实验五

有以下程序：

#include<iostream>

using namespace std；

class Time

// 定义 Time 类

{

public：

// 数据成员为公用的

int hour;

int minute;

int sec ;

};

int main()

{

Time tl;

//定义 t1 为 Time 类对象

cin>>t1.hour;

//输入设定的时间

cin>>t1.minute;

cin>>t1.sec;

cout<<t1.hourl<<”：”<<t1.minute<<”：”<<t1.sec<<endl ;

return 0;

}

改写程序，要求：

(1)将数据成员改为私有的；

(2)将输入和输出的功能改为由成员函数实现；

(3)在类体内定义成员函数。

然后编译和运行程序。请分析什么成员应指定为公用的?什么成员应指定为私有的? 什么函

数最好放在类中定义? 什么函数最好在类外定义?

#include<iostream>

using namespace std;

class Time

{

public:

void input()

{

cin >> hour;

cin >> minute;

cin >> sec;

}

void output()

{

cout << hour << ":" << minute << ":" << sec << ":" << endl;

}

private:

int hour;

int minute;

int sec;

};

int main()

{

Time t1;

cout << "输入时间" << endl;

t1.input();

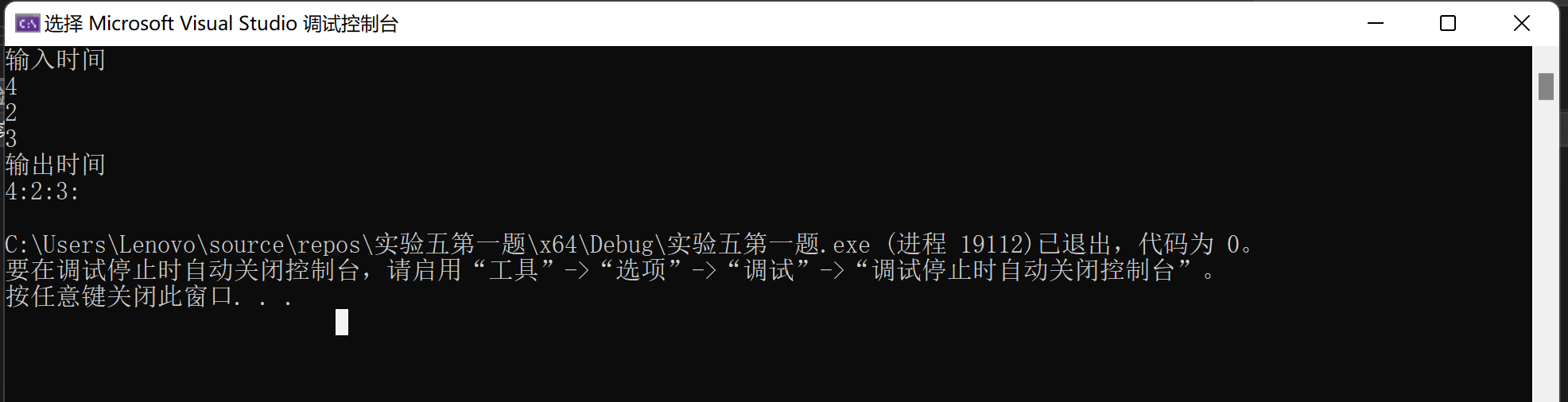
cout << "输出时间" << endl;

t1.output();

return 0;

}

结果：



2、分别给出如下的 3 个文件：

(1)含类定义的头文件 student.h，

//student.h

(这是头文件，在此文件中进行类的声明)

class Student

//类声明

{

public:

//公用成员函数原型声明

void display();

private:

int num;14

char name[20];

char sex;

};

(2)包含成员函数定义的源文件 student.cpp

//student.cpp

在此文件中进行函数的定义

#include <iostream>

#include”student.h”

//不要漏写此行，否则编译通不过

void Student::display()

//在类外定义 display 类函数

{

cout<<”num：”<<num<<endl;

cout<<”name：”<<name<<endl;

cout<<”sex：”<<sex<<endl;

}

(3)包含主函数的源文件 main.cpp。

为了组成一个完整的源程序，应当有包括主函数的源文件：

//main.cpp

主函数模块

#include <iostream>

//将类声明头文件包含进来

#include “student.h”

int main()

{

Student stud;

//定义对象

Student stud1(007,”tcg”,’m’);

stud.display();

//执行 stud 对象的 display 函数

return 0;

}

请完善该程序，在类中增加一个对数据成员赋初值的成员函数 set\_value。上机调试并运行。

student.h

#include<iostream>

#include"student.h"

using namespace std;

int main()

{

Student stud;

stud.set\_value(007, "tcg", 'm');

stud.display();

return 0;

}

Student.cpp

#include <iostream>

#include"student.h" //不要漏写此行，否则编译通不过

using namespace std;

void Student::display() //在类外定义 display 类函数

{

cout << "num：" << num << endl;

cout << "name:" << name << endl;

cout << "sex:" << sex << endl;

}

void Student::set\_value(int n,const char na[], char s)

{

num = n;

for (int i = 0; i < 20; i++)

{

name[i] = na[i];

}

sex = s;

}

Main.cpp

#include<iostream>

#include"student.h"

using namespace std;

int main()

{

Student stud;

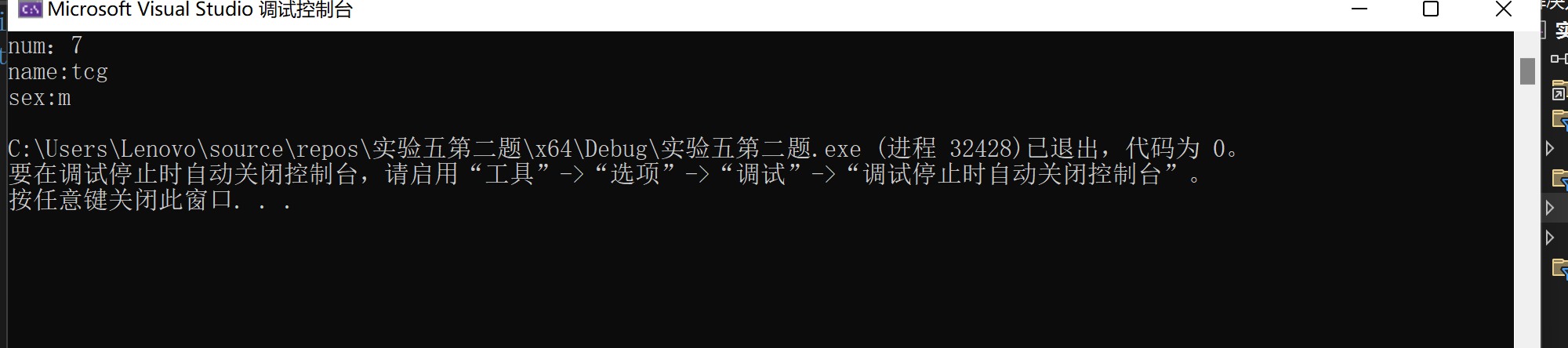
stud.set\_value(007, "tcg", 'm');

stud.display();

return 0;

}

结果



3、需要求 3 个长方柱的体积，请编一个基于对象的程序。数据成员包括 length(长)、width(宽)、

height(高)。要求用成员函数实现以下功能：

(1)由键盘分别输入 3 个长方柱的长、宽、高；

(2)计算长方柱的体积；

(3)输出 3 个长方柱的体积。

请编程序，上机调试并运行。

#include<iostream>

using namespace std;

class changfangzhu

{

public:

changfangzhu(double l, double w, double h)

{

length = l;

width = w;

height = h;

v = l \* w \* h;

}

void show()

{

cout << "体积为" << v << endl;

}

double length;

double width;

double height;

double v;

};

int main()

{

double l1, w1, h1;

cout << "输入长宽高" << endl;

cin >> l1 >> w1 >> h1;

changfangzhu c1(l1,w1,h1);

c1.show();

double l2, w2, h2;

cout << "输入长宽高" << endl;

cin >> l2 >> w2 >> h2;

changfangzhu c2(l2,w2,h2);

c2.show();

double l3, w3, h3;

cout << "输入长宽高" << endl;

cin >> l3 >> w3>> h3;

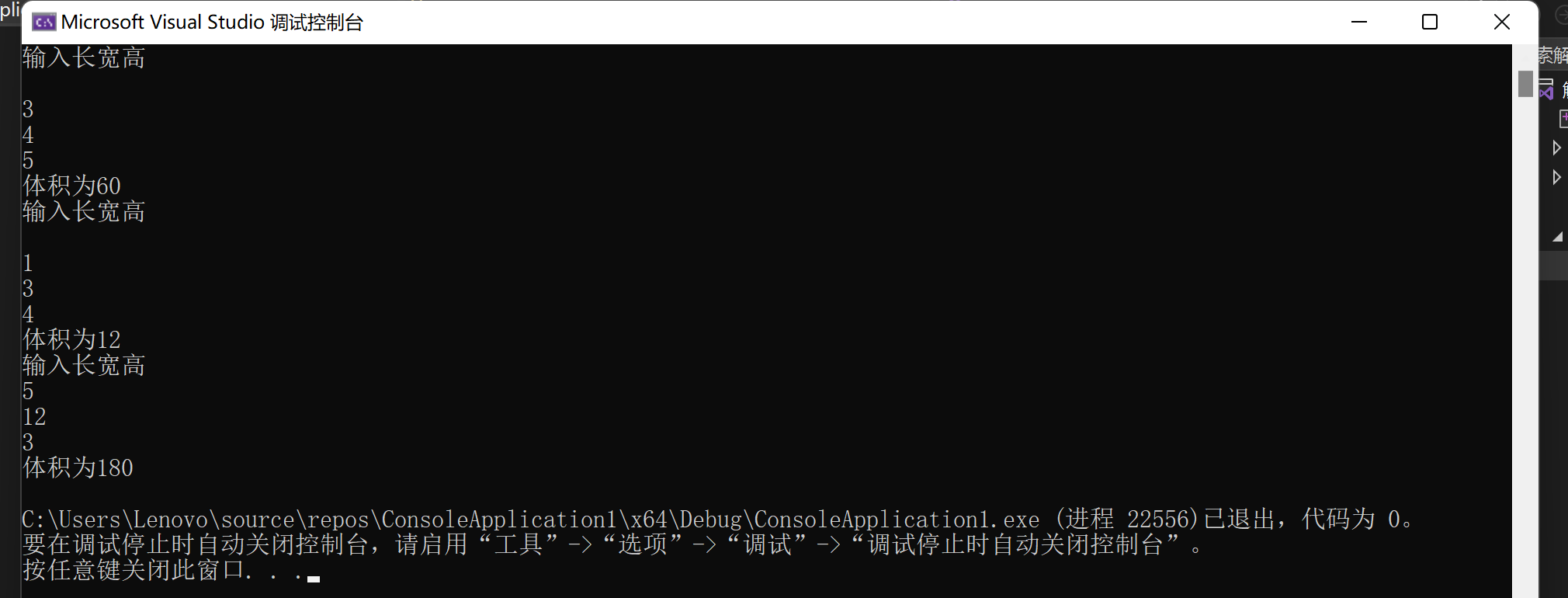
changfangzhu c3(l3,w3,h3);

c3.show();

return 0;

}

结果：



4、编写程序：定义抽象基类 Shape，由它派生出五个派生类：Circle（圆形）、Square（正

方形）、Rectangle（ 长方形）、Trapezoid （梯形）和 Triangle （三角形），用虚函数分别计算

各种图形的面积，并求出它们的和。要求用基类指针数组。使它的每一个元素指向一个派生类的

对象。

注：主函数中定义如下对象

Circle circle(10.2);

Square square(3);

Rectangle rectangle(3,4);

Trapezoid trapezoid(2.0,4.5,3);

Triangle triangle(4,5,6);

#include<iostream>

#include<cmath>

using namespace std;

class Shape

{

public:

double s;

virtual double area()

{

s = 0.0;

return s;

}

};

const double pi = 3.14159;

class Circle :public Shape

{

private:

double r;

public:

Circle(double R)

{

r = R;

}

double area()

{

s = pi \* r \* r;

return s;

}

};

class Square :public Shape

{

private:

double x;

public:

Square(double X)

{

x = X;

}

double area()

{

s = x \* x;

return s;

}

};

class Rectangle :public Shape

{

private:

double x, y;

public:

Rectangle(double X,double Y)

{

x = X;

y = Y;

}

double area()

{

s = x \* y;

return s;

}

};

class Trapezoid :public Shape

{

private:

double x, y, h;

public:

Trapezoid(double X, double Y, double H)

{

x = X;

y = Y;

h = H;

}

double area()

{

s = (x + y) \* h \* 0.5;

return s;

}

};

class Triangle :public Shape

{

private:

double x, y, z;

public:

Triangle(double X, double Y, double Z)

{

x = X;

y = Y;

z = Z;

}

double area()

{

double p = (x + y + z) \* 0.5;

s = pow(p \* (p - x) \* (p - y) \* (p - z), 2);

return s;

}

};

int main()

{

double n = 0;

Circle circle(10.2);

Square square(3);

Rectangle rectangle(3, 4);

Trapezoid trapezoid(2.0, 4.5, 3);

Triangle triangle(4, 5, 6);

Shape\* s[5];

s[0] = &circle;

s[1] = &square;

s[2] = &rectangle;

s[3] = &trapezoid;

s[4] = &triangle;

for (int i = 0; i < 5; i++)

{

n = n + s[i]->area();

if (i == 4)

{

cout << "面积和为：" << n << endl;

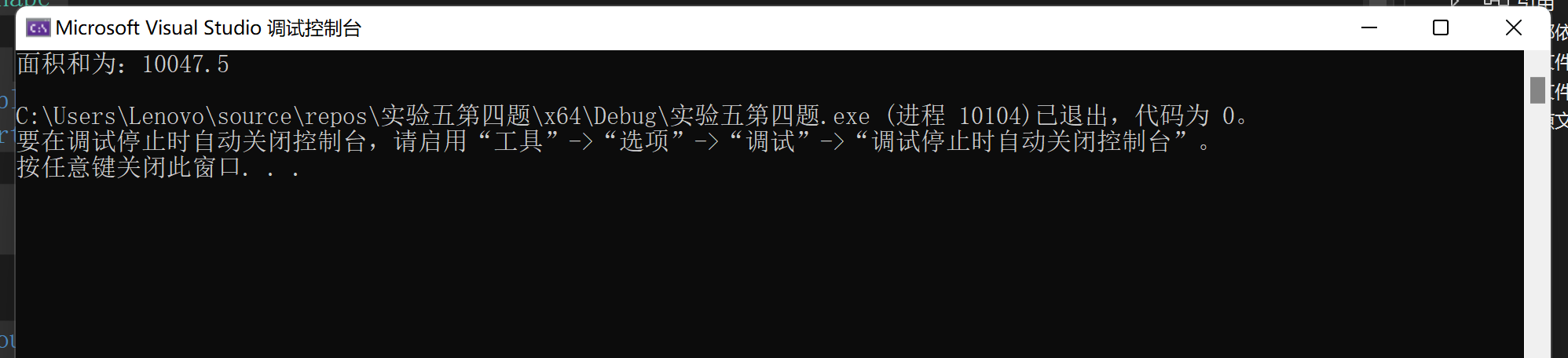
}

}

return 0;

}

结果



5、设计一个函数，返回两个向量的交集，函数头如下：

template<typename T>

vector<T> intersection(const vector<T>& v1, const vector<T>& v2)

两个向量的交际为它们的共同包含的元素，编写一个测试程序，提示用户输入两个向量，每

个包含 5 个字符串，打印输出它们的交集。

#include<iostream>

#include<vector> //vector

#include<string>

#include <algorithm> //sort

using namespace std;

template<typename T>

vector<T> intersection(vector<T>& v1, vector<T>& v2)

{

vector<T> v;

sort(v1.begin(), v1.end());

sort(v2.begin(), v2.end());

set\_intersection(v1.begin(), v1.end(), v2.begin(), v2.end(), back\_inserter(v));

return v;

} //求两向量的交集

int main()

{

string i;

vector<string>v1, v2, v;

cout << "输入5个字符串：" << endl;

while (1)

{

getline(cin, i);

static int h = 0;

v1.push\_back(i);

h++;

if (h == 5)

{

break;

}

};

cout << endl;

cout << "再输入5个字符串：" << endl;

while (1)

{

getline(cin, i);

static int h = 0;

v2.push\_back(i);

h++;

if (h == 5)

{

break;

}

};

v = intersection(v1, v2);

cout << "交集为：";

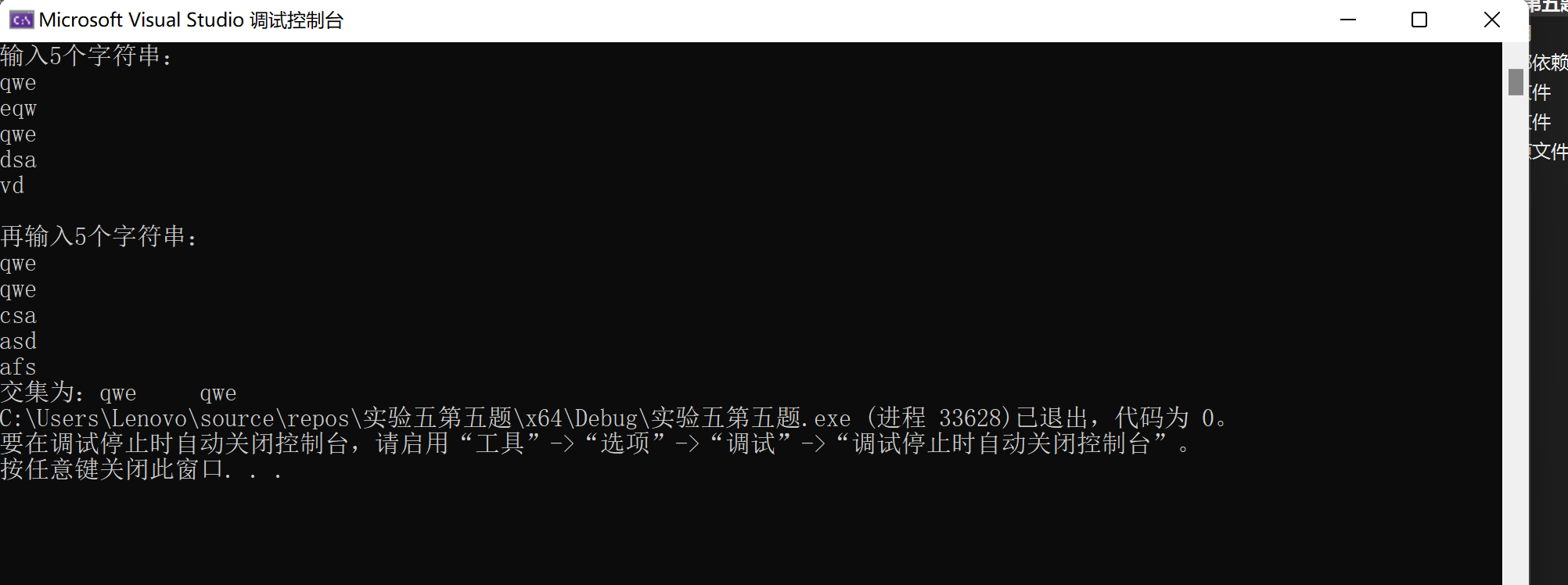
for (string n : v)

cout << n << '\t';

return 0;

}

结果



遇到的问题：

有一些操作不太熟练，没怎么使用，不太会。

解决问题：

通过搜索知道一些基本的使用，同时请教同学，了解使用方法。

体会：

经过一个学期的打程序，对计算机c＋＋的程序了解更深了，学会了很多实用的方法。