**计算机程序设计基础（C++)**

**实验报告**

专业班级： 软件工程2404

学 号： 8209240417

姓 名： 杨祎谌

**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：**

**实验三 函数**

**一、实验目的**

本实验主要培养、训练学生对函数的理解，要求：

1. 掌握函数的定义、声明的方法；

2. 掌握函数的编写要求；

3. 掌握函数的调用方法；

4. 掌握函数参数的传递方法；

5. 掌握变量的作用域；

6. 掌握多文件编程方法。

**二、实验内容与要求**

1、输入自然数m和n，

（1）求他们的最大公约数（或称最大公因数）。

要求输入、输出在主函数中进行，求公约数。

1. 在函数中求最大公约数与最小公倍数。（提示：使用引用参数）

2. 编写程序满足：声明一个函数，判断一个整数是否为素数，使用如下函数头：

bool is\_prime(int num) ,如果num是素数函数返回true，否则返回false；

利用函数is\_prime找出前200个素数，并按每行10个输出：

     2     3      5      7    11    13    17    19    23    29

3、编程实现摄氏温度到华氏温度的转换：

编写一个头文件，包含下面两个函数：

double celsius\_to\_fah(double cel)    //摄氏温度到华氏温度

double fahrenheit\_to\_cels(double fah) //华氏温度到摄氏温度

实现头文件，并编写测试程序，调用函数显示如下结果：

Celsius    Fahrenheit   |   Fahrenheit     Celsius

40.0       105.0        |   120.0            48.89

39.0       102.0        |   110.0            43.33

……       ……        |   ……             ……

31.0        87.8        |   30.0             -1.11

（测试程序为主模块，即main( )函数所在的CPP文件，头文件mytemperature.h只有函数声明；函数定义写在另一CPP文件mytemperature.cpp）

4、创建名为mytriangle.h的头文件，包括：

bool is\_valid(double side1,double side2,double side3)

double\_area(double side1,double side2, double side3)

面积=sqrt(s(s-side1)(s-side2)(s-side3))

其中s=(side1+side2+side3)/2

写测试程序：读取三角形三边长，如输入合法，计算面积，否则输出错误信息。

（测试程序为主模块，即main( )函数所在的CPP文件，头文件mytriangle.h只有函数声明；函数定义写在另一CPP文件mytriangle.cpp）

**3与4选一个完成**

5、猴子吃桃：猴子第一天摘若干桃子，当即吃了一半，还不过瘾，又吃了一个。第二天又将剩下的桃子吃掉一半，又多吃一个，以后每天如此，到第10天，发现只剩最后一个桃子，问，第一天猴子共摘多少桃子（用递归实现）。

**三、实验思考题**

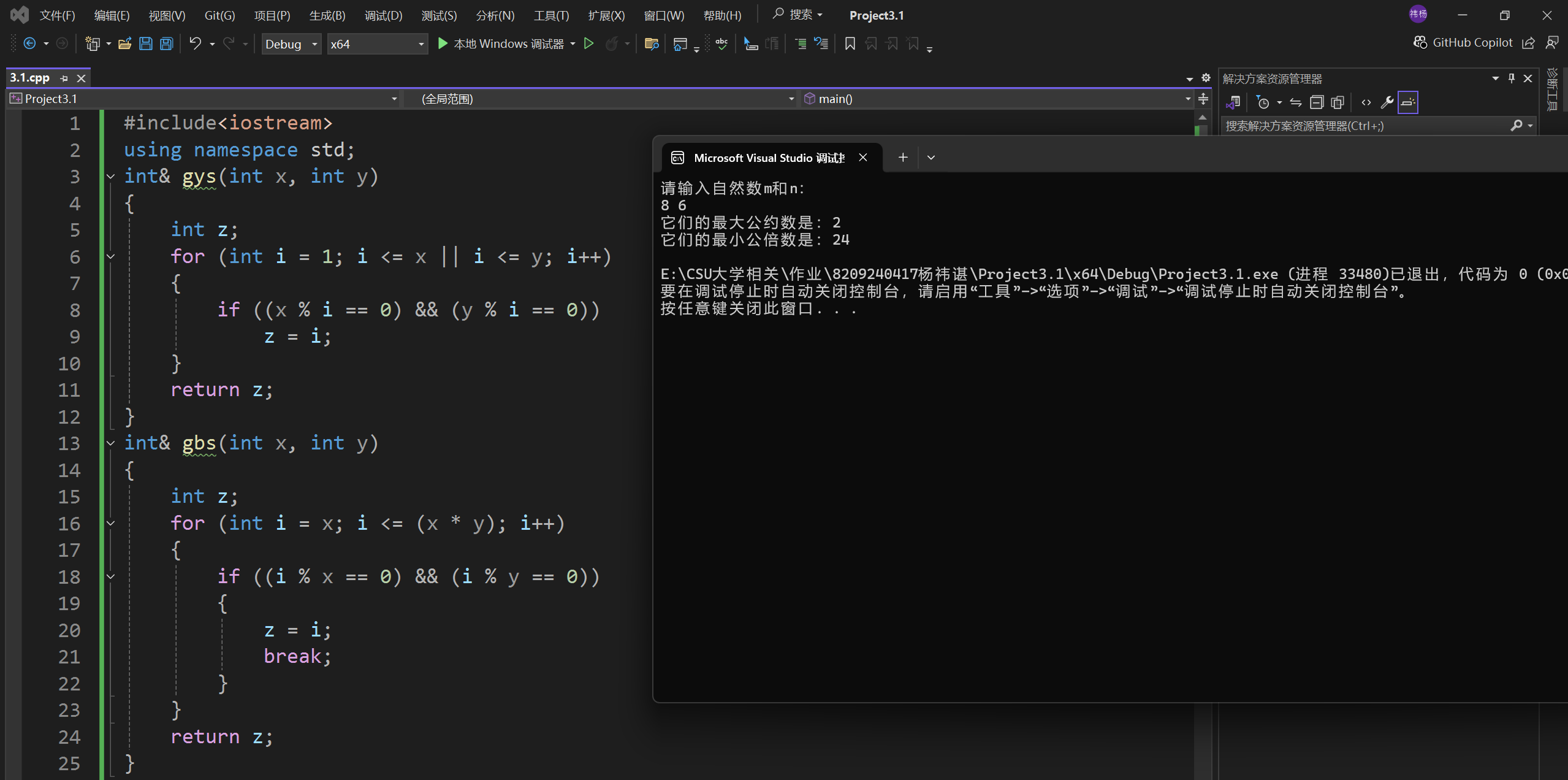
1. 本实验中函数中返回的值为什么与函数类型一致？

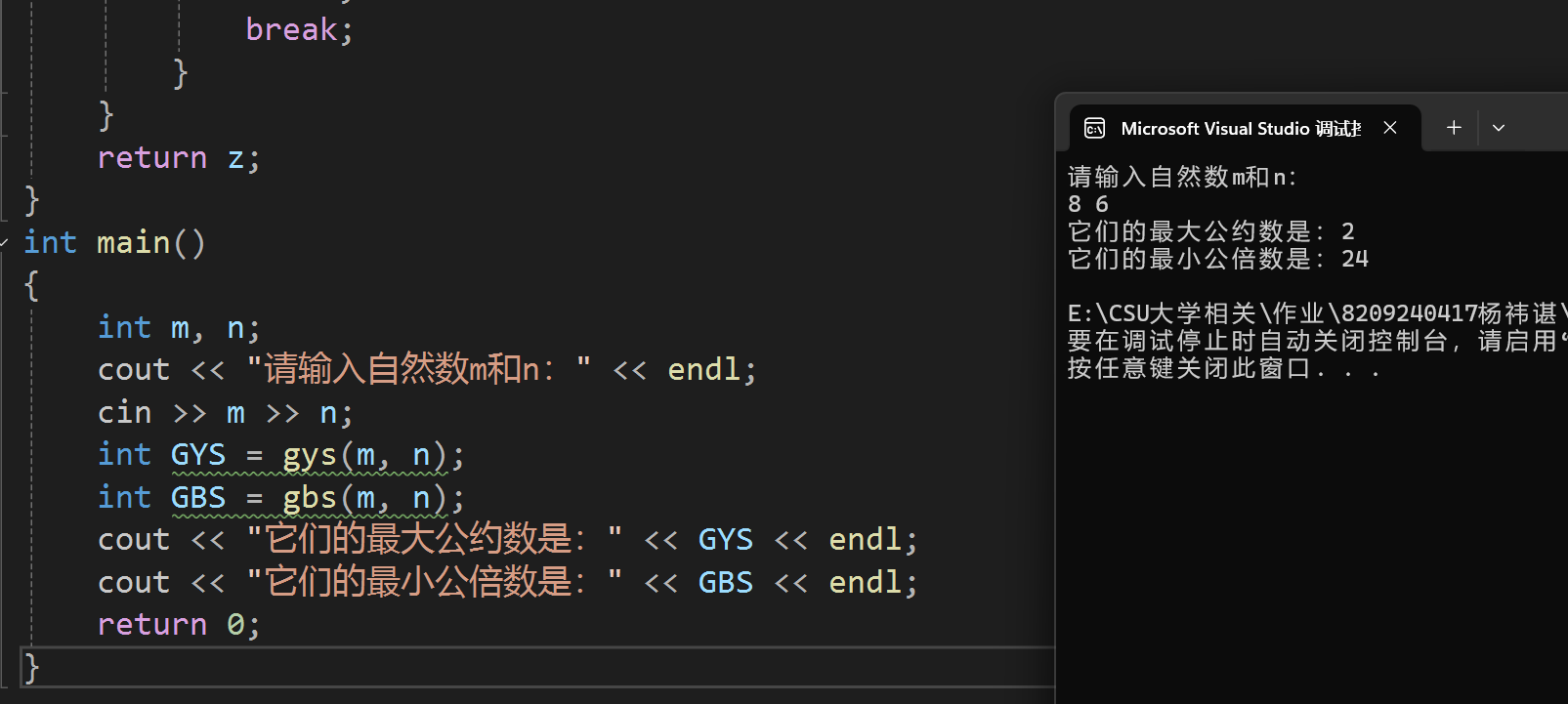
**返回类型为函数返回值的类型，可以是系统中任一基本数据类型或用户已定义的一种数据类型，它是函数执行过程中通过return语句要求的返回值的类型。函数类型决定了返回值的类型。函数原型中的返回类型必须与定义该函数是完全一致，函数声明时的返回类型和一致性主要是为了提供类型安全性和代码可读性。编译器会根据函数的返回 类型来决定如何存储和返回值，以保障程序的正确性和可维护性。**

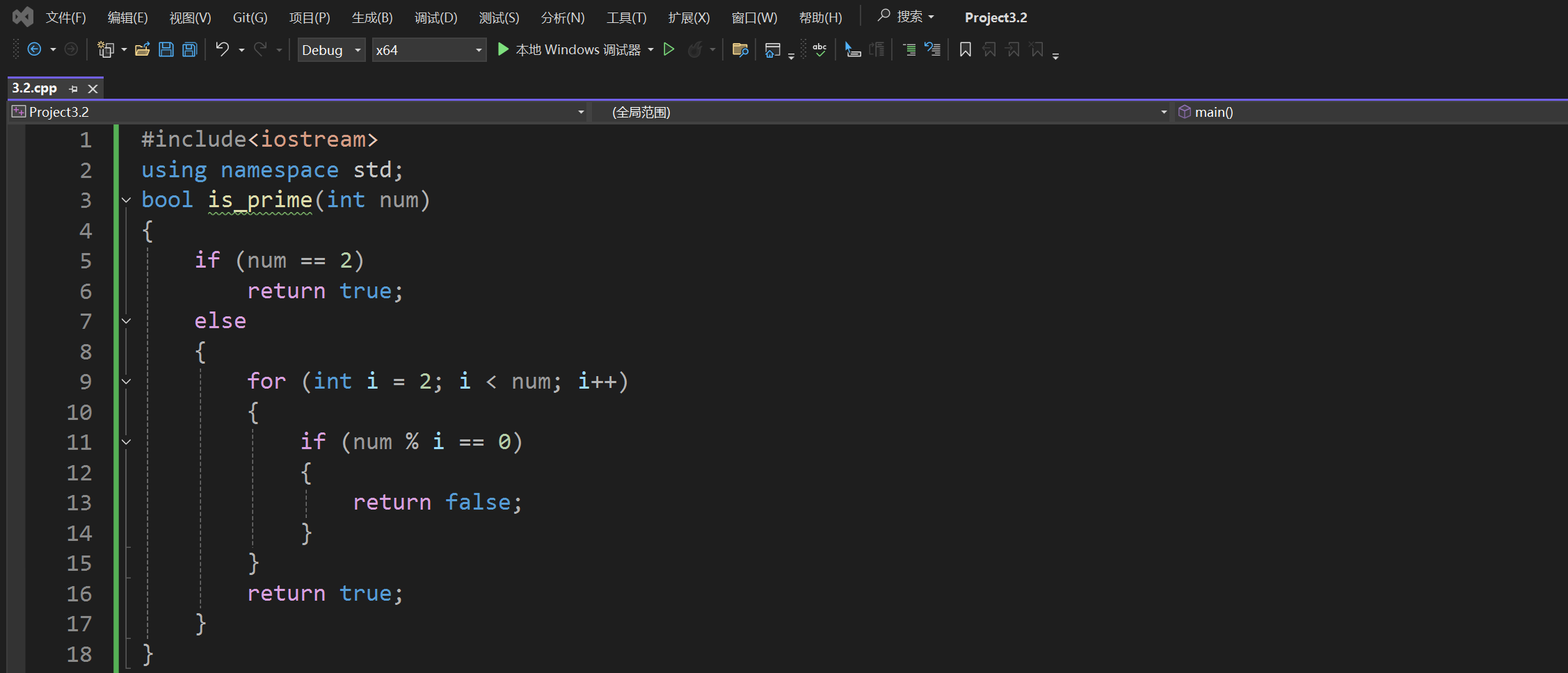
1. 本实验中主函数调用函数时采用的是何种传递方式？

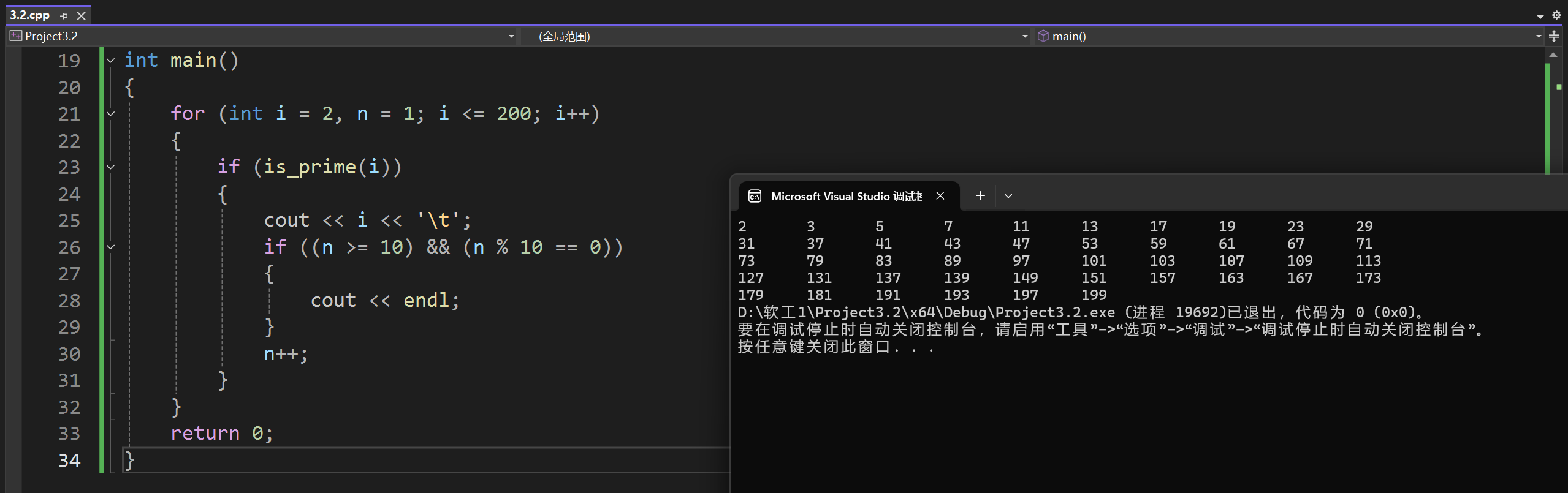
**按值传递**

1. **算法分析，程序结果**

**1**

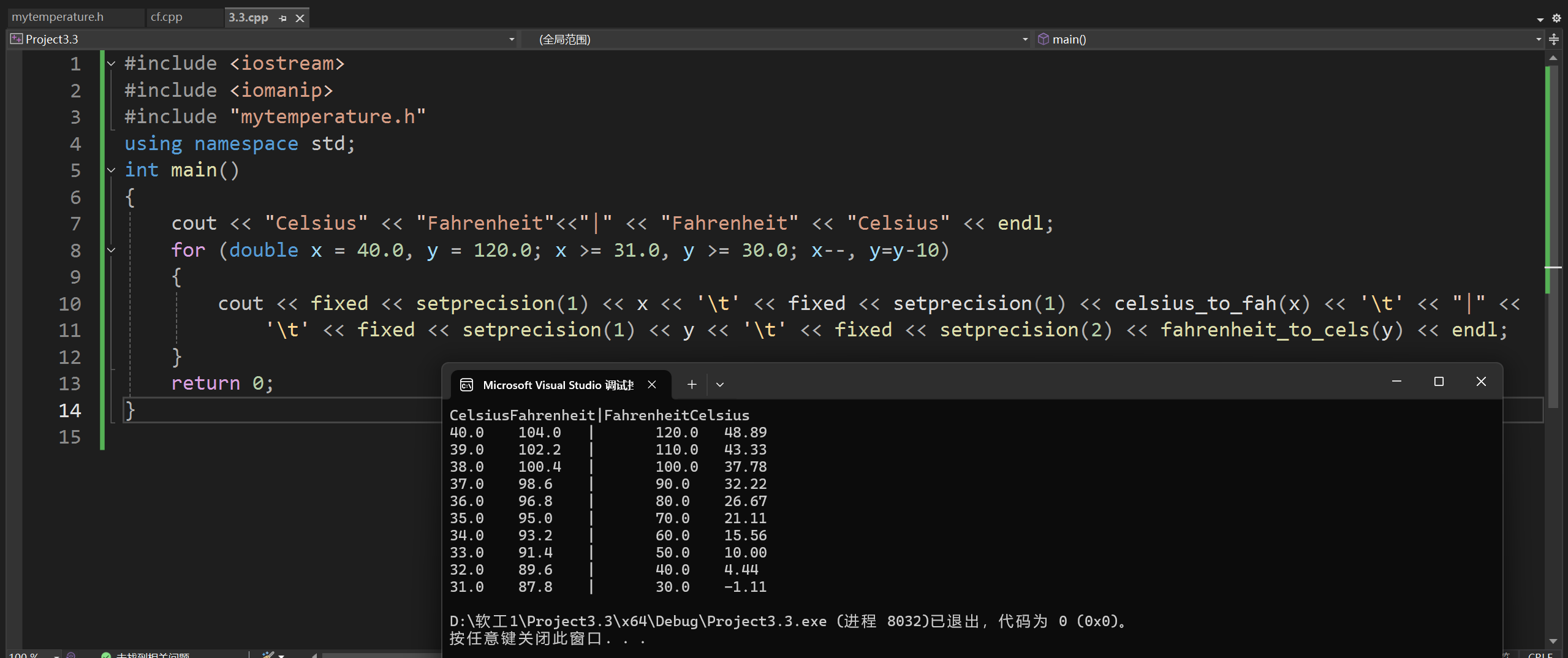


2

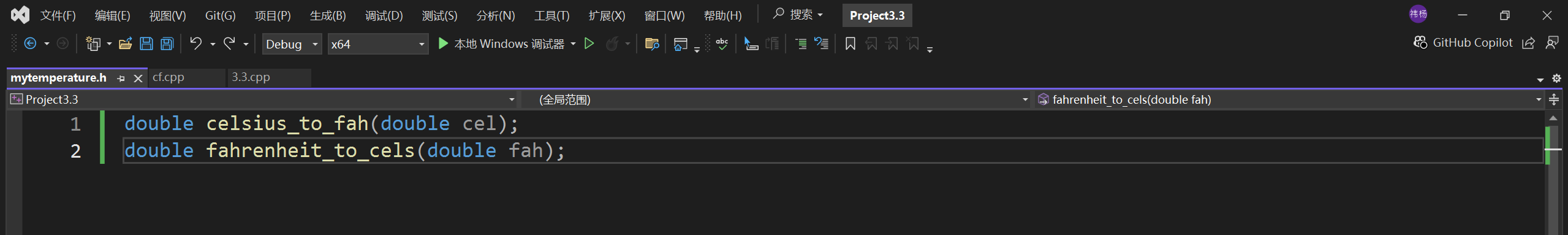


3

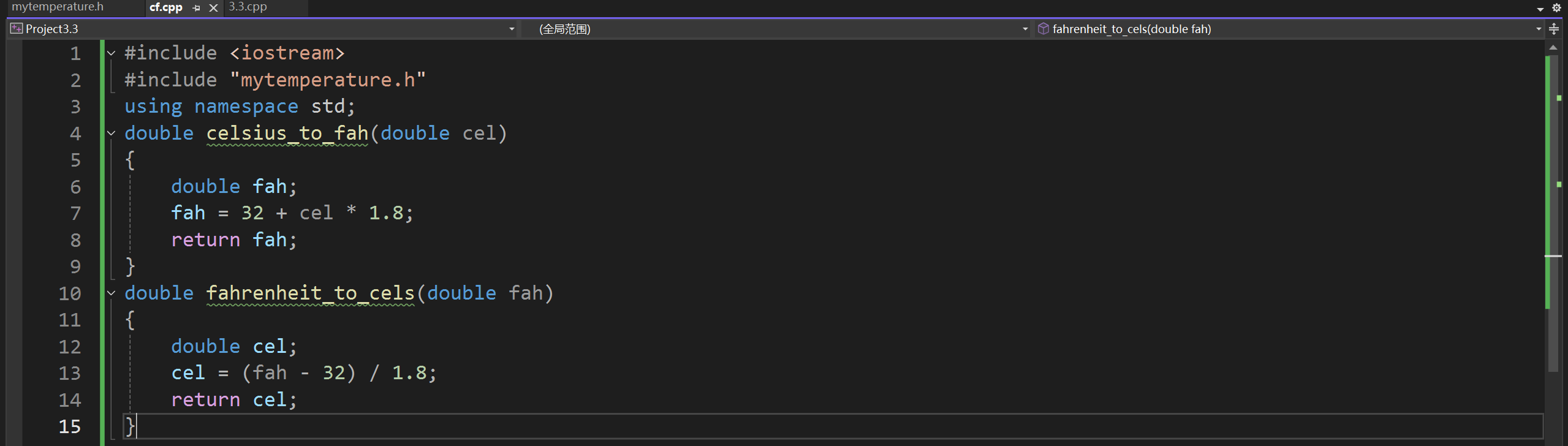
**主模块**



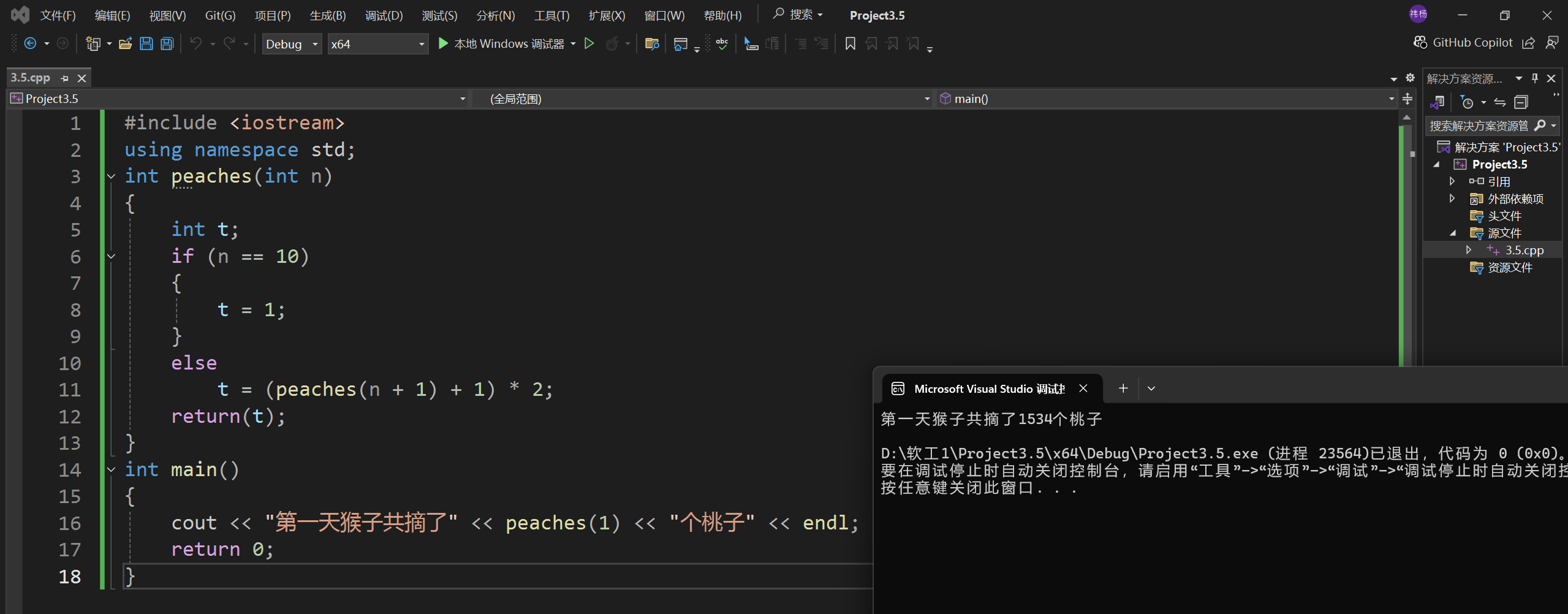
**头文件mytemperature.h只有函数声明**



**函数定义写在另一CPP文件cf.cpp**



**5**



1. **遇到的问题与解决方法**

在实验一中对最大公因数和最小公倍数的循环语句和if else语句逻辑不清，不知道怎么通过循环语句计算出最大公因数和最小公倍数，后通过学习书上的例题及与同学交流讨论写出了正确的代码。

在实验三中使用fixed和setprecision控制输出结果的小数位数，但因忘记打#include<iomanip>而报错，后因仔细检查并和书上相关例题比对发现错误。

在实验五中，第一次写代码时错将定义的t变量在else语句中写为t=(n/2)-1，导致逻辑错误，运行不正确。后经过仔细的检查且重新梳理思路意识到t=1是结果=初值，而else语句中的t应该为结果=题目意思的计算式，我一开始的想法与正确思路是颠倒的。在实验一中对最大公因数和最小公倍数的循环语句和if else语句逻辑不清，不知道怎么通过循环语句计算出最大公因数和最小公倍数，后通过学习书上的例题及与同学交流讨论写出了正确的代码。

**六、体会**

在C++编程过程中处处都是细节，需要我非常细心才能避免出错。通过这次实验，让我对函数的调用形式、函数调用的执行过程以及函数参数的按值传递有了更深刻、实质性的理解，没有书本上的抽象概念，比上课讲解理念概念时更能理解了。我也明白的动手实操的重要性。我也发现，我的程序总是会在循环或者嵌套循环或者判断逻辑上出现一些问题，这些问题都是因为思路不够清晰造成的。我要注重锻炼自己的逻辑思维能力，多去复习巩固琐碎的知识点。

**实验四 数组与指针**

【**实验目的**】

1、进一步加深对数组的理解，掌握数组的定义方法；

2、掌握数组的处理方法、数组作为函数参数的使用方法，以及搜索与排序的应用。

3、掌握指针的概念、指针变量定义格式以及指针的运算；

4、掌握指针与数组、函数的关系；

5、理解内存动态分配的含义、熟练掌握内存动态分配方法；

6、掌握递归函数的定义方法。

【实验内容与步骤】

**（一）数组**

1、打印不同的数：

编写一个程序，读入10个数，输出其中不同的数（即如果一个数出现多次，只打印一次）。

提示：读入的数如果是一个新的值，则将其存入一个数组。否则，将其丢弃。输入完毕后，数组中保存的就是不同的数。

下面是一个运行样例：

Enter

Enter ten numbers: 1 2 3 2 1 6 3 4 5 2

The distinct numbers are: 1 2 3 6 4 5

2、起泡排序：

利用起泡排序算法编写一个排序函数。起泡排序算法分若干趟对数组进行处理。每趟处理中，对相邻元素进行比较。若为降序，则交换；否则，保持原顺序。此技术被称为起泡排序（bubble sort）或下沉排序（sinking sort），因为较小的值逐渐地“冒泡”到上部，而较大值逐渐下沉到底部。

算法可描述如下：

bool changed = true;

do

{

changed = false;

for (int j = 0; j < listSize – 1; j++)

if (list[j] > list[j+1])

{

swap list[j] with list[j+1];

changed = true;

}

} while (changed);

很明显，循环结束后，列表变为升序。容易证明do循环最多执行listSize – 1次。

编写测试程序，读入一个含有10个双精度数字的数组，调用函数并显示排列后的数字。

3、游戏：存物柜问题：

一个学校有100个存物柜，100个学生。开学第一天所有存物柜都是关闭的。第一个学生（记为S1）来到学校后，打开所有的存物柜。第二个学生S2，从第二个存物柜（记为L2）开始，每隔两个存物柜，将它们关闭。第三个学生S3从第三个存物柜L3开始，每隔三个，将它们的状态改变（开着的关上，关着的打开）。学生S4，从L4开始，每隔四个改变它们的状态。学生S5，从L5开始，每隔五个改变状态。依此类推，直至学生S100改变L100的状态。

当所有学生完成这个过程，那些存物柜是开着的？编写一个程序求解此问题，显示所有开着的柜子号码，号码之间用一个空格隔开。

提示：使用一个100个布尔型元素的数组，每个元素代表存物柜是开（true）或关（false）。最初所有的储物柜都是关闭的。

4、合并两个排列好的数组：

编写如下函数，合并两个排列好的数组，形成一个新的排列好的数组。

void merge(const int list1[], int size1, const int list2[], int size2, int list3[])

使用size1+size2次比较实现函数。编写测试程序，提示用户输入两个排列好的数组，并显示合并以后的数组。下面是一个运行样例。注意，输入数据的第一个数字是数组的元素数，而不是数组的一部分。假定数组大小不超过80。

Enter

Enter

Enter list1: 5 1 5 16 61 111

Enter list1: 4 2 4 5 6

The merged list is 1 2 4 5 5 6 16 61 111

5、检验子串：

编写如下函数，检验C字符串s1是否是C字符串s2的子串。如果匹配，返回s1在s2中的下标，否则返回–1。

int indexOf(const char s1[], const char s2[])

编写测试程序，读入两个C字符串，检验C字符串s1是否是C字符串s2的子串。下面是程序的运行样例：

Enter

Enter

Enter

Enter the first string: welcome

Enter the second string: We welcome you!

indexOf(“welcome”, “We welcome you!”) is 3

Enter

Enter the first string: welcome

Enter the second string: We invite you!

indexOf(“welcome”, “We invite you!”) is –1

6、字符串中每个字母出现的次数：

请使用如下函数头编写函数，数出字符串中每个字母出现的次数。

void count(const char s[], int counts[])

counts是一个有26个元素的整数数组。const[0]，const[1]，…，const[25]分别记录a，b，…，z出现的次数。字母不分大小写，例如字母A和字母a都被看作a。

编写测试程序，读入字符串并调用count函数，显示非零的次数。下面是程序的一个运行样例：

Enter

Enter a string: Welcome to New York!

c: 1 times

e: 3 times

k: 1 times

l: 1 times

m: 1 times

n: 1 times

o: 3 times

r: 1 times

t: 1 times

w: 2 times

y: 1 times

**（二）指针**

1、上机验证下列程序的运行结果（有错误的话自己补充完善）

(1) void main()

｛

int i,j,\*pi,\*pj; //此处的\*表示定义指针变量，而非间接运算符

pi=&i;

pj=&j;

i=5;j=7;

cout<<i<<’\t’<<j<<’\t’<<pi<<’\t’<<pj;

cout<<&i<<’\t’<<\*&i<<’\t’<<&j<<’\t’<<\*&j;

}

运行结果：

上述结果中，pi与&i,pj与&j是地址值，随编译程序而变化，不确定。

(2) int main() //C语言程序，要了解

{

int a[]={1,2,3};

int \*p,i;

p=a; //将数组a首地址送给p

for (i=0;i<3;i++)

printf("%d,%d,%d,%d\n",a[i],p[i],\*(p+i),\*(a+i)); //与cout功能差不多

}

运行结果：

1,1,1,1

2,2,2,2

3,3,3,3

通过这两道题目，希望学生掌握数组元素与指向数组的指针的不同。

a[i]表示数组中下标为i的元素。

a[i]←p[i]←\*(p+i)←\*(a+i)

a是数组名，表示数组首地址，(p+i)表示数组中第i个元素的地址，\*(p+i) 相当于a[i]。

(3)通过如下的问题理解递归函数的定义与调用（递归未讲，可以后做）

//#include “stdio.h”

void f(char \*st,int i)

{

st[i]=’\0’;

cout<<st; // printf(“%s\n”,st);

if (i>1) f(st,i-1);

}

void main()

{

char st[]=”abcd”;

f(st,4);

}

补充完整，运行时输出为\_\_\_\_\_\_\_\_

(4)下面程序的主函数中能保证p[0]输出1，p[1]输出2吗？如何修改以保证之（提示：在函数f中使用new生成动态数组；在main中用delete释放。）

#include<iostream>

using namespace std;

int \*f()

{

int list[]={1,2,3,4};

return list;

}

void main()

{

int \*p=f();

cout<<p[0]<<endl;

cout<<p[1]<<endl;

}

2、程序设计

(1)编写函数检查字符串s1是否为字符串s2的子串，若是，返回第一次匹配的下标，否则返回-1。在主程序中输入字符串s1与s2，调用函数实现。

函数原型：int indexof(const char \*s1,const char \*s2);

(2)编写一个函数将以字符串形式表示的一个16进制数转换为10进制数，并在主函数中测试。函数原型 int parseHex(const char \*const hexString);

如：调用函数 parseHex(“A5”);返回165

1. 主程序中建立一动态数组（使用new），数组元素及元素个数由键盘输入，动态调试观察指针及指针指向的内容；设计一个函数对数组由小到大排序；主程序中用指针方式输出数组元素；最后释放数组内存（delete）。

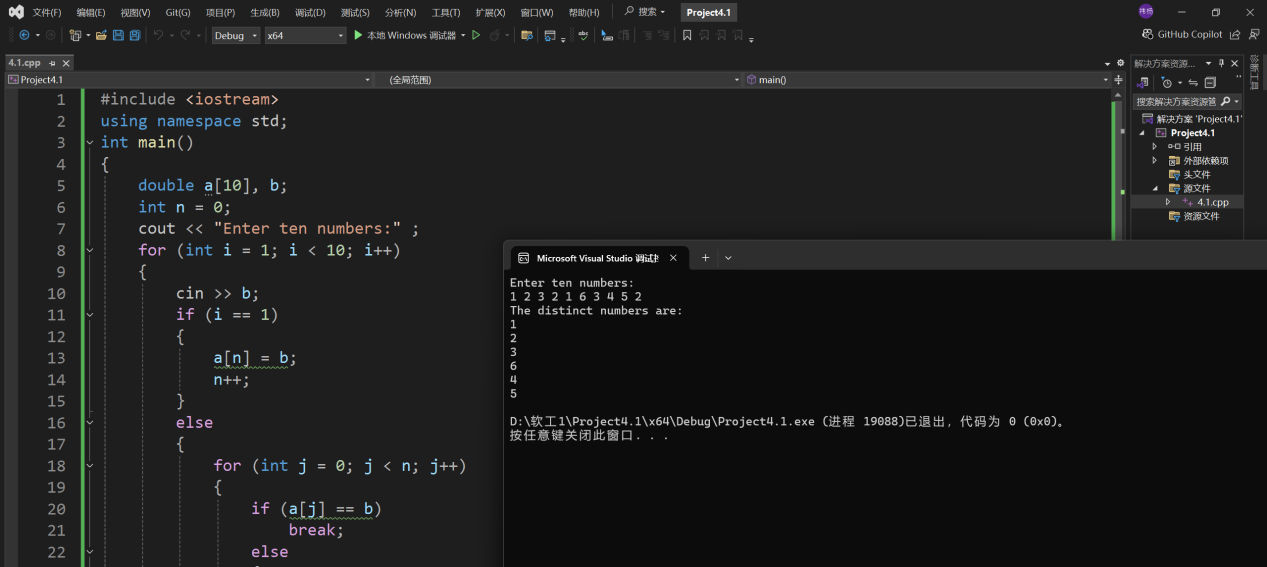
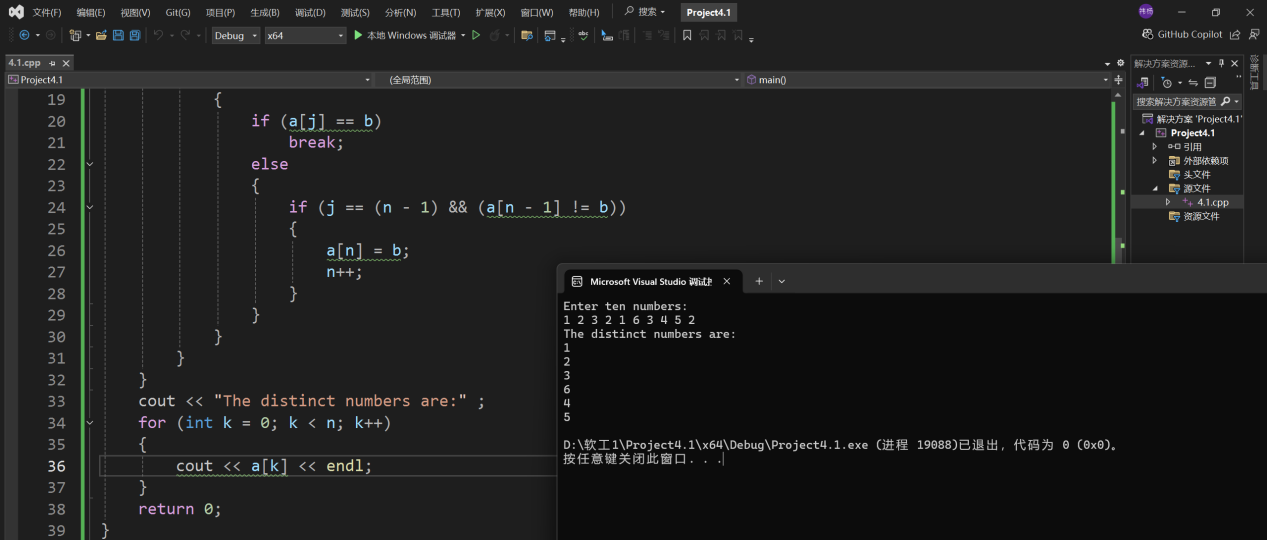
【完成实验报告】

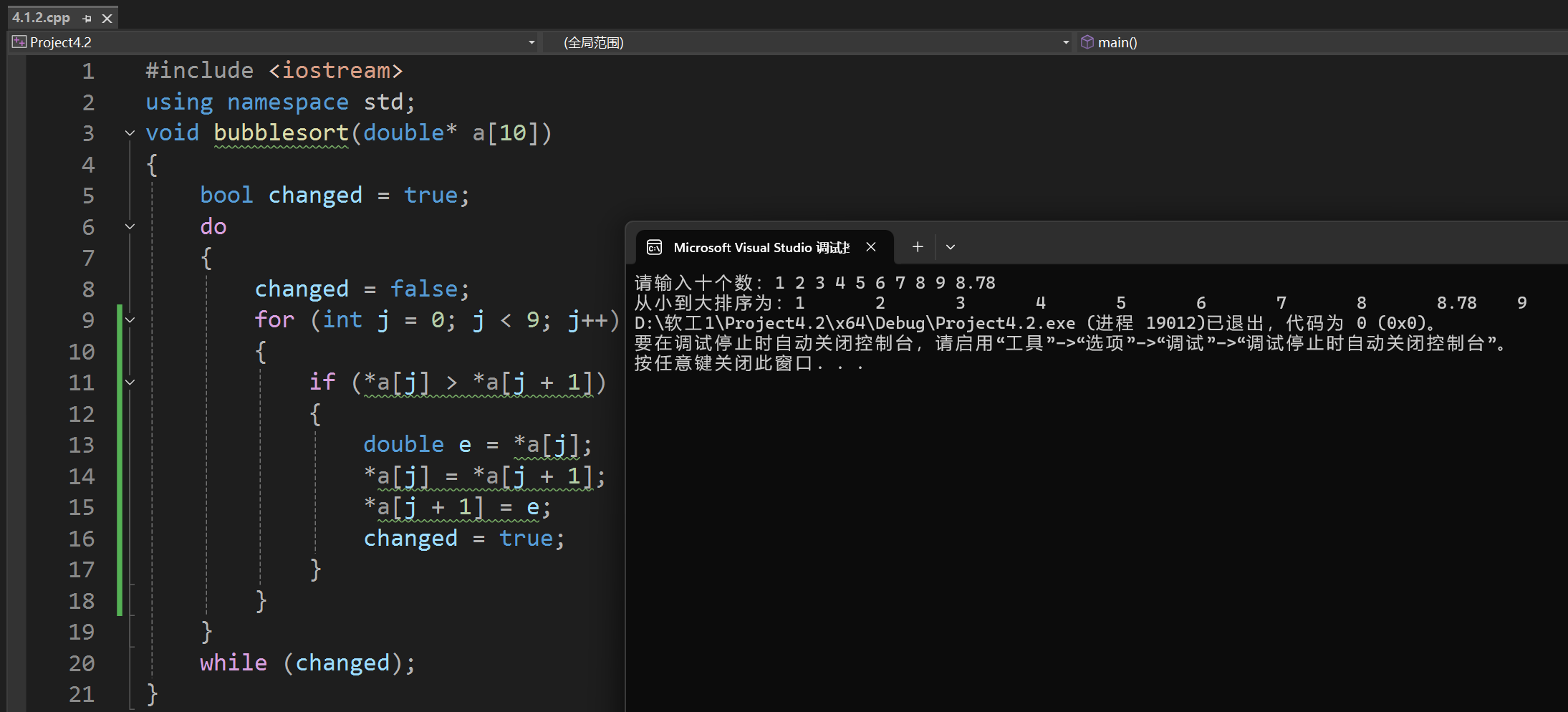
**实验报告只要求写程序设计部分**

1. **算法分析，程序结果**

**（一）数组**

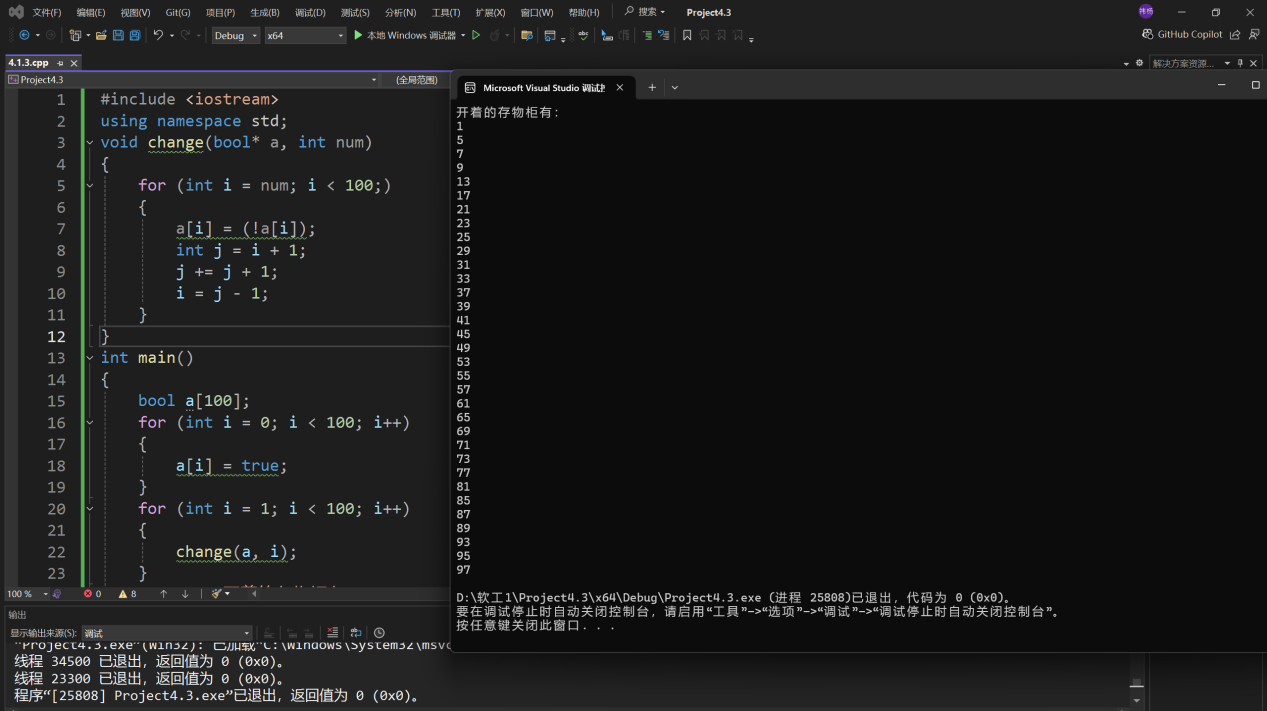
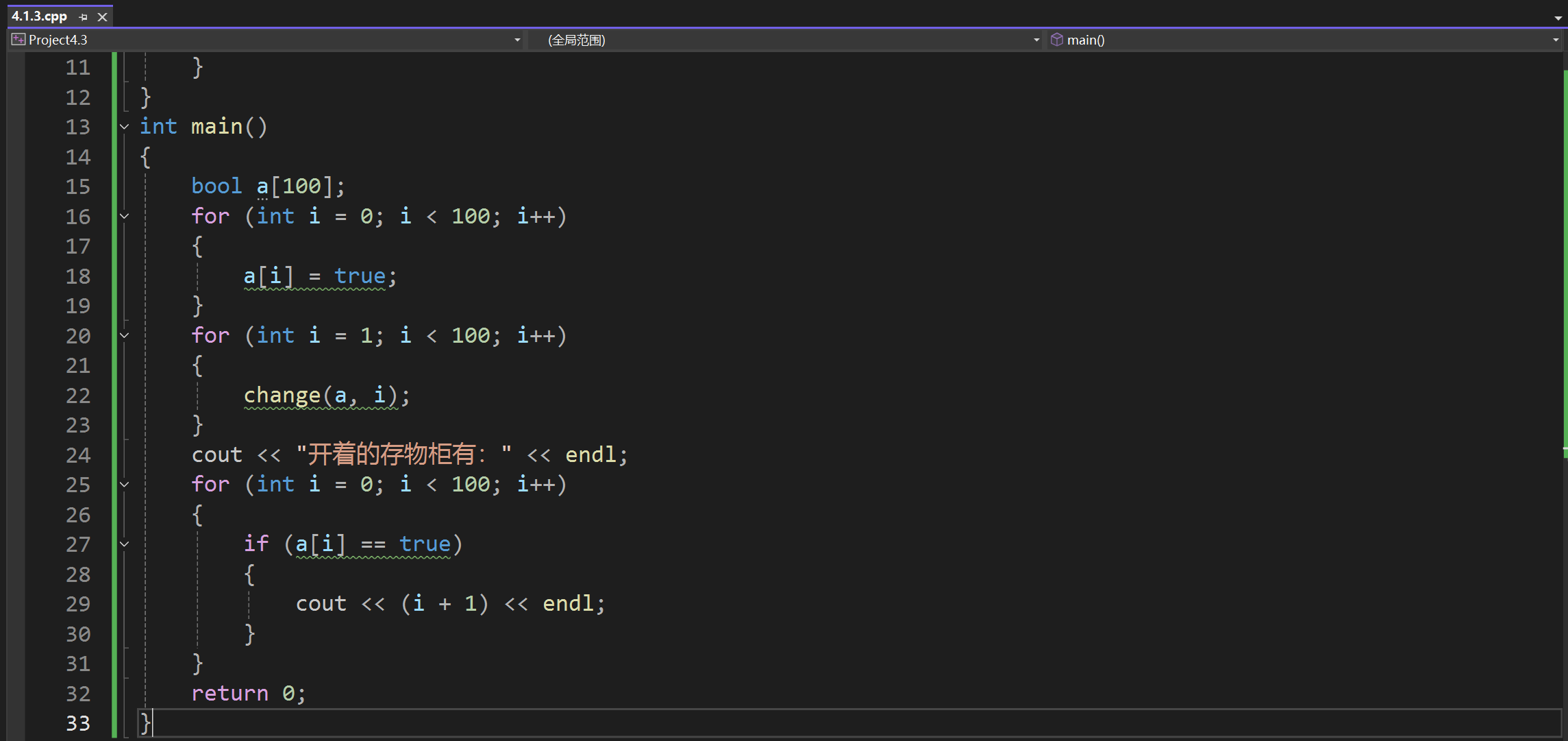
**1**

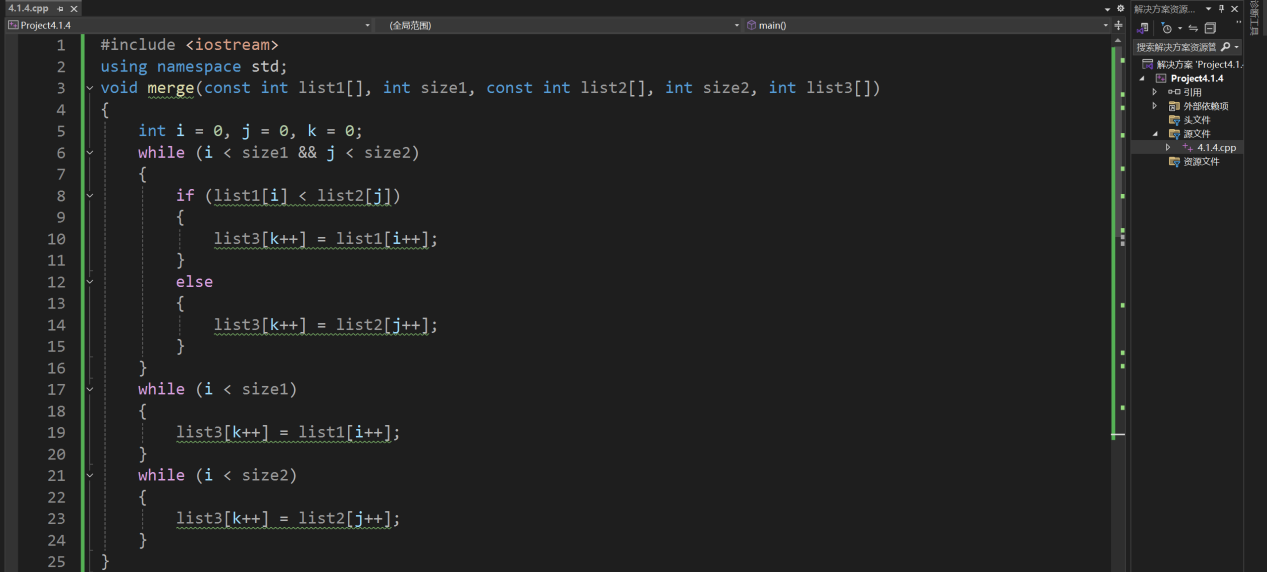


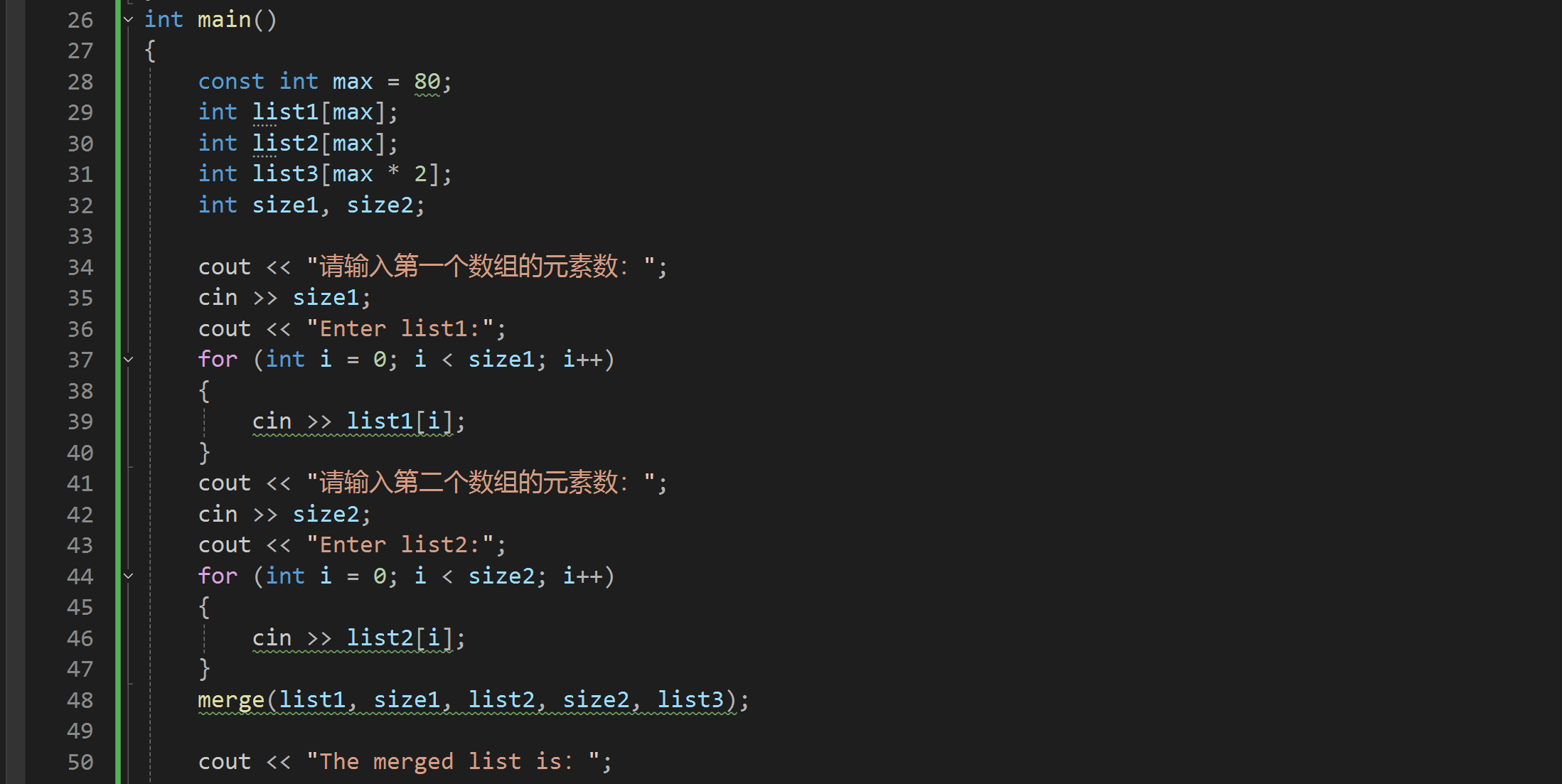
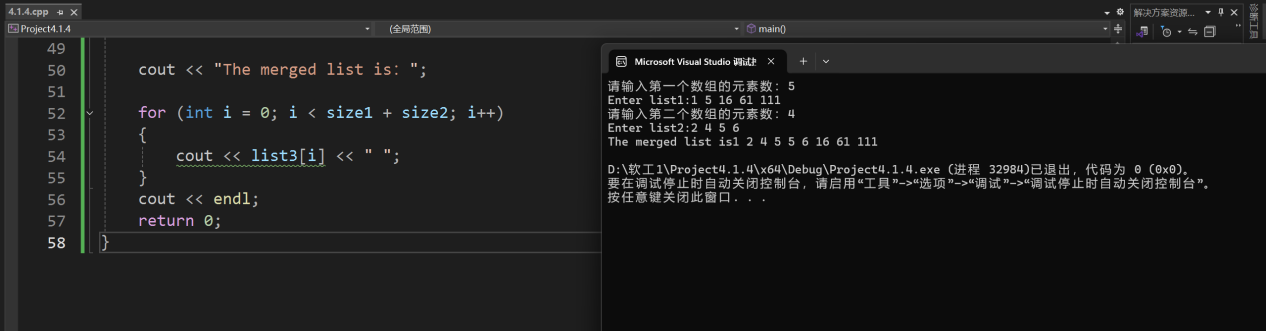
**2**



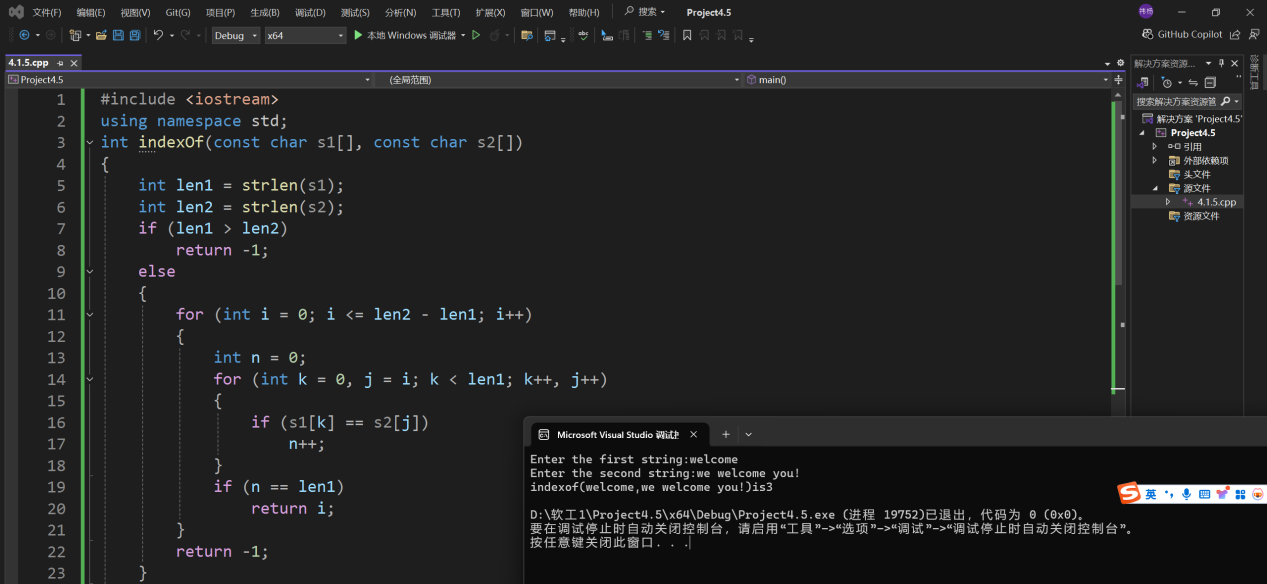
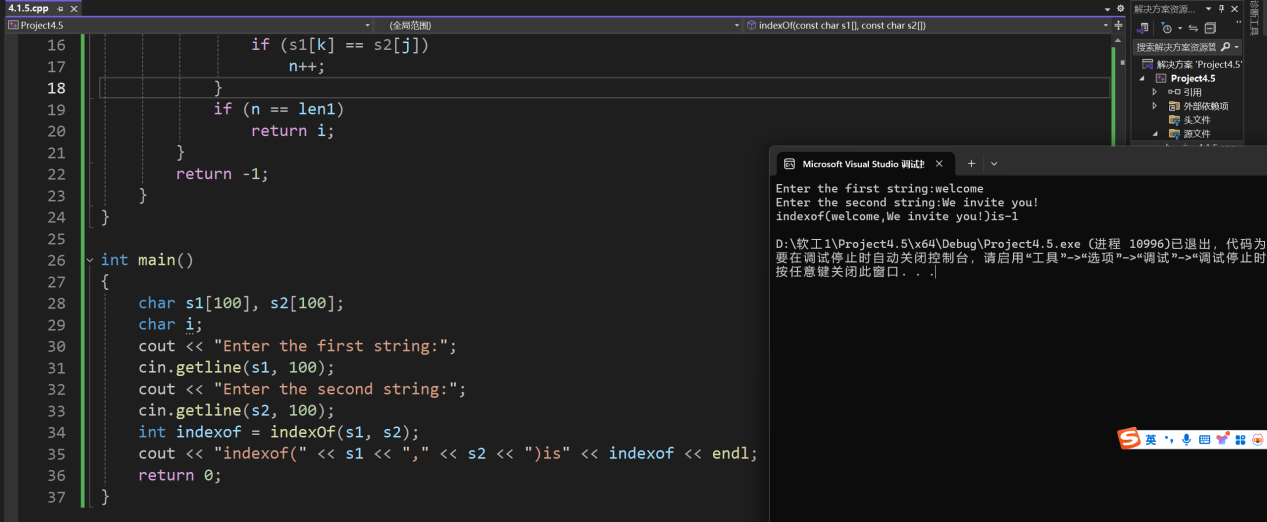
**3**

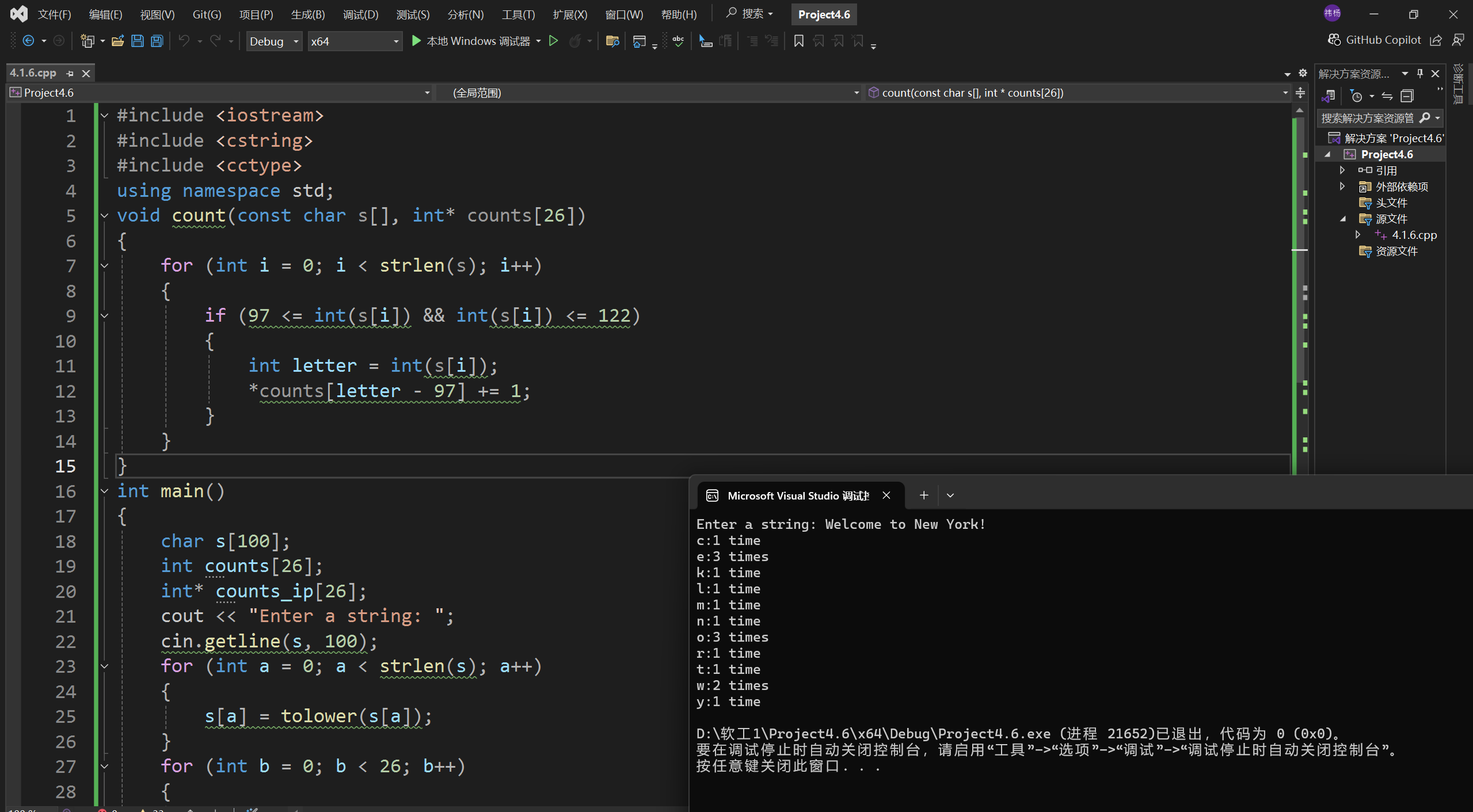


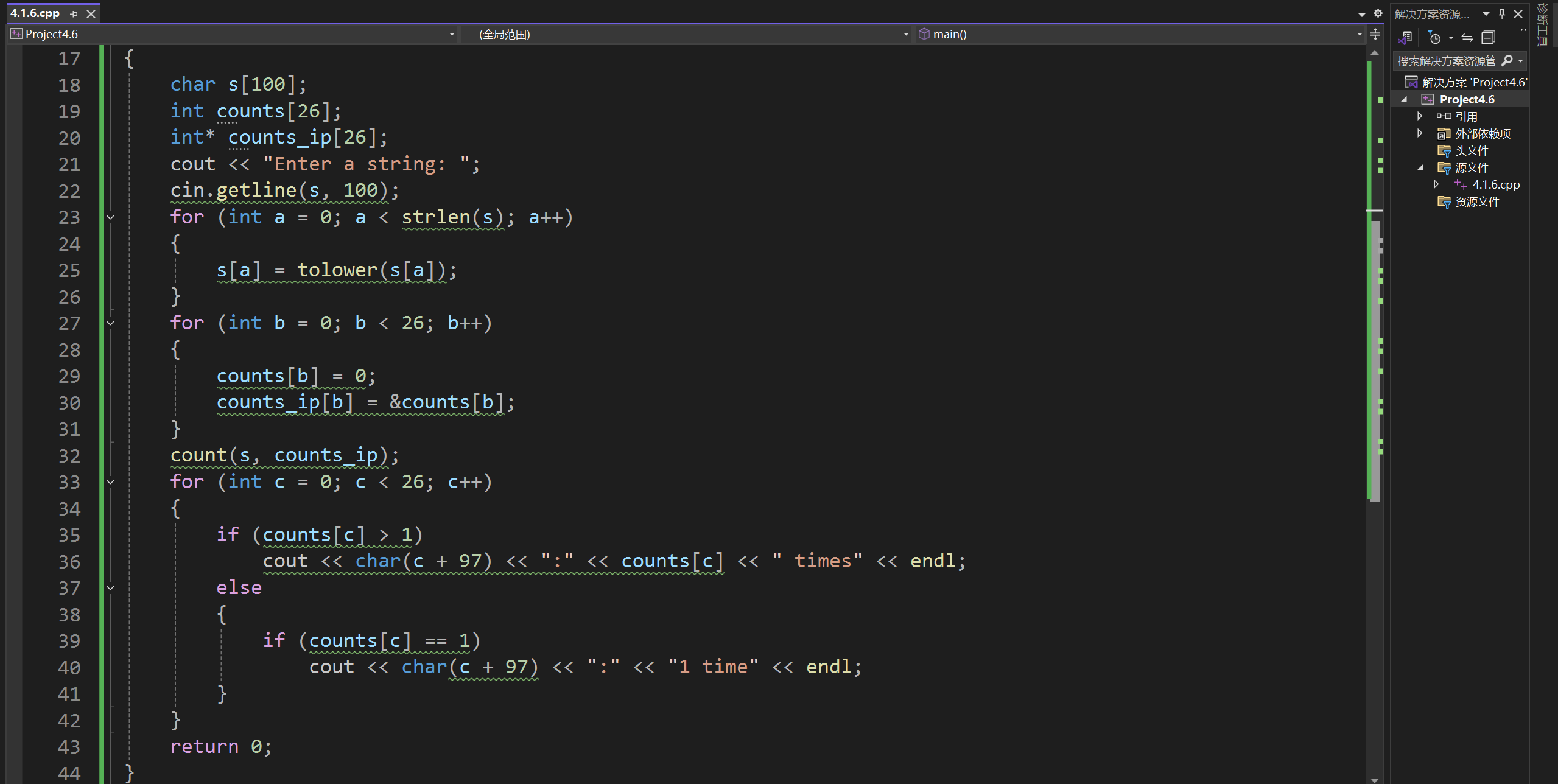
**4**



**5**

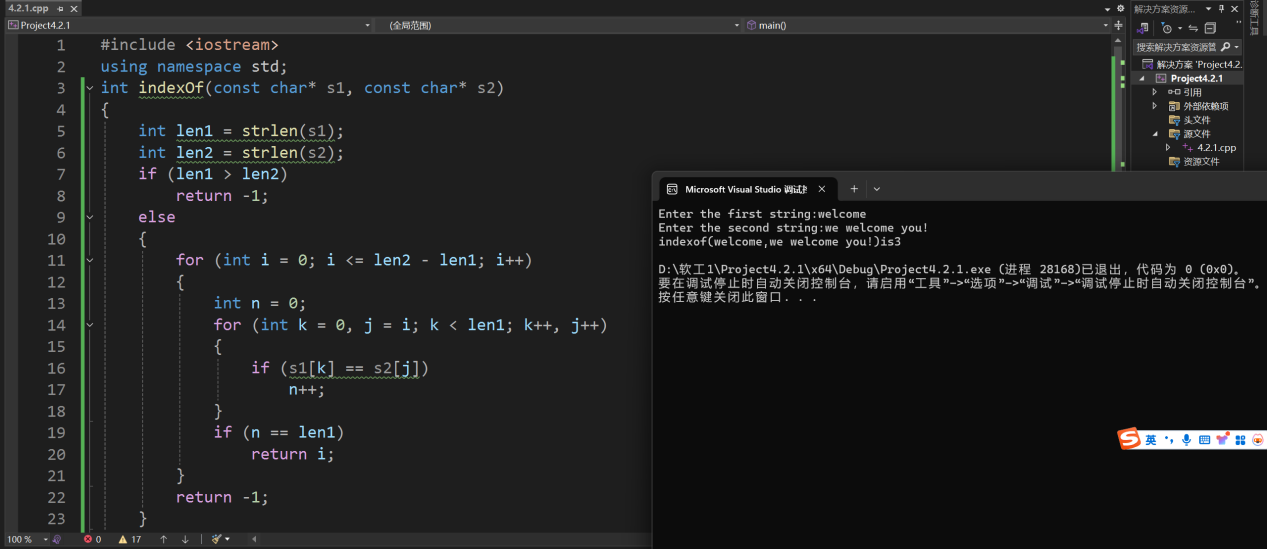


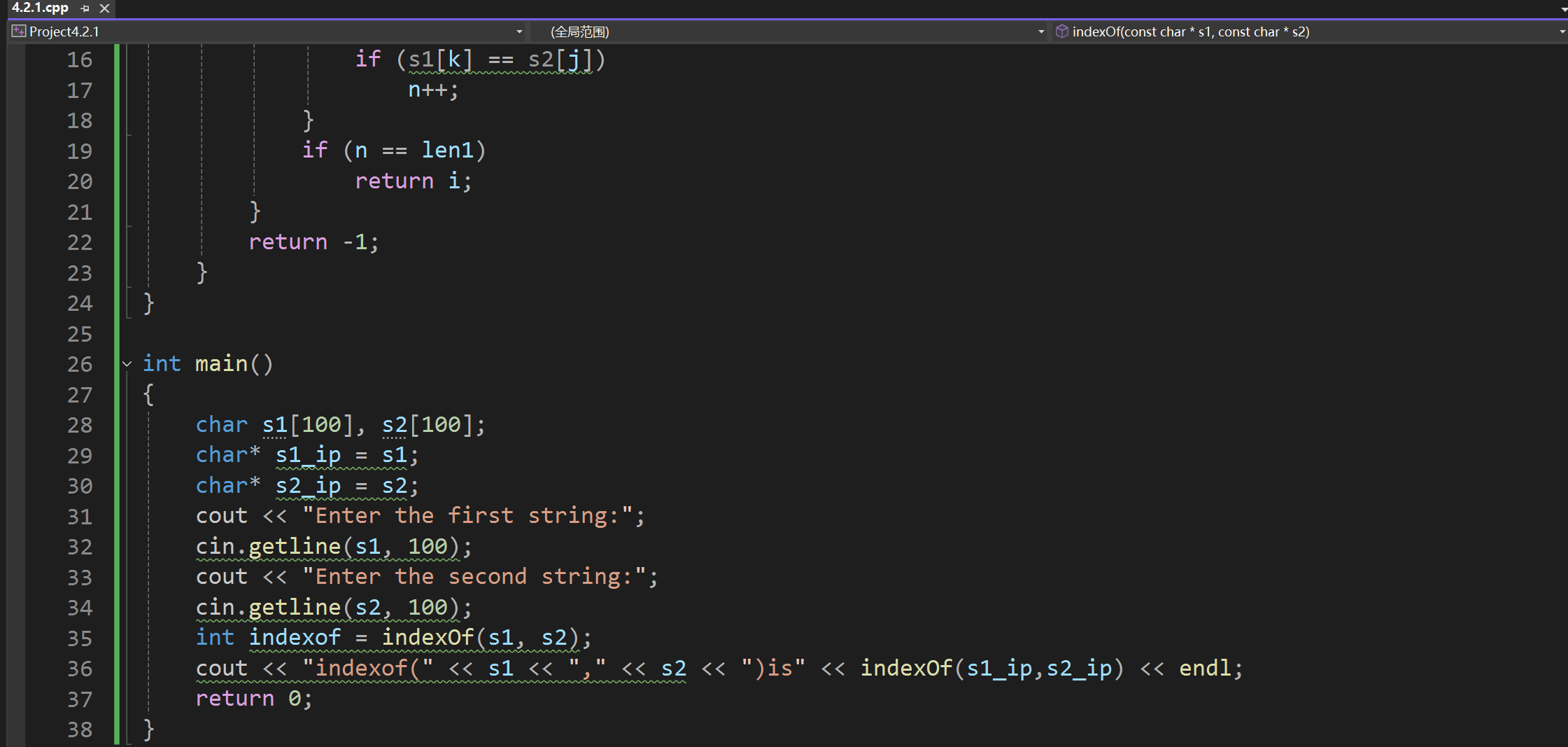
**6**



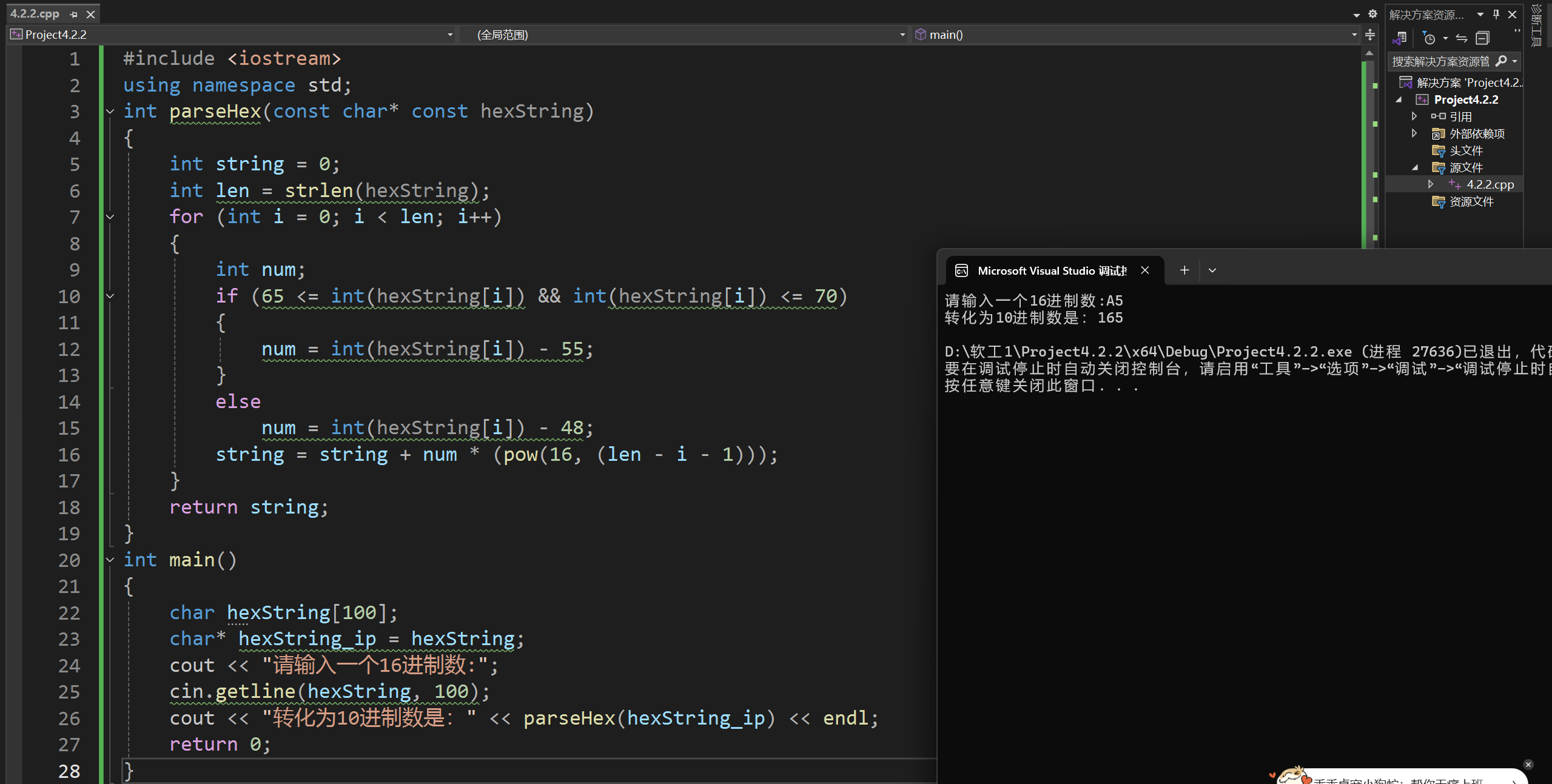
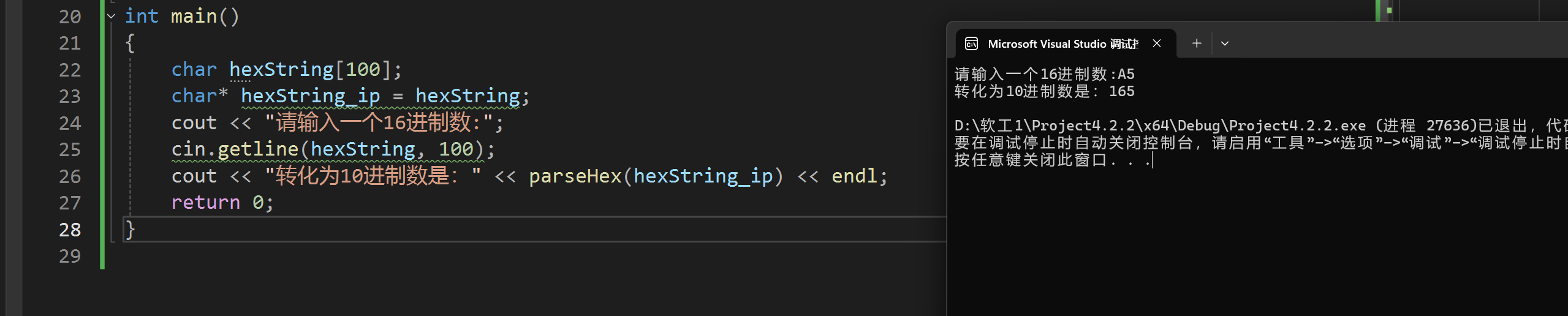
**（二）指针 程序设计**

**1**

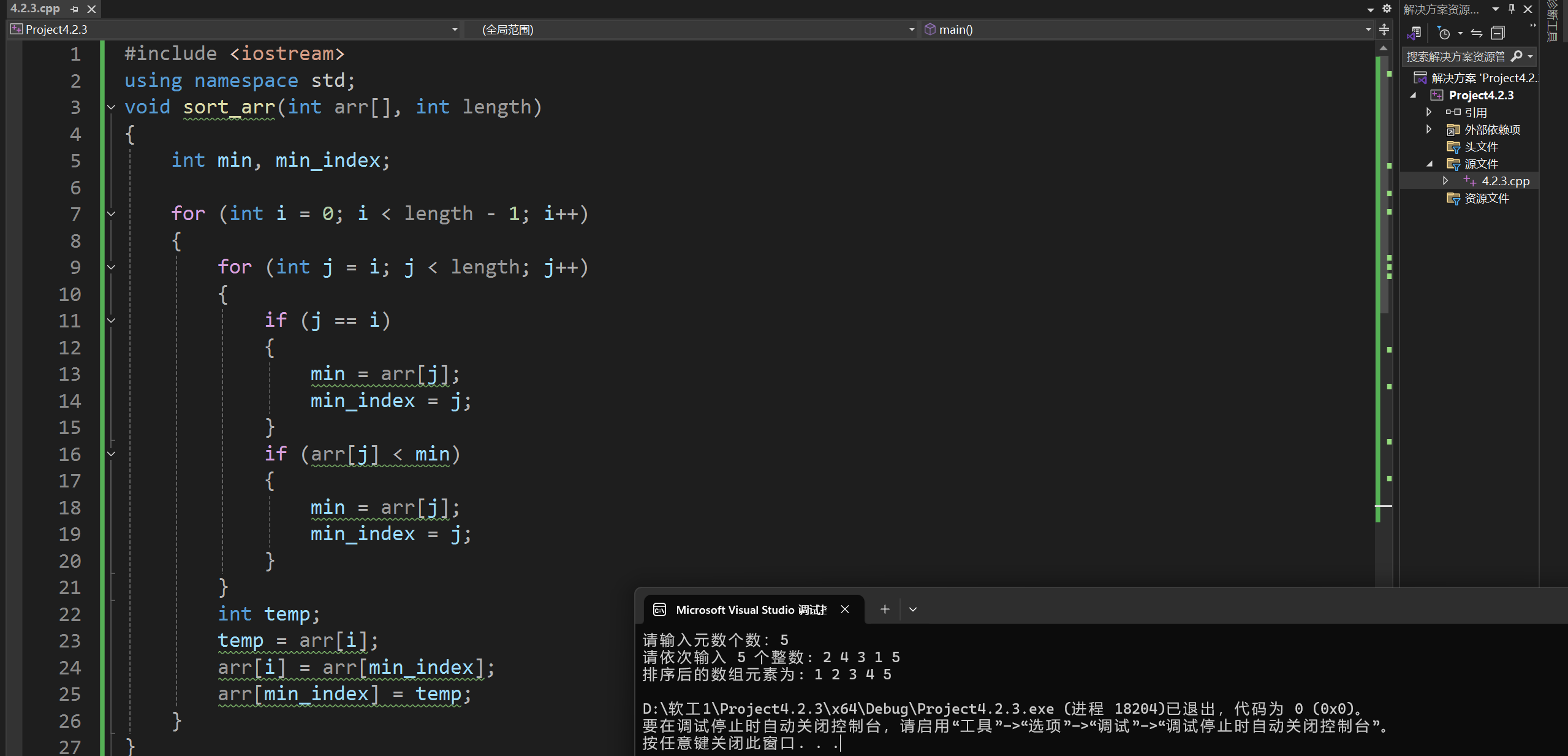




**2**

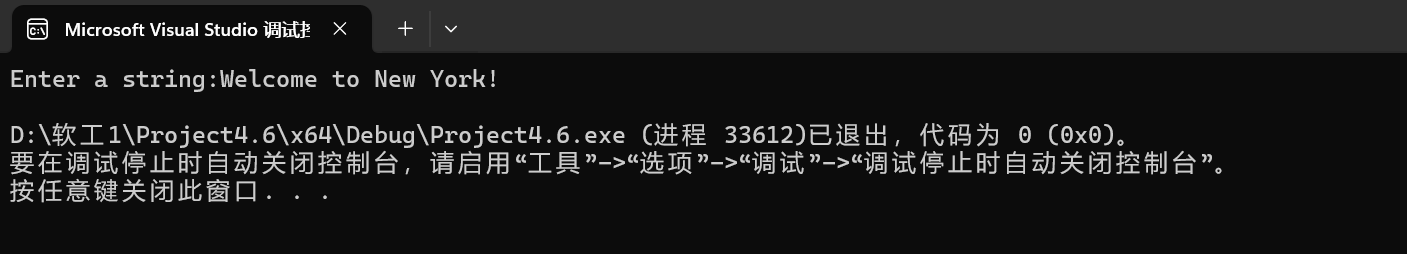


**3**





**四、遇到的问题与解决方法**

数组的第六题，代码都是正确的，但不能正常输出，如下图

通过查询AI，原因可能是字符编码及判断问题，在代码中使用了 isalpha 函数来判断字符是否为字母，这个函数依据的是 ASCII 编码标准（在常见的实现中），对于一些特殊字符或者非标准 ASCII 编码环境下的输入可能会出现判断失误。不过对于 "welcome to new york" 这个常规的字符串来说，通常不是这个原因导致无输出。所以问题大概率出在 main 函数最后的输出循环部分：在这个循环中，只有当 counts[i] 的值大于 0 时才会输出对应字母及其出现次数。但对于输入 "welcome" 字符串，它里面包含的字母 'w'、'e'、'l'、'c'、'o'、'm' 对应的 counts 数组索引分别是 22、4、11、2、14、12 等，可能由于在 count 函数中统计出现了错误，导致这些索引位置对应的 counts 数组元素值没有正确增加，依然保持初始值 0，所以循环判断条件 counts[i] > 0 始终不满足，也就不会有任何输出结果呈现出来。从代码逻辑上看本身没有明显错误，但是有一种可能是在调试环境中出现了异常情况影响了函数执行效果。比如编译器的优化设置可能导致代码执行顺序不符合预期（虽然在这里不太常见），或者调试时存在一些干扰因素（例如断点设置影响了变量值的正常更新等，不过这种情况也比较少见）。最后通过检查编译器及调试环境设置，重新新建项目打了一遍代码则结果能正常输出，排查了问题。在好几个实验中，经常遇到编译器没有报错，说明没有语法问题，但输出结果不正确或者 无法输出。通过不断尝试输出中间变量以及断点调试或在CSDN或AI搜索查询，解决问题

**五、体会**

数组和指针内容繁多且复杂，在灵活运用下可以实现很多功能。但是如果想要实现相对比较复杂、更多的功能，就要掌握并且理解更多的有关于数组的知识。例如在了解了动态数组的原理以及定义方式后，可以很快的掌握动态数组的定义以及使用，在使用了新的动态数组的知识后，运算过程、代码编写的方式都变得更加方便。但是，在使用动态数组时，也要记得在数组使用完毕后通过delete将数组数据删除。调用指针可以为实现更多的程序功能，为函数提供了更多的可操作性，提高了代码的可读性，在接下来的一段时间中，要巩固指针相关的知识，将指针掌握，以为自己的程序提供更多的可能性。在解决问题时，可以通过目前已学的知识对其进行解决，也可以额外学习一种新的更加高效的解决问题的方法，以拓宽自己的知识面，获得更多解决问题的能力，最终提高自己的综合编程能力以及学习能力。在程序的编写过程中，极有可能会因为一个很小的错误（比如一个符号）导致整个程

序得不到理想的输出，而这种错误往往很难发现，这需要我锻炼思维的严谨性，养成耐心细

心的习惯。