**计算机程序设计基础（C++)**

**实验报告**

专业班级：

学 号：

姓 名：

**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，求公约数由函数实现。

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？

答：因为函数名也是一个变量，函数的返回值最终是要赋值给函数名所代表的变量，所以二者类型应一致。

1. 本实验中主函数调用函数时采用的是何种传递方式？

答：地址传递。

1. **算法分析，程序结果**

**1.程序：**

#include<iostream>

using namespace std;

int a, b;

int min(int a, int b)

{

if (a < b) {

int t;

t = a, a = b, b = t;

}

int min;

for (min = a; min % a != 0 || min % b != 0; min++) {

}

return min;

}

int max(int a, int b)

{

if (a < b) {

int t;

t = a, a = b, b = t;

}

int max;

for (max = b; a % max != 0 || b % max != 0; max--) {

}

return max;

}int main() {

int a, b;

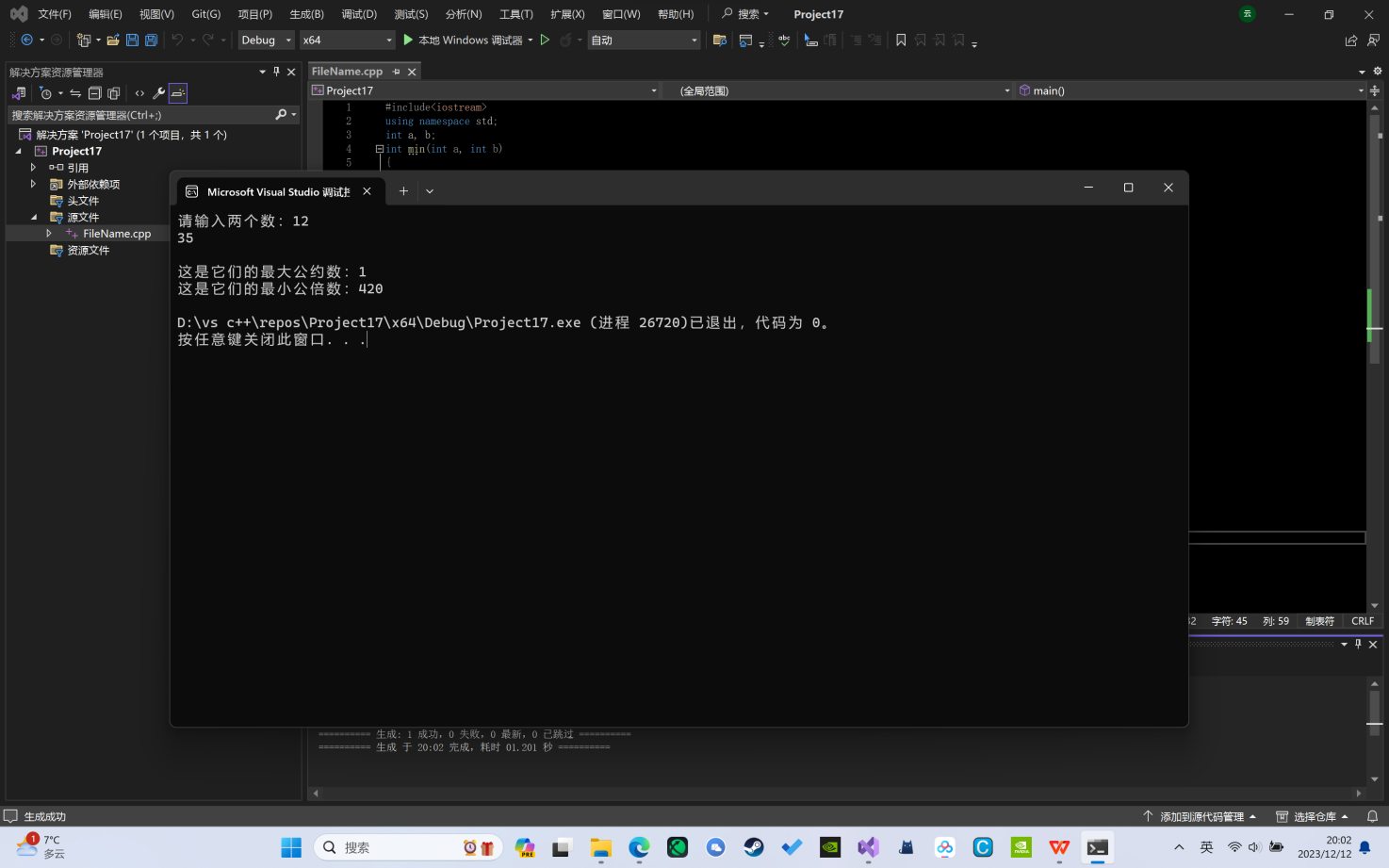
cout << "请输入两个数：";

cin >> a >> b;

cout << "\n这是它们的最大公约数：" << max(a, b) << endl;

cout << "这是它们的最小公倍数：" << min(a, b) << endl;

}

****

**2.程序：**

#include <iostream>

#include <stdio.h>

using namespace std;

bool is\_prime(int num) {

int a;

for (a = num - 1; num % a != 0 ; a--) {

}

if (a == 1) {

return 1;

}

else {

return 0;

}

}

int main() {

int su[200];

int n=2, m=0;

while (m < 200) {

if (is\_prime(n) == 1) {

su[m] = n;

m++;

}

n++;

}

int i = 0;

while (i < 200) {

printf("%u\t", su[i]);

if ((i + 1) % 10 == 0) {

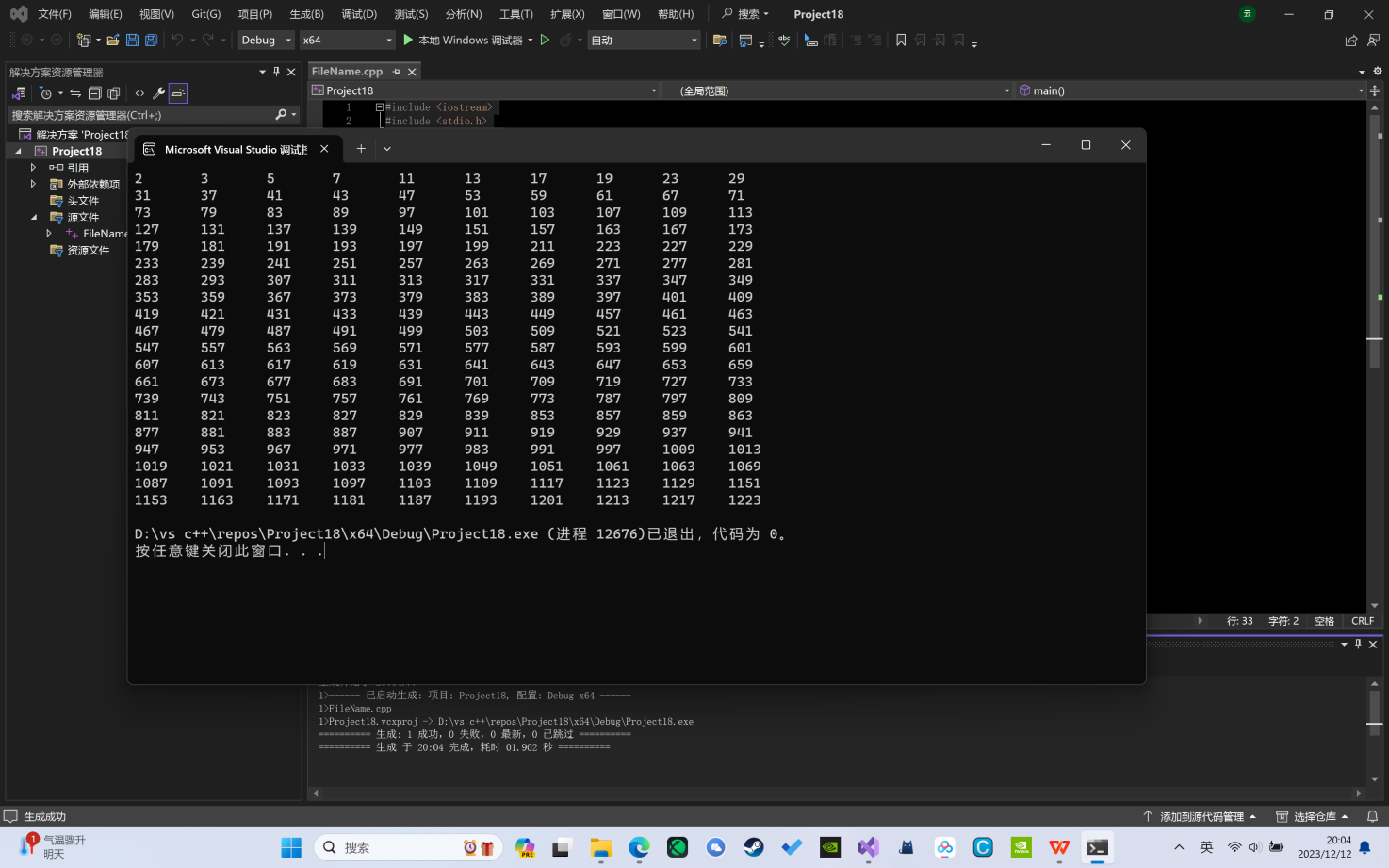
cout << endl;

}

i++;

}

}



4.程序：

头文件mytriangle.h：

#pragma once

#include <math.h>

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

double area(double side1, double side2, double side3);

**cpp文件mytriangle.cpp：**

#include <math.h>

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

if (side1 + side2 > side3 && side2 + side3 > side1 && side1 + side3 > side2) {

return 1;

}

else {

return 0;

}

}

double area(double side1, double side2, double side3) {

double square, s;

s = (side1 + side2 + side3) / 2;

square = sqrt(s \* (s - side1) \* (s - side2) \* (s - side3));

return square;

}

**cpp文件（主函数）：**#include <iostream>

#include "mytriangle.h"

using namespace std;

int main() {

double a, b, c, C;

cout << "请输入三角形的三条边：" << endl;

cin >> a >> b >> c;

if (is\_valid(a, b, c) == 1) {

cout << "该三角形的面积为：" << area(a, b, c) << endl;

}

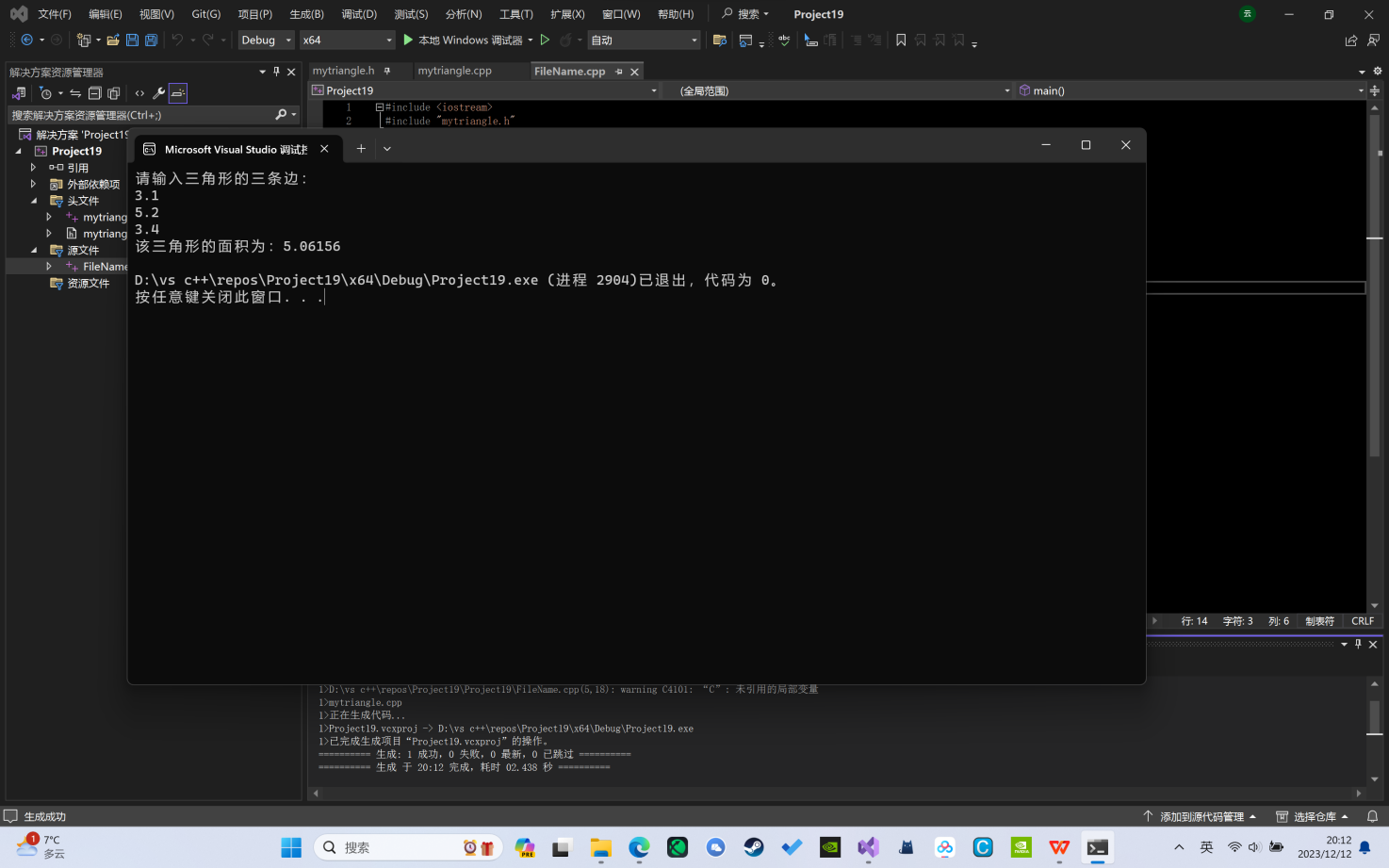
else {

cout << "这不是一个三角形！" << endl;

return 0;

}

}



**5.程序：**

#include<iostream>

using namespace std;

int peach(int tao, int day) {

if (day == 1) {

return tao;

}

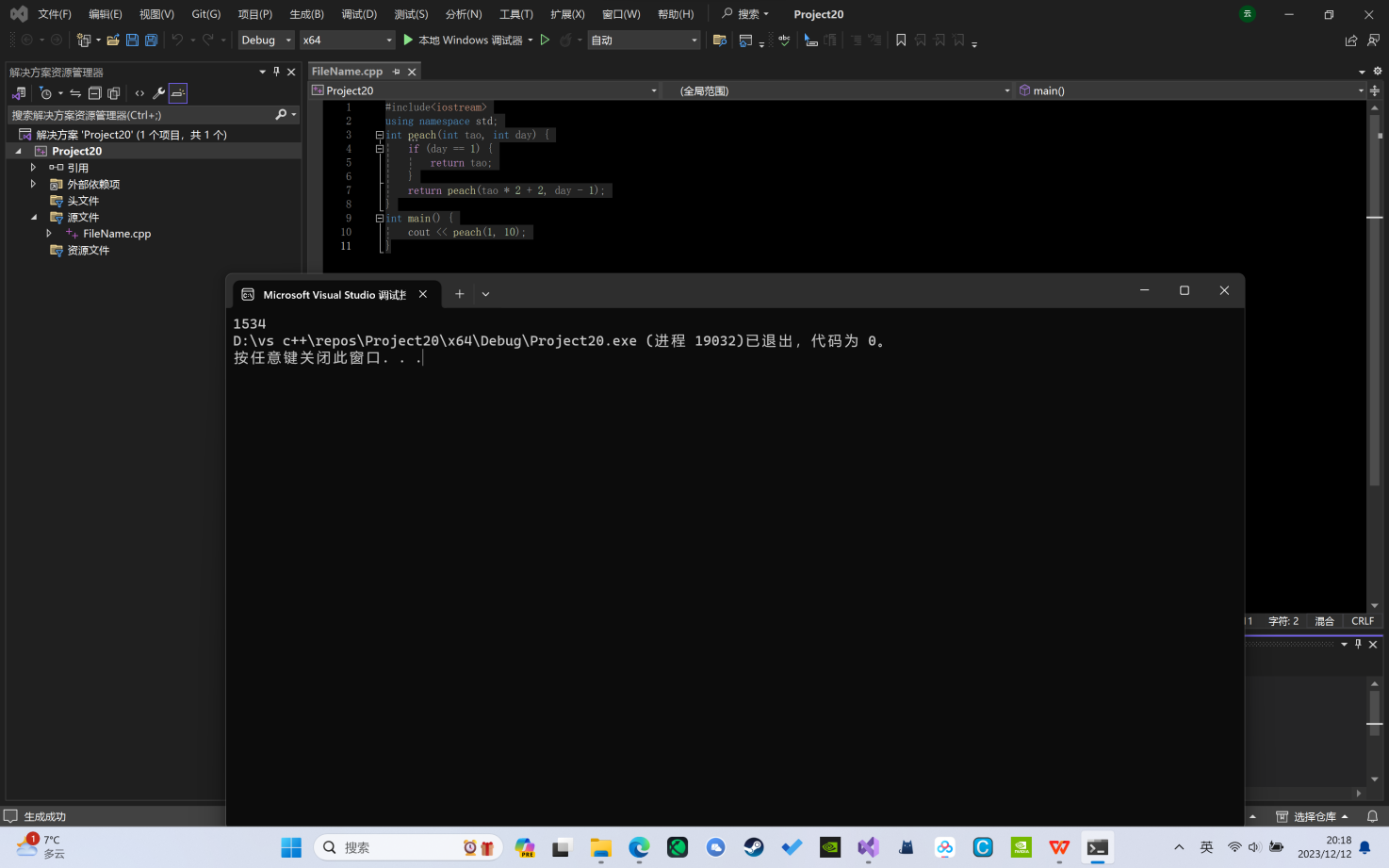
return peach(tao \* 2 + 2, day - 1);

}

int main() {

cout << peach(1, 10);

}



**五、遇到的问题与解决方法**

**此次实验中，大多数题目还是比较简单的，最大的难点应该在于最后的递归算法。最初我对于递归了解甚少，后来查阅了相关书籍才明白递归算法的本质，最终解出本题。**

1. **体会**

**更加深入地了解了c++程序运行的过程和本质。**

**了解了一些基础的算法（如递归）。**

**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abcdabcaba\_\_\_\_\_\_\_\_

(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**

**1.程序：**#include <iostream>

#include <stdio.h>

using namespace std;

int main() {

int a[10];

int i = 0;

cout << "Enter 10 numbers:";

while (i < 10) {

scanf\_s("%u", &a[i]);

i++;

}

int j = 0;

while (j < 10) {

int n = 0;

while (n < j) {

if (a[j] == a[n]) {

break;

}

else {

n++;

}

}

if (n == j) {

printf("%u", a[j]);

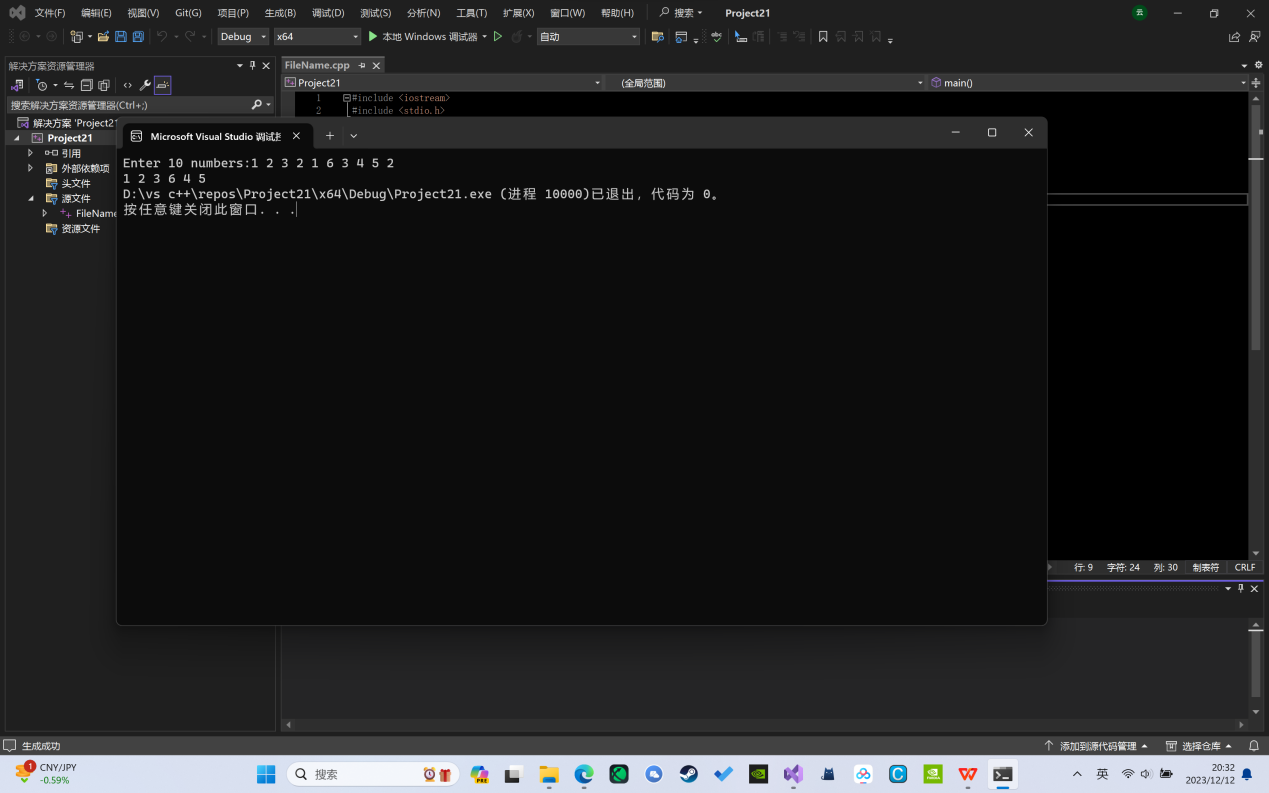
printf(" ");

}

j++;

}

}



**2.程序：**

#include<iostream>

#include<stdio.h>

using namespace std;

void paixu(double a[]) {

bool changed = true;

do

{

changed = false;

int j = 0;

while (j < 9) {

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

{

double k;

k = a[j], a[j] = a[j + 1], a[j + 1] = k;

changed = true;

}

j++;

}

} while (changed == true);

}

int main() {

double list[10];

int i = 0;

while (i < 10) {

scanf\_s("%lf", &list[i]);

i++;

}

paixu(list);

int n = 0;

while(n < 10) {

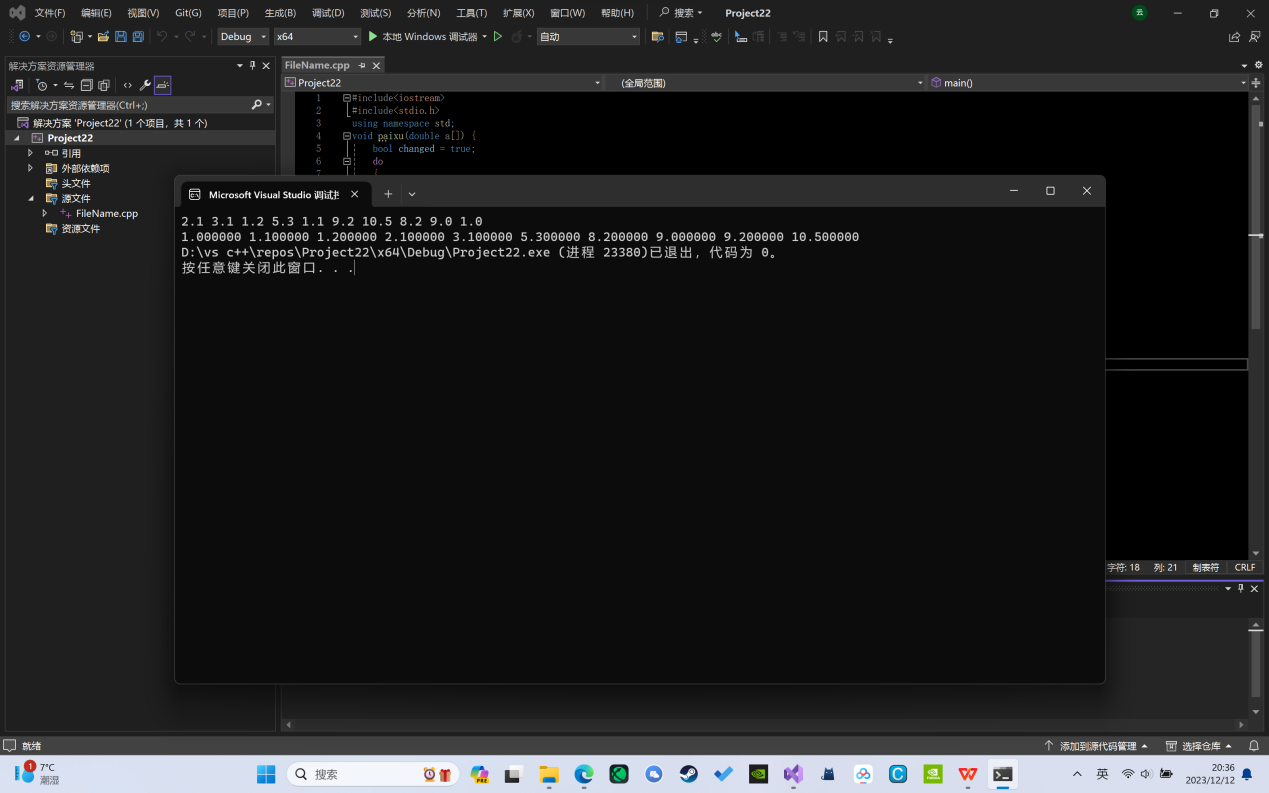
printf("%f", list[n]);

printf(" ");

n++;

}

}



**3.程序：**#include <iostream>

#include <stdio.h>

using namespace std;

int main() {

bool guizi[100] = { 0 };

int i = 1;

do {

int j = 0;

while (j<100) {

if ((j + 1) % i == 0) {

guizi[j] = !guizi[j];

}

j++;

}

i++;

} while (i < 101);

int m = 0;

while (m < 100) {

if (guizi[m] == 1) {

printf("%d", m+1);

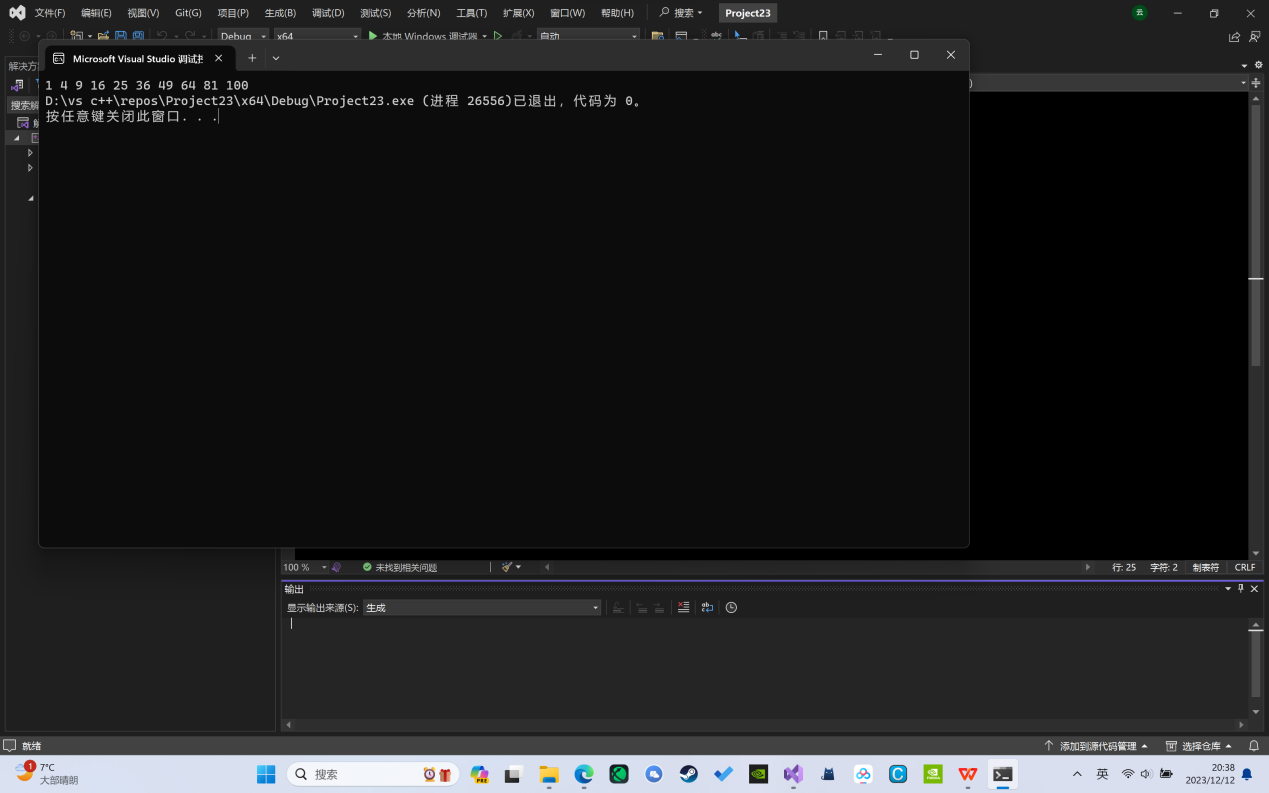
printf(" ");

}

m++;

}

}



**4.程序：**

#include <iostream>

#include <stdio.h>

using namespace std;

void paixu(int a[],int size3) {

bool changed = true;

do

{

changed = false;

int j = 0;

while (j < size3-1) {

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

{

int k;

k = a[j], a[j] = a[j + 1], a[j + 1] = k;

changed = true;

}

j++;

}

} while (changed == true);

}

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

int i = 0, n = 0, m = 0;

while (i < size1) {

list3[m] = list1[i];

i++, m++;

}

m = size1;

while (n < size2) {

list3[m] = list2[n];

n++, m++;

}

paixu(list3,size1+size2);

}

int main() {

int s1, s2,a=0;

printf("Enter list1,first num is the size of the list:");

cin >> s1;

int\* one = new int[s1];

while (a < s1) {

scanf\_s("%d", &one[a]);

a++;

}

cout << '\n';

a = 0;

printf("Enter list2,first num is the size of the list:");

cin >> s2;

int\* two = new int[s2];

while (a < s2) {

scanf\_s("%d", &two[a]);

a++;

}

cout << '\n';

int\* three = new int[s1 + s2];

merge(one, s1, two, s2, three);

delete[] one;

delete[] two;

cout << "The merged list is:";

int b = 0;

while (b < s1 + s2) {

printf("%d", three[b]);

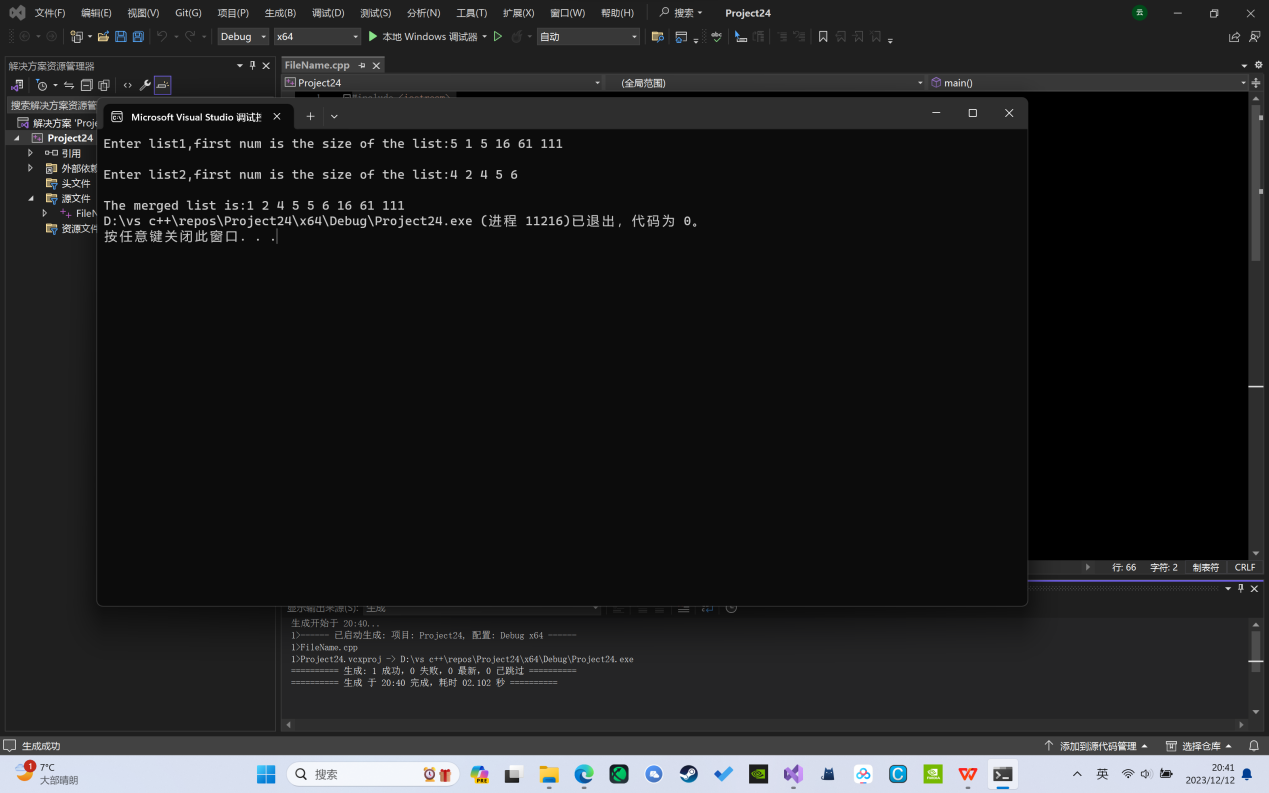
printf(" ");

b++;

}

delete[] three;

}



1. **程序：**

**头文件：**

#pragma once

int indexOf(char a[],int m, char b[],int n,int size1) {

if (m == size1) {

return n - size1 + 1;

}

else if (a[m] == b[n]) {

return indexOf(a, m + 1, b, n + 1, size1);

}

else {

return -1;

}

}

cpp文件：  
#include<string>

#include<cstring>

#include"FileName.h"

using namespace std;

int main() {

//输入部分

string s1, s2;

printf("Enter the first string:");

getline(cin, s1);

char\* str1 = new char[1000];

strcpy(str1, s1.c\_str());

printf("Enter the second string:");

fflush(stdout);

rewind(stdin);

getline(cin, s2);

char\* str2 = new char[1000];

strcpy(str2,s2.c\_str());

fflush(stdout);

//运算部分

int a = s1.size(), b = s2.size();

int j = 0;

do {

if (indexOf(str1, 0, str2, j, a) != -1) {

break;

}

j++;

}

while (a < b - j);

if (indexOf(str1, 0, str2, j, a) == -1) {

cout << "indexOf(\22" << s1 << "\22,\22" << s2 << ")is:" << -1;

}

else {

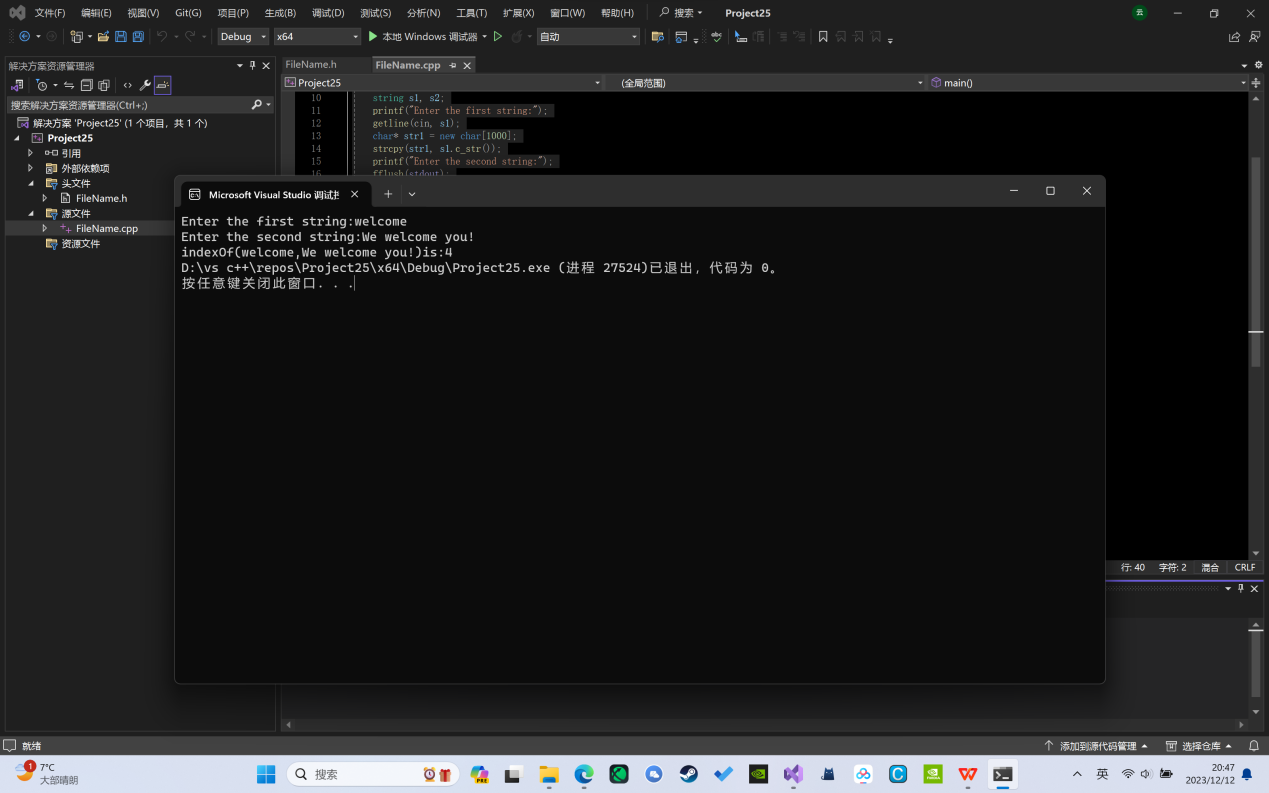
cout << "indexOf(\22" << s1 << "\22,\22" << s2 << ")is:" << indexOf(str1, 0, str2, j, a);

}

delete[]str1;

delete[]str2;

}



1. 程序：

头文件：  
#pragma once

#include<cstring>

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

int a = 0;

while (a < strlen(s)) {

int b = 0;

while (b < 26) {

if (static\_cast<int>(s[a]) == b + 65 || static\_cast<int>(s[a]) == b + 97) {

counts[b]++;

}

b++;

}

a++;

}

}

cpp文件：

#pragma warning( disable : 4996)

#include<iostream>

#include<string>

#include<cstring>

#include"FileName.h"

using namespace std;

int main() {

string str1;

int counts1[26] = { 0 };

char\* char1 = new char[1000];

cout << "Enter a string:";

getline(cin, str1);

strcpy(char1, str1.c\_str());

count(char1, counts1);

int c = 0;

while (c < 26) {

if (counts1[c] == 0) {

c++;

continue;

}

else {

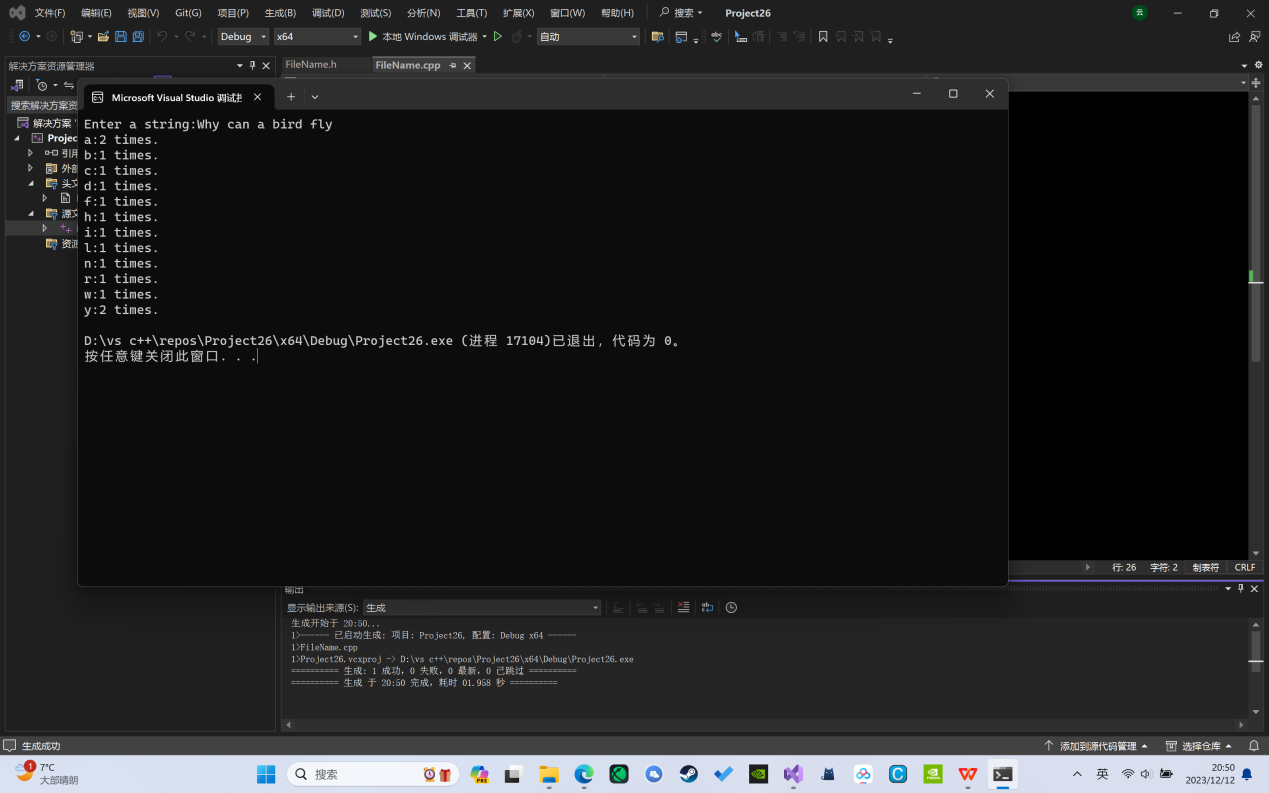
cout << static\_cast<char>(c+97) << ":" << counts1[c] << " times." << endl;

c++;

}

}

}

****

**7.程序：**#include<iostream>

using namespace std;

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;

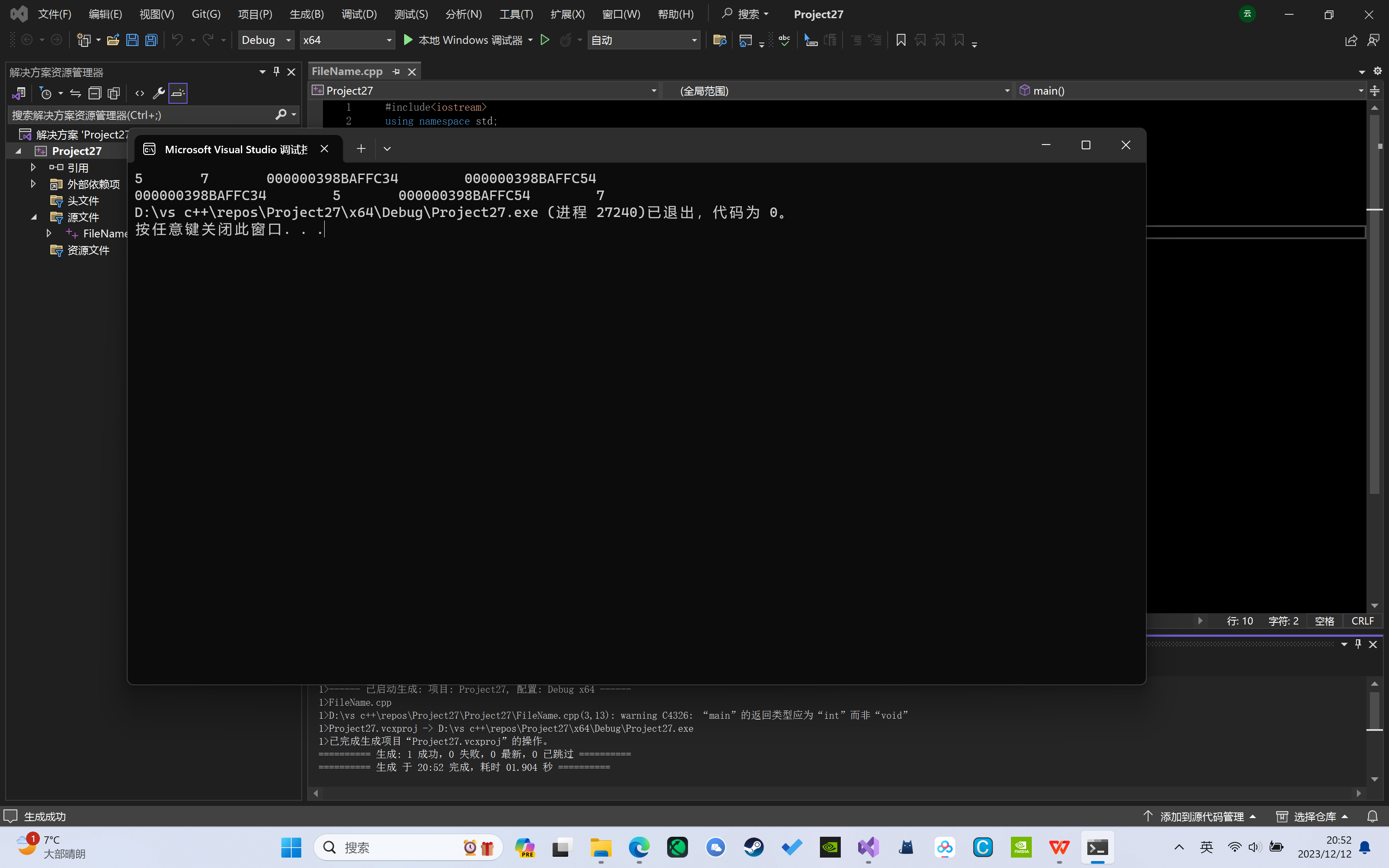
pj = &j;

i = 5; j = 7;

cout << i << '\t' << j << '\t' << pi << '\t' << pj << endl;

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

}



8.程序：

#include<iostream>

using namespace std;

int main() //C语言程序，要了解

{

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

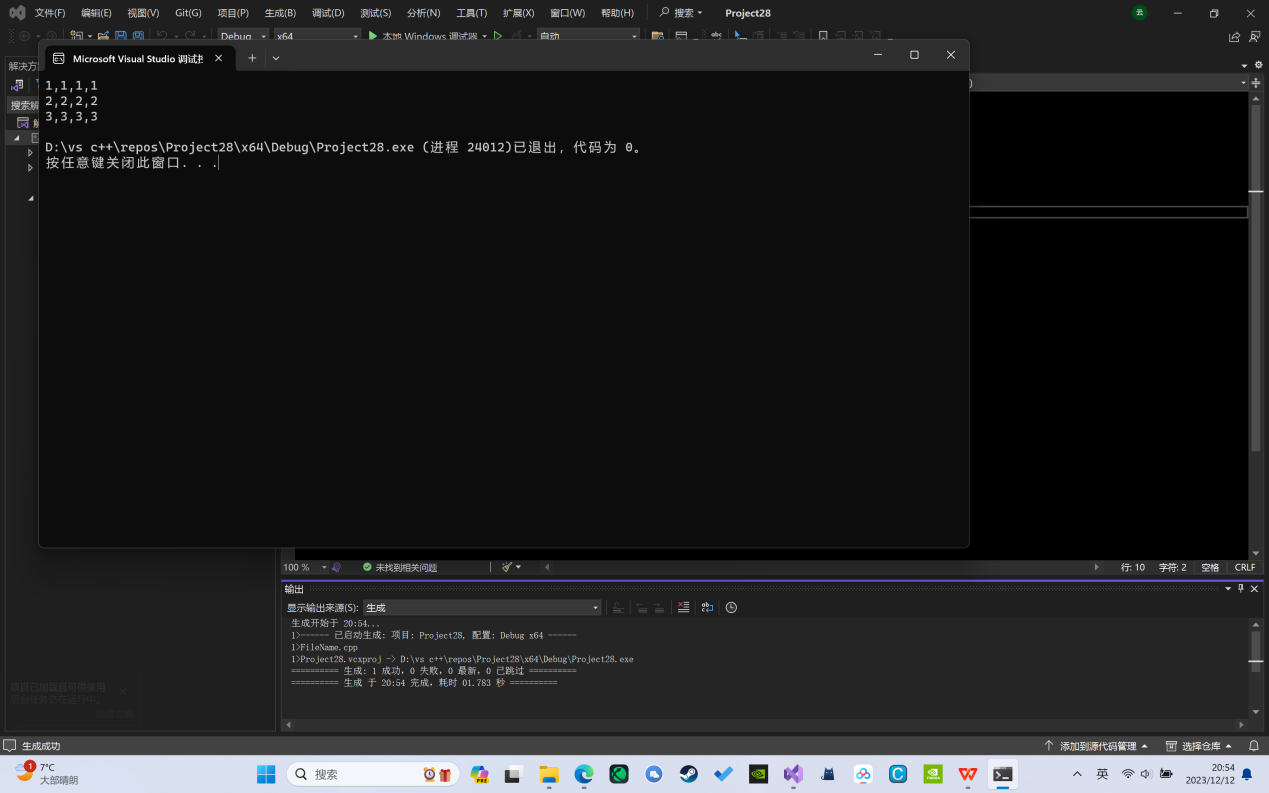
int\* p, i;

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

}



**9.**程序：

#include<stdio.h>

#include<iostream>

using namespace std;

void f(char\* st, int i)

{

st[i] = '\0';

cout << st; // printf(“%s\n”,st);

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

}

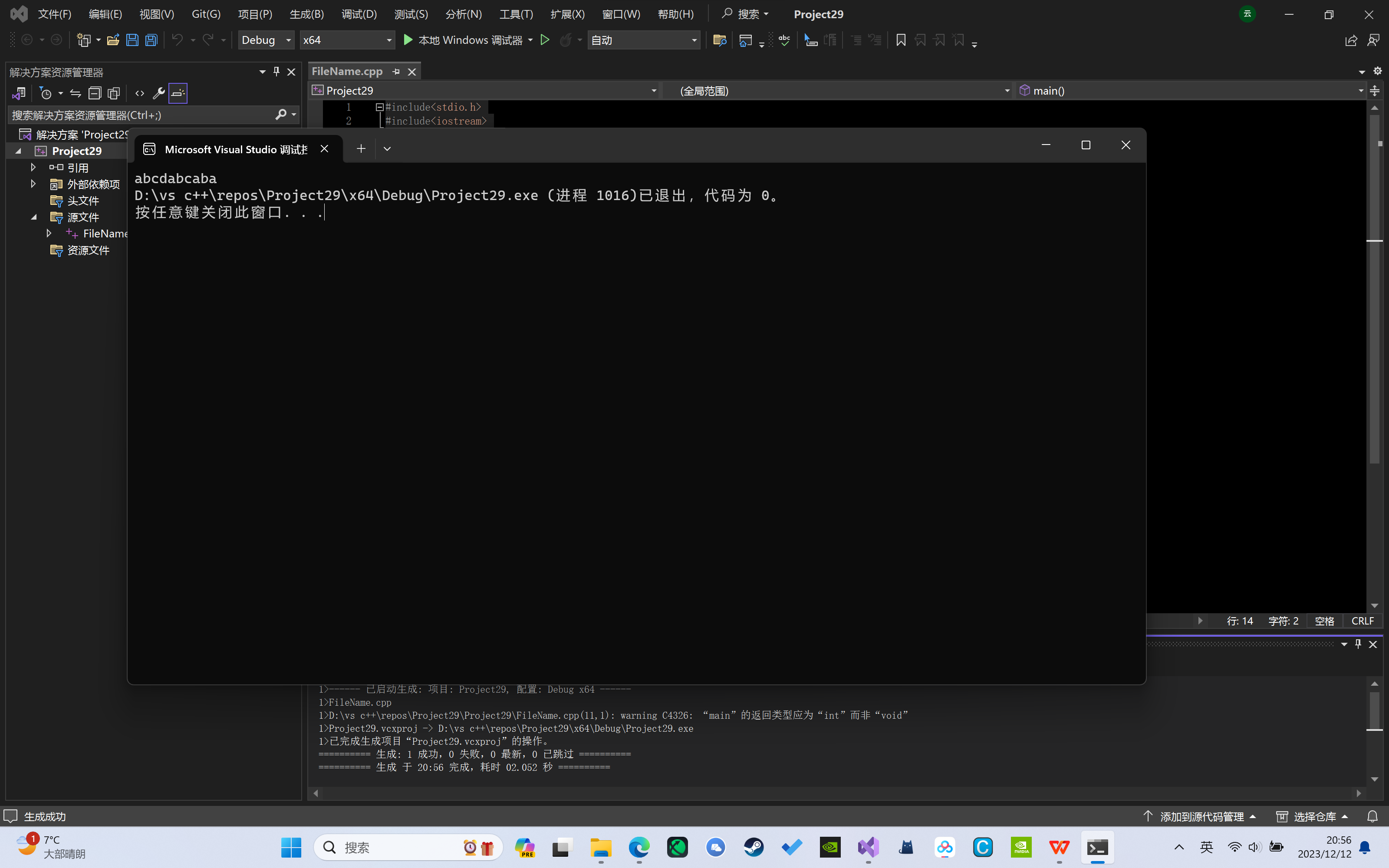
void main()

{

char st[] = "abcd";

f(st, 4);

}



**10.**程序：

#include<iostream>

using namespace std;

int\* f()

{

int\* list = new int[4]() ;

list[0] = 1;

list[1] = 2;

list[2] = 3;

list[3] = 4;

return list;

}

void main()

{

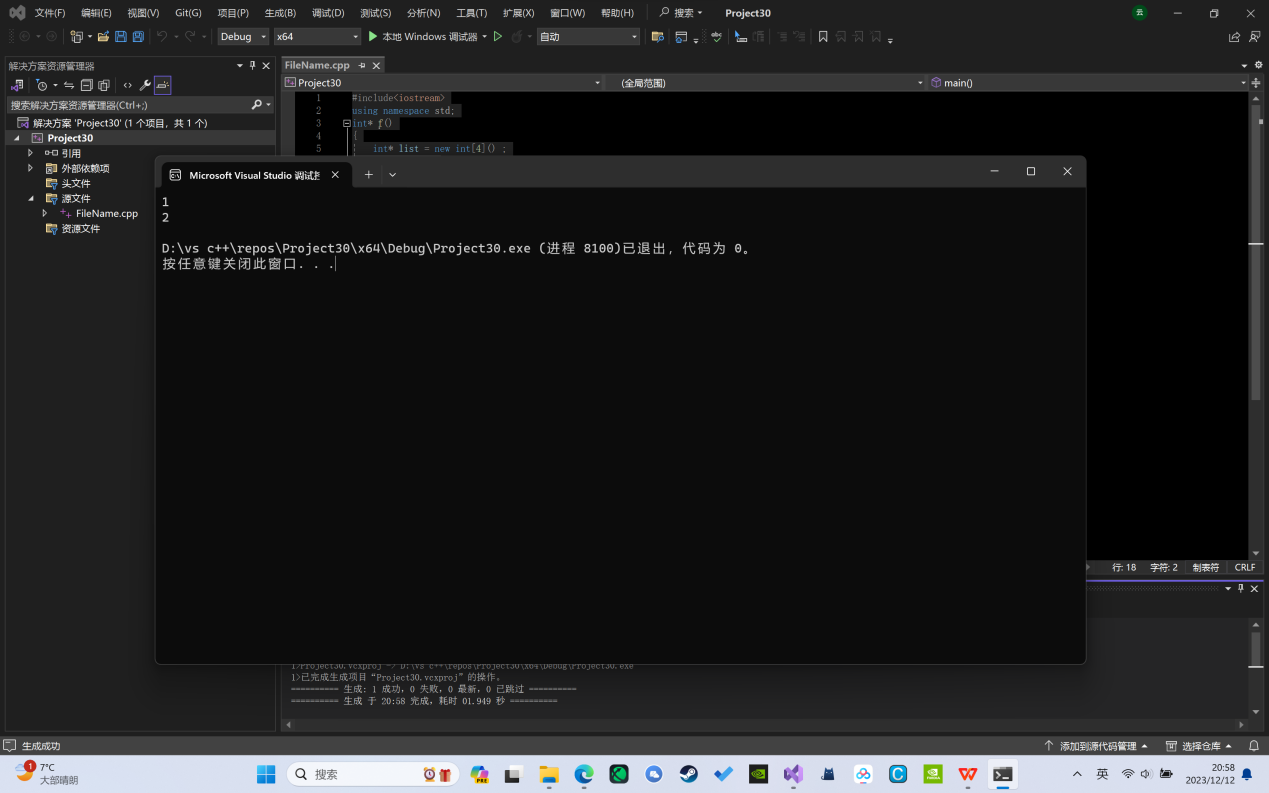
int\* p = f();

cout << p[0] << endl;

cout << p[1] << endl;

delete [] p;

}



1. 同题5。
2. **程序：**

#include <iostream>

#include <string.h>

using namespace std;

int parseHex(char hexString[]) {

char a[103] = { "0x" };

strcat\_s(a,hexString);

cout <<strtol(a, NULL, 16);

return 0;

}

int main() {

char b[100] = { 0 };

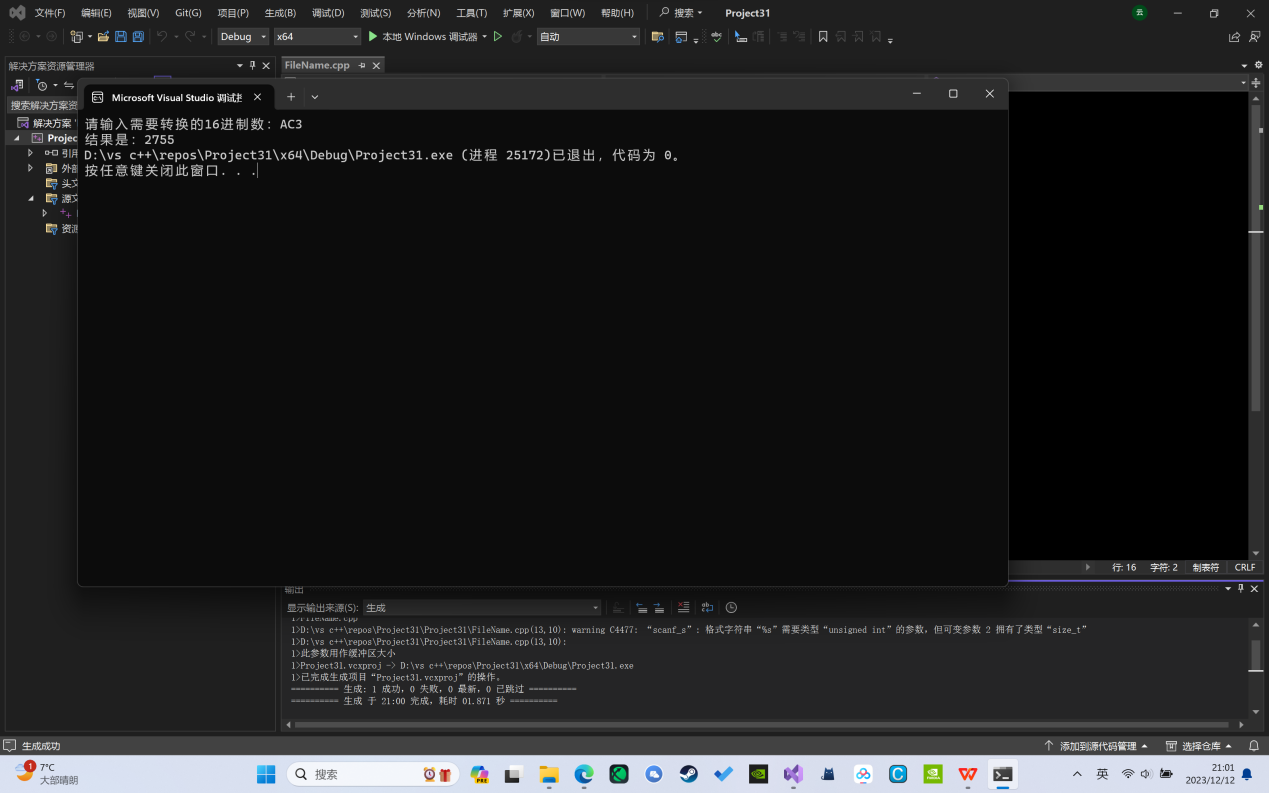
printf("请输入需要转换的16进制数：");

scanf\_s("%s", b,sizeof(b));

cout << "结果是：";

parseHex(b);

}



**四、遇到的问题与解决方法**

**1.本次实验难度较大，尤其是那几道关于输入数组或字符串并进行操作的题难度不小；**

**2.本次实验中，我在尝试接收输入的字符时发现，C语言风格的scanf和printf函数并不能处理C++语言中的string字符串类，最后只能选择getline函数；**

**3.scanf函数在连续使用时，上一个scanf函数并不会处理空格键的换行符，**

**导致\0滞留在缓冲区，下一个scanf函数直接被跳过。所以，在使用多个scanf函数时要用getchar函数处理\0；**

**4.题5判断字串问题中，字串判别函数采用了递归算法；**

**5.在C++语言环境中，使用printf函数被认为是不安全的，但是可以在预处理命令前加入“**#pragma warning( disable : 4996)”就可以无视警告。

**五、体会**

学会了更多算法，掌握了更多C++以及C语言语法及函数（如atoi函数，strtol函数等）