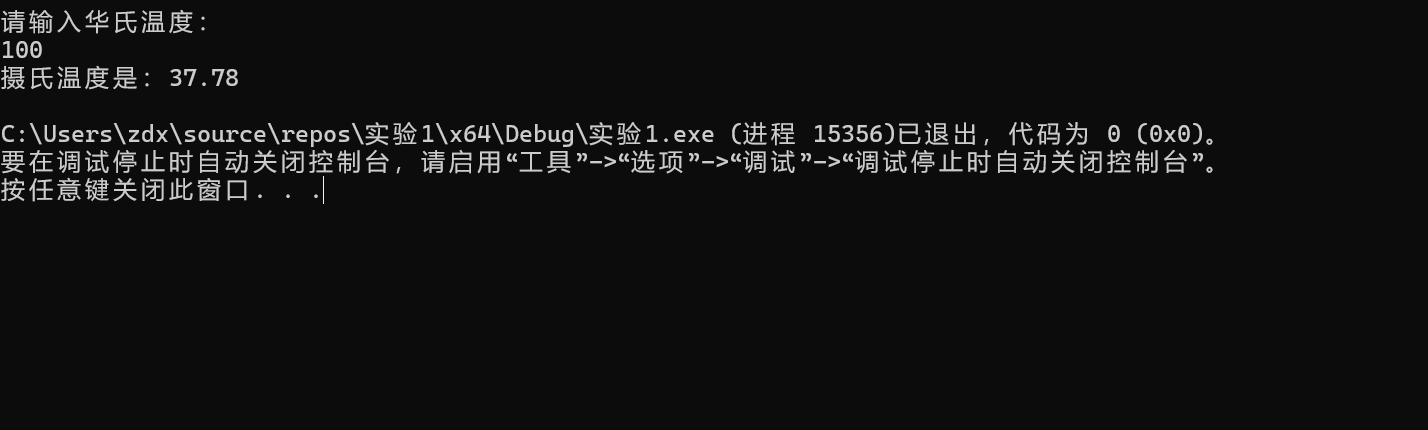
C = (H - 32) \* 5 / 9;

cout << fixed << setprecision(2);

cout << "摄氏温度是：" << C << endl;

return 0;

}



**四、遇到的问题与解决方法**

1.不懂如何定义Π，上网查。

2.不会华氏温度与摄氏温度的关系，上网查。

**五、体会**

**要多加练习，不会的内容要及时查询。**

**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>

#include <iomanip>

using namespace std;

int main()

{

char a\_char;

cout << "请输入一个字母：" << endl;

cin >> a\_char;

if (islower(a\_char))

{

cout << static\_cast<char>(toupper(a\_char)) << endl;

}

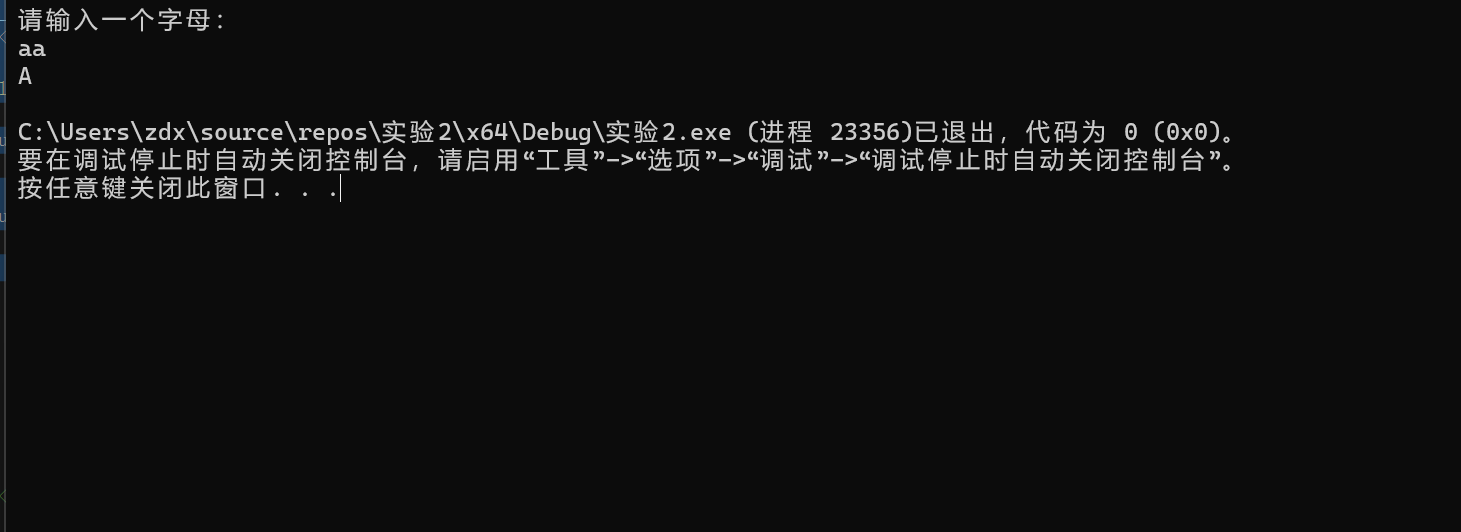
else {

cout << static\_cast<int>(a\_char + 1) << endl;

}

return 0;

}



**2.**int main()

{

double x, y;

cin >> x;

if (1> x > 0) {

y = 3 - (2 \* x);

cout << y << endl;

}

else if (5 > x >= 1 ) {

y = 1 / (2 \* x) + 1;

cout << y << endl;

}

else {

y = x \* x;

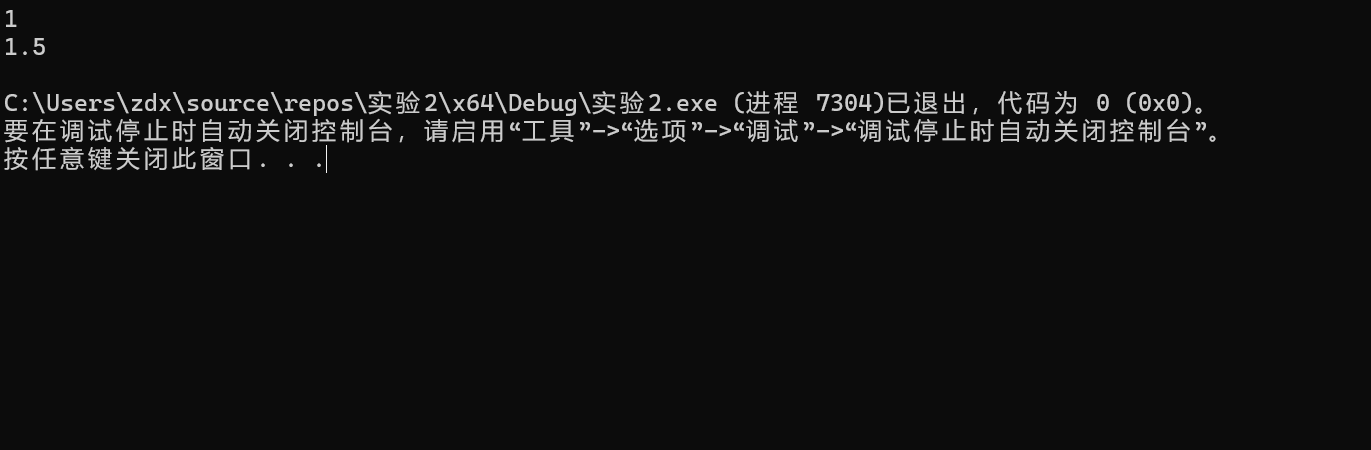
cout << y << endl;

}

return 0;

}

2.6,1.5,25,无结果



**3.**#include<iostream>

#include <iomanip>

using namespace std;

int main()

{

double x, y, z;

cout<< "请输入第一条边" << endl;

cin >> x;

cout << "请输入第二条边" << endl;

cin >> y;

cout << "请输入第三条边" << endl;

cin >> z;

if (x + y < z) {

cout << "错误，不是三角形" << endl;

}

else if (x + z < y) {

cout << "错误，不是三角形" << endl;

}

else if (y + z < x) {

cout << "错误，不是三角形" << endl;

}

else {

double C;

C = x + y + z;

cout << C << endl;

if (x = y) {

cout << "是等腰三角形" << endl;

}

else if (x = z) {

cout << "是等腰三角形" << endl;

}

else if (y = z) {

cout << "是等腰三角形" << endl;

}

else if (x = y = z) {

cout << "是等腰三角形" << endl;

}

else {

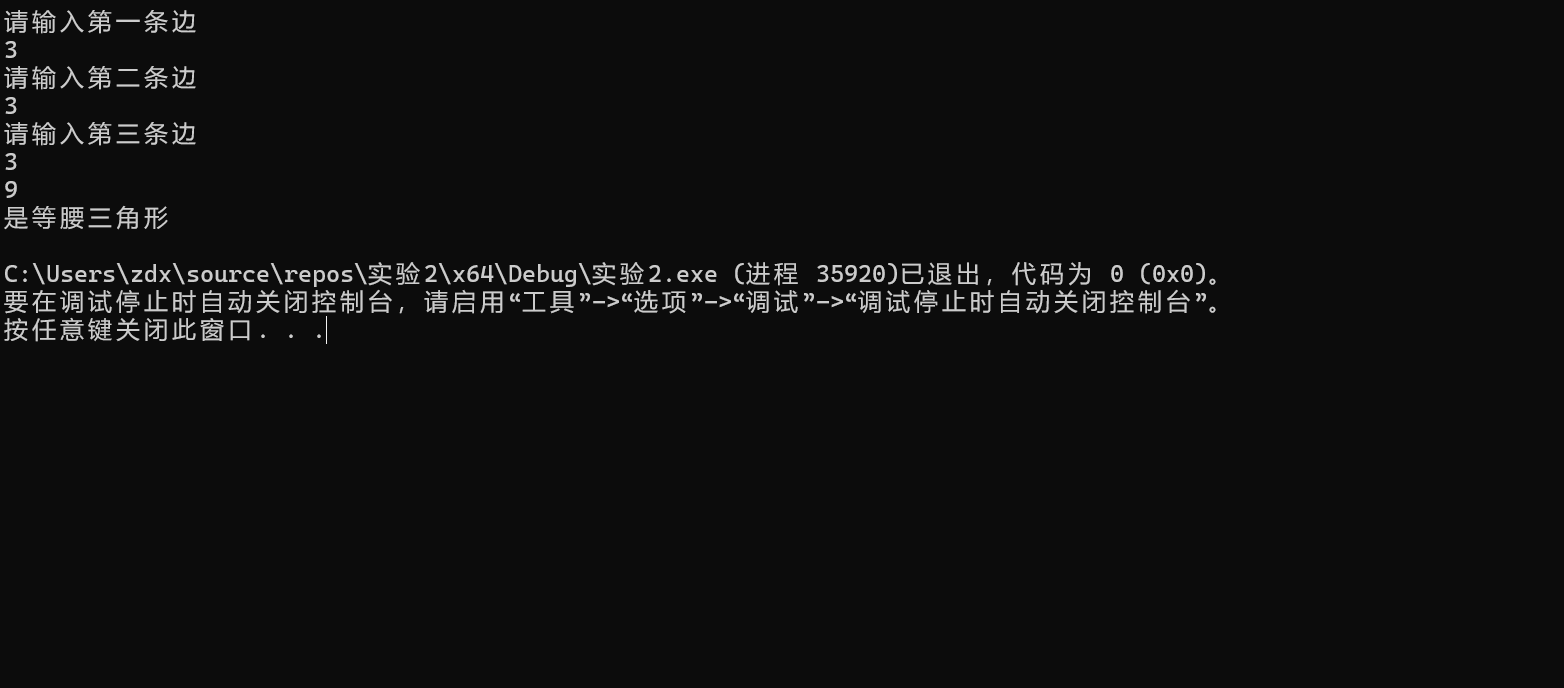
cout << "不是等腰三角形" << endl;

}

}

return 0;

}



**4.**#include <iostream>

using namespace std;

int main() {

double num1, num2;

char Symbol;

cout << "请输入第一个数字：";

cin >> num1;

cout << "请输入运算符（+、-、\*、/、%）：";

cin >> Symbol;

cout << "请输入第二个数字：";

cin >> num2;

double result;

switch (Symbol) {

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，请重新输入！" << endl;

return 1;

}

result = num1 / num2;

break;

case '%':

if (num2 == 0) {

cout << "除数不能为0，请重新输入！" << endl;

return 1;

}

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

break;

default:

cout << "非法运算符，请重新输入！" << endl;

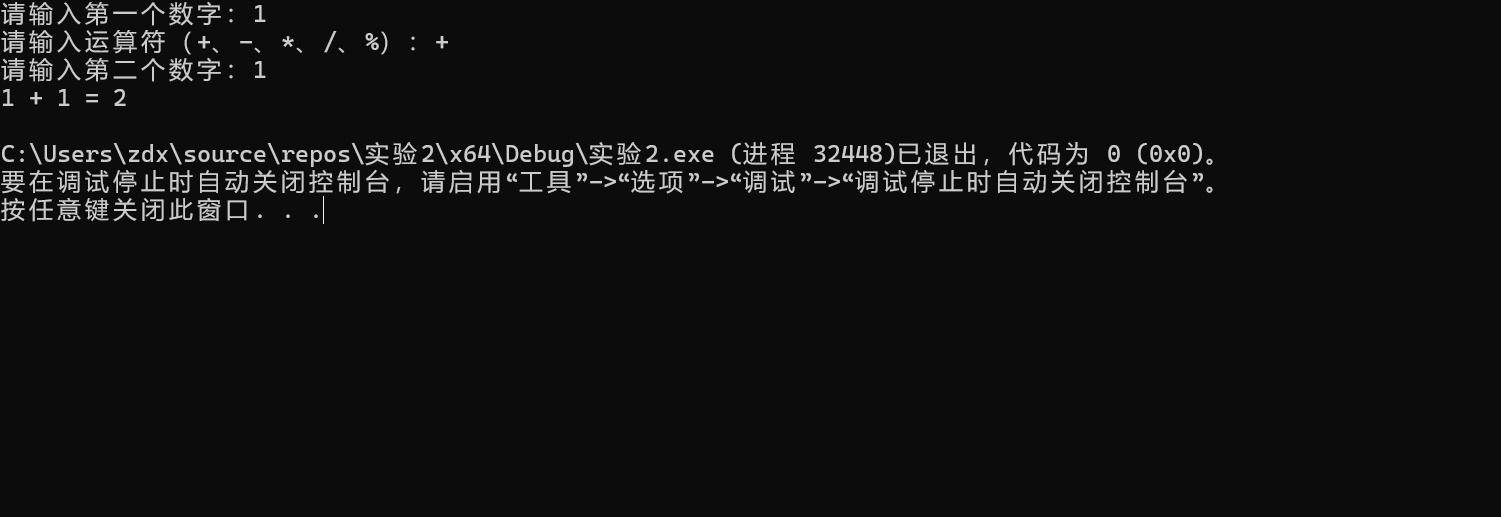
return 1;

}

cout << num1 << " " << Symbol << " " << num2 << " = " << result << endl;

return 0;

}



**5.**#include <iostream>

#include <string>

using namespace std;

int main() {

string inputStr;

int letterCount = 0;

int spaceCount = 0;

int digitCount = 0;

int otherCount = 0;

getline(cin, inputStr);

for (char c : inputStr) {

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

letterCount++;

}

else if (c == ' ') {

spaceCount++;

}

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

digitCount++;

}

else {

otherCount++;

}

}

cout << "英文字母的个数：" << letterCount << endl;

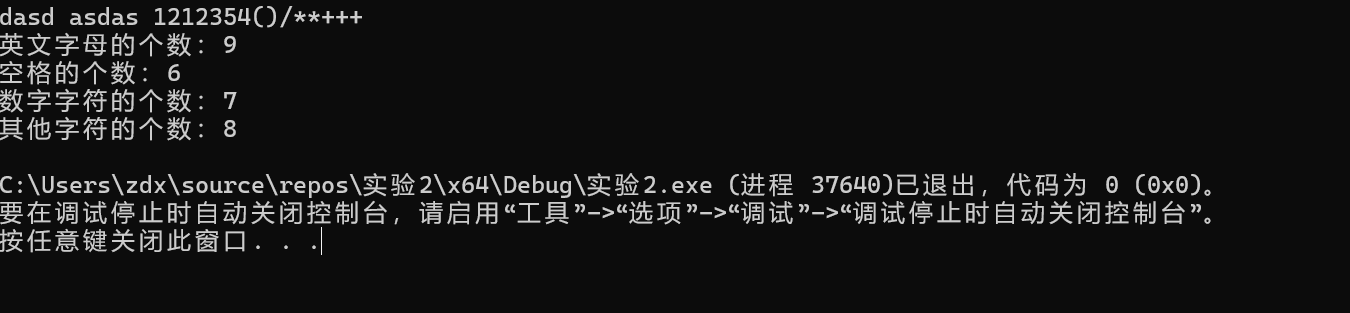
cout << "空格的个数：" << spaceCount << endl;

cout << "数字字符的个数：" << digitCount << endl;

cout << "其他字符的个数：" << otherCount << endl;

return 0;

}



**6.**#include <iostream>

#include <string>

using namespace std;

int main() {

int a, b;

int temp;

cout << "请输入a" << endl;

cin >> a;

cout << "请输入b" << endl;

cin >> b;

while (b != 0) {

int temp = b;

b = a % b;

a = temp;

}

int x = a;

cout << "请再输入a" << endl;

cin >> a;

cout << "请再输入b" << endl;

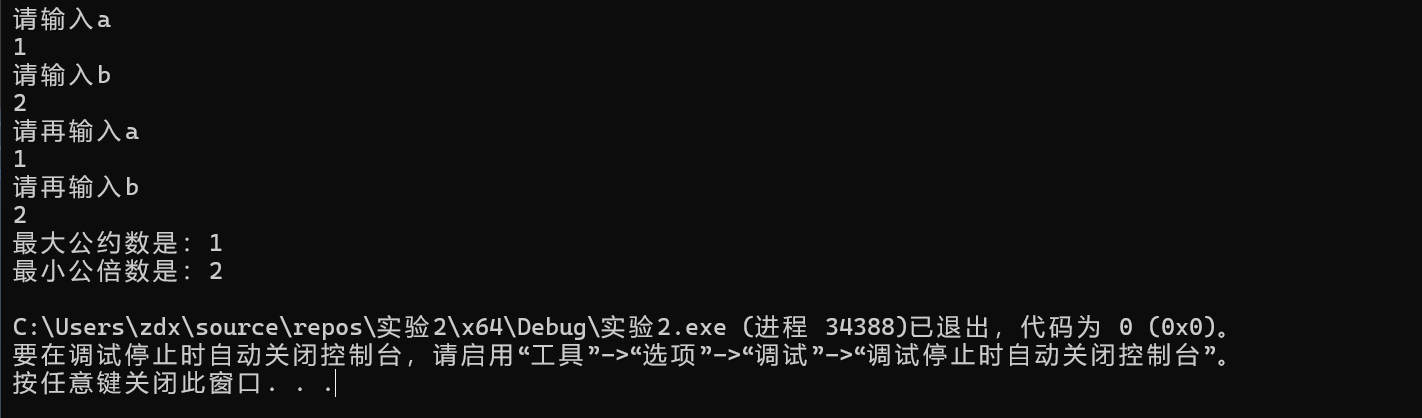
cin >> b;

cout << "最大公约数是：" << x << endl;

cout << "最小公倍数是：" << (a \* b) / x << endl;

return 0;

}



**7.**#include <iostream>

#include <string>

using namespace std;

int main() {

int i,j;

for (i = 1; i < 6; i++) {

for (j = 0; j < i; j++) {

cout << "\*" ;

}

cout << "\n" ;

}

return 0;

}



**8.**#include <iostream>

#include <string>

using namespace std;

int main() {

double a;

double xn;

double xn\_p\_1;

cout << "请输入a" << endl;

cin >> a;

bool isNegative = false;

if (a < 0) {

isNegative = true;

a = abs(a);

}

xn = a;

do {

xn\_p\_1 = 0.5 \* (xn + a / xn);

if (abs(xn\_p\_1 - xn) < 1e-5) {

break;

}

xn = xn\_p\_1;

} while (true);

if (isNegative) {

cout << "输入的数字a为负，其平方根为虚数：" << sqrt(-a) << "i" << endl;

}

else {

cout << "数字a的平方根为：" << xn\_p\_1 << endl;

}

}

（1）如果直接按照原始迭代公式计算，当输入a为负时，在迭代过程中会出现对负数进行开平方的情况（因为迭代公式中有a / xn这一项），这在实数范围内是无定义的，程序会得到错误的结果或者出现运行时错误（取决于具体的编译器和运行环境）。

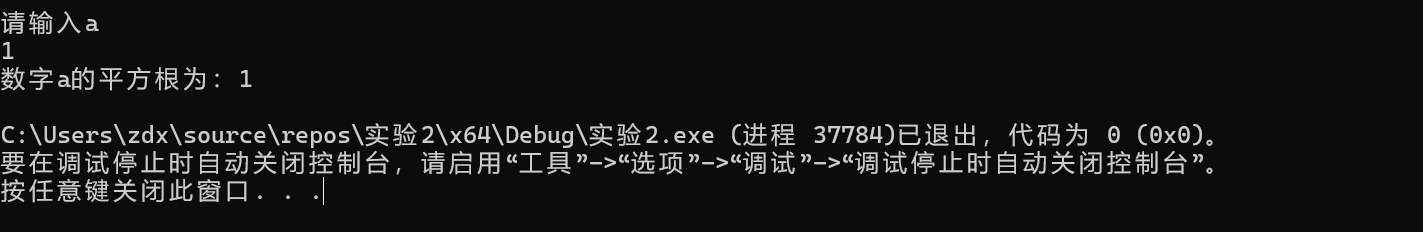
在上述修改后的程序中，当检测到输入a为负时，先将a取绝对值进行正常的迭代计算求出其绝对值的平方根，然后在输出时根据记录的符号信息，以虚数的形式输出结果（即输出sqrt(-a)并添加i表示虚数单位），这样就可以处理任何输入的a值了。

（2）理论上可以尝试设置更小的精度要求，比如|xn + 1 - xn| < 1e - 5。但在实际应用中，存在以下限制：

浮点数精度限制：计算机中的浮点数是用有限的二进制位来表示的，存在一定的精度误差。当迭代的精度要求设置得过高（如小于计算机能准确表示的浮点数精度）时，可能会出现迭代无法终止的情况。因为由于浮点数的舍入误差，即使在理论上已经达到了所需的精度，但实际计算中由于舍入等原因，可能永远无法满足|xn + 1 - xn| < 10 - 10这样非常严格的条件。

计算时间和资源：设置更小的精度要求通常意味着需要更多的迭代次数才能满足条件。这会导致程序的计算时间大幅增加，并且可能占用更多的计算机资源（如内存、CPU 时间等）。

在实际测试中，如果将程序中的精度要求改为|xn + 1 - xn| < 1e - 5，对于一些简单的输入值可能会正常终止并得到结果，但对于某些值可能会出现迭代时间过长甚至无法终止的情况，具体取决于输入值和计算机的硬件环境等因素。



**9.**#include <iostream>

#include <string>

using namespace std;

int main() {

double m = 0.8;

int f = 2;

double tc = 0;

int td = 0;

int c = f;

while (c < 100) {

double ct = static\_cast<double>(c) \* m;

tc += ct;

td += 1;

c \*= 2;

}

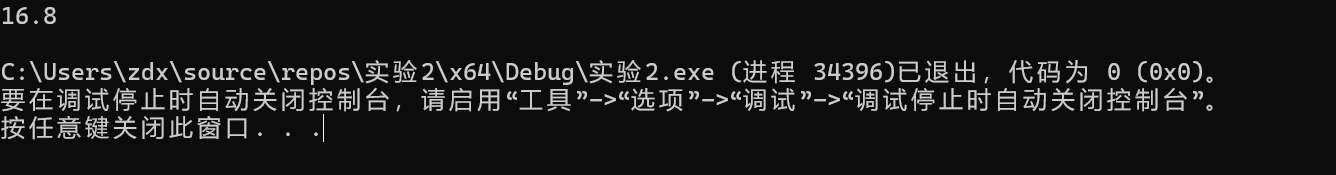
double acpd = tc / static\_cast<double>(td);

cout << acpd << endl;

return 0;

}

16.8



**四、遇到的问题与解决方法**

**1.不知如何表达大小写字母，上网查；2.搜寻识别空格的办法。**

**五、体会**

**要多学习，多搜寻资料。**