**实验三**

3.1

#include<iostream>

int Funtion(int& x, int& y)

{

int t = 1;

for (int i = 2;i < (x < y ? x : y);i++)

{

if (x % i == 0)

t = i;

}

return t, x\* y / t;

}

int main()

{

int m = 0, n = 0;

int max = 0, min = 0;

std::cout << "请输入两个数：" << std::endl;

std::cin >> m >> n;

if (m == 0 || n == 0)

{

std::cout << "程序出错" << std::endl;

}

else

{

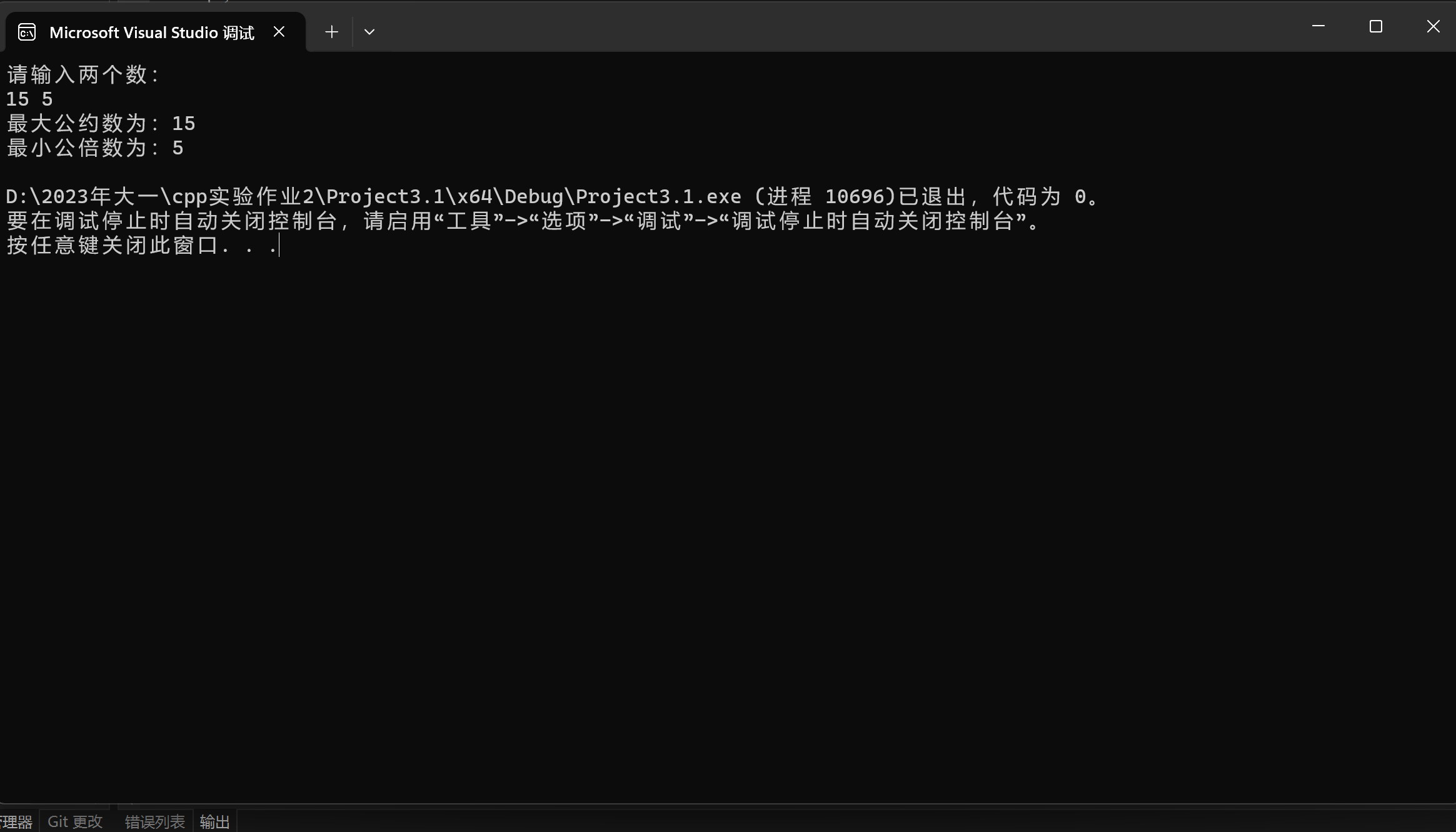
Funtion(m, n);

std::cout << "最大公约数为：" << m << std::endl

<< "最小公倍数为：" << n << std::endl;

}

}



3.2

#include<iostream>

bool is\_prime(int num)

{

int count = 2;

for (int i = 2;i < num;i++)

{

if (num % i != 0)

{

count++;

}

}

if (count < num)

{

return false;

}

else if (count == num)

{

return true;

}

}

int main()

{

int num = 0;

int x = 0;

std::cout << "请输入一个数：" << std::endl;

std::cin >> num;

bool is\_prime(int num);

std::cout << is\_prime(num) << std::endl;

int Const = 200;

int count = 0;

int i = 2;//众所周知1既不是素数也不是合数

while (count <= Const)

{

x = is\_prime(i);

if (x)

{

std::cout << i << "\t";

count++;

if (count % 10 == 0)

std::cout << "\n";

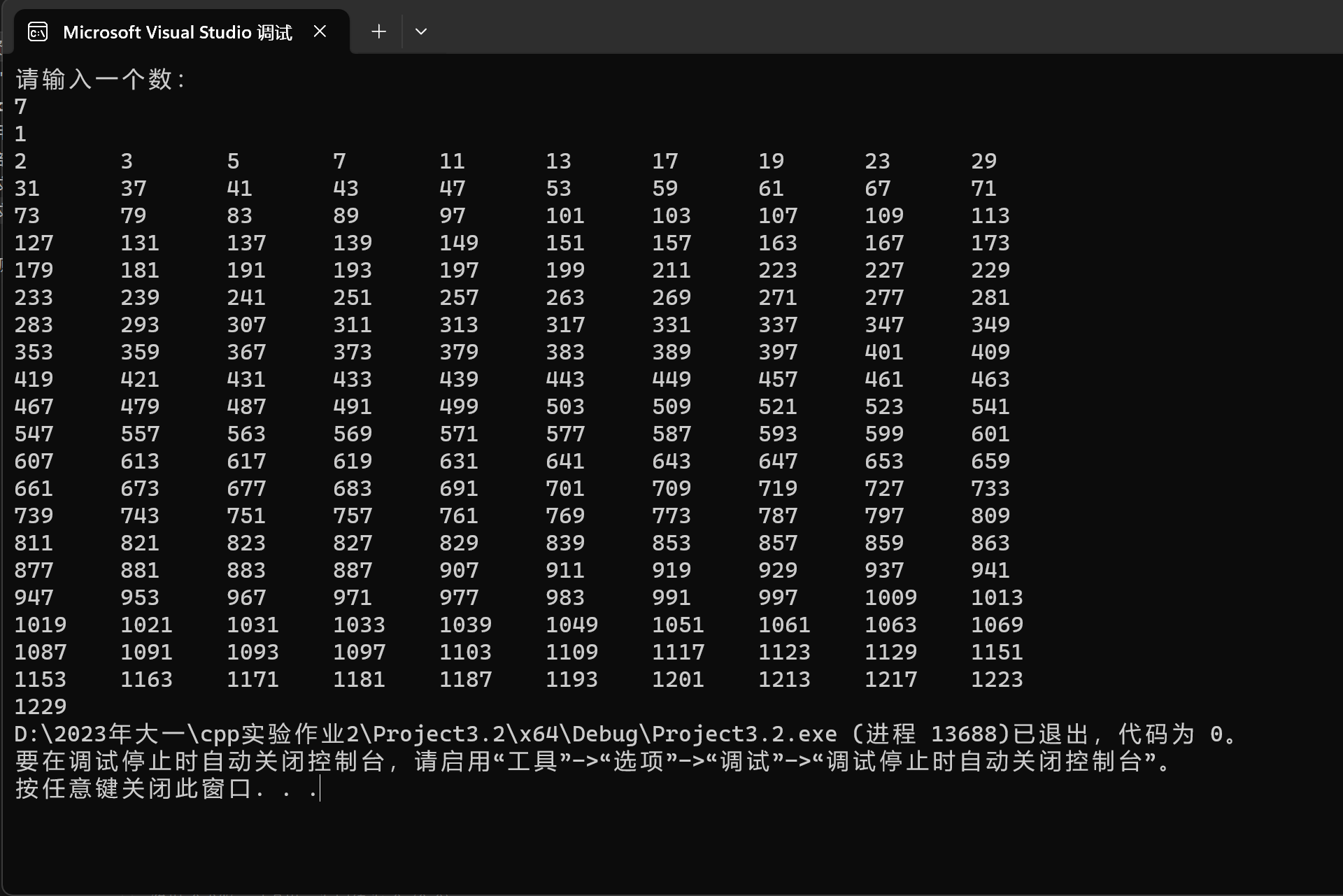
}

i++;

}

return 0;

}



3.3

Project3.3.cpp：

#include<iostream>

#include"mytemperature.h"

int main()

{

std::cout << "查找资料后发现cin后强制不换行需要设计函数" << std::endl;

std::cout << "Celsius Fahrenheit | Fahrenheit Celsius" << std::endl;

double cel = 0;

double fah = 0;

std::cout << "请输入：" << std::endl;

while (1)

{

std::cin >> cel;

std::cout << "\t" "\t""\t" "\t" << celsius\_to\_fah(cel) << "\t" "\t""\t" "\t""|""\t" "\t""\t" "\t";

std::cin >> fah;

std::cout << "\t" "\t""\t" "\t" << fahrenheit\_to\_cels(fah);

}

}

mytemperature.cpp:

double celsius\_to\_fah(double cel)

{

return cel \* 1.8 + 32;

}

double fahrenheit\_to\_cels(double fah)

{

return (fah - 32) / 1.8;

}

mytemperature.h:

#pragma once

double celsius\_to\_fah(double cel);

double fahrenheit\_to\_cels(double fah);



3.5

#include<iostream>

void monky(int a, int b = 1)

{

int y = (b + 1) \* 2;

if (a == 1)

{

std::cout << y;

}

else monky(a - 1, y);

}

int main()

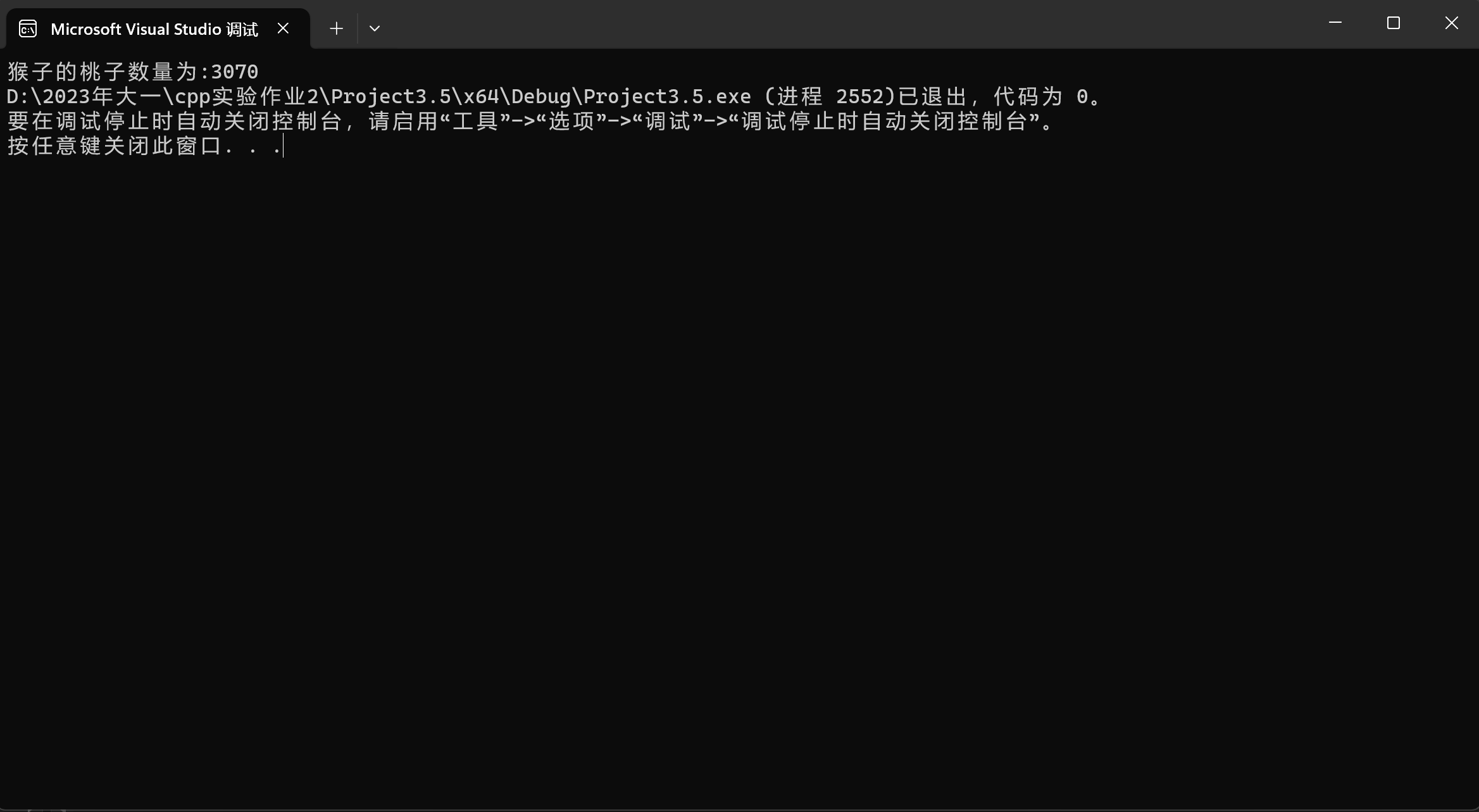
{

int a = 10;

std::cout << "猴子的桃子数量为:";

monky(a);

}



实验四数组和指针

4.1

#include<iostream>

using namespace std;

int main()

{

const int C = 10;

int list[C];

cout << "Enter ten numbers：" << endl;

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

cin >> list[i];

cout << "The distinct numbers are:" << list[0] << "\t";

for (int m = 1;m < C;m++)

{

for (int j = 0;j < m;j++)

{

if (list[m] != list[j])

{

if (j == m - 1)

cout << list[m] << "\t";

else

continue;

}

else

break;

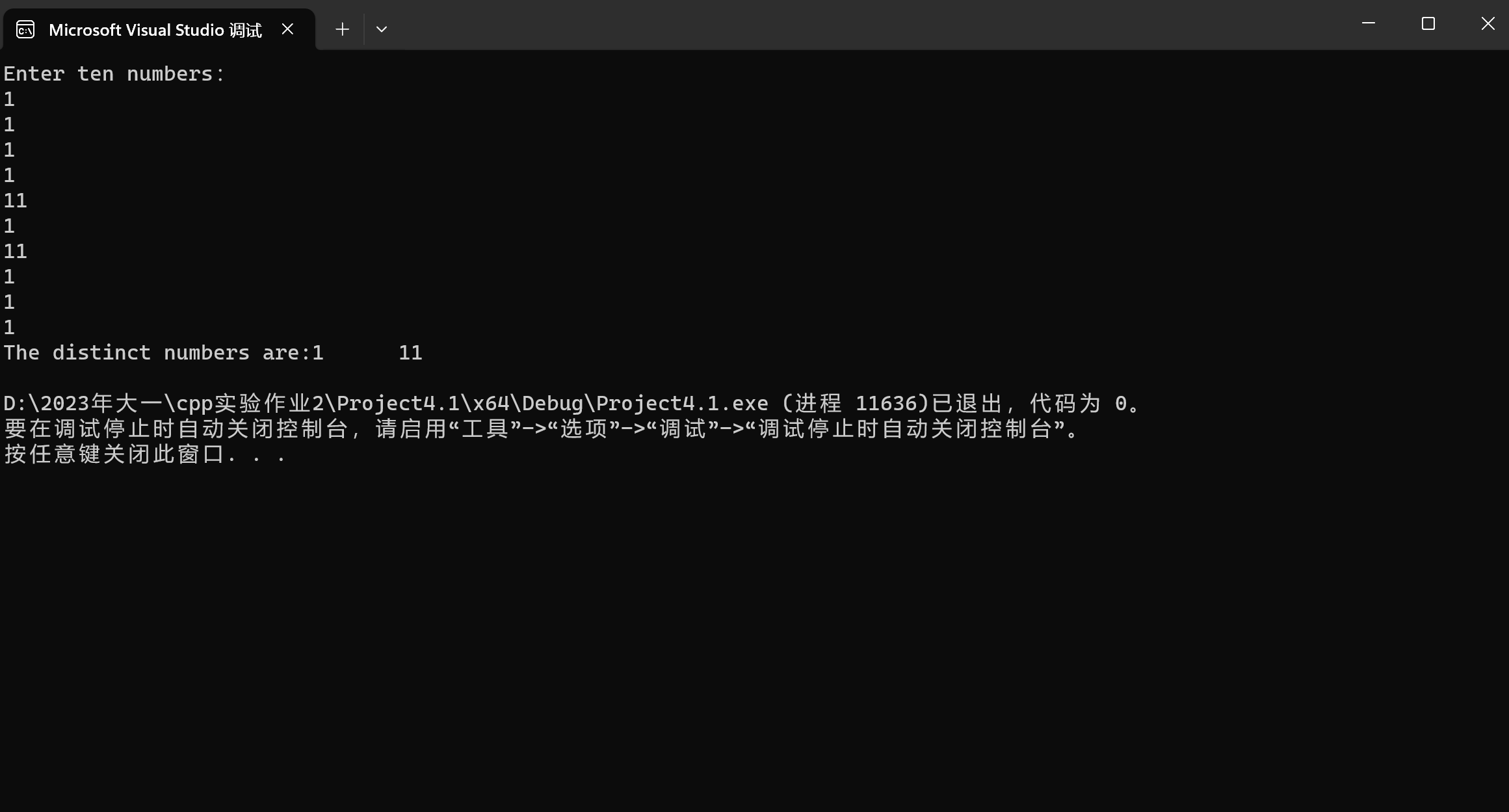
}

}

cout << endl;

return 0;

}



4.2

#include<iostream>

using namespace std;

const int C = 10;

void bubblesort(double list[10])

{

double temp;

bool changed = true;

do

{

changed = false;

for (int j = 0; j < C - 1; j++) {

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

temp = list[j];

list[j] = list[j + 1];

list[j + 1] = temp;

changed = true;

}

}

} while (changed == true);

}

int main()

{

double list[C];

cout << "Please enter ten number : \n";

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

cin >> list[i];

bubblesort(list);

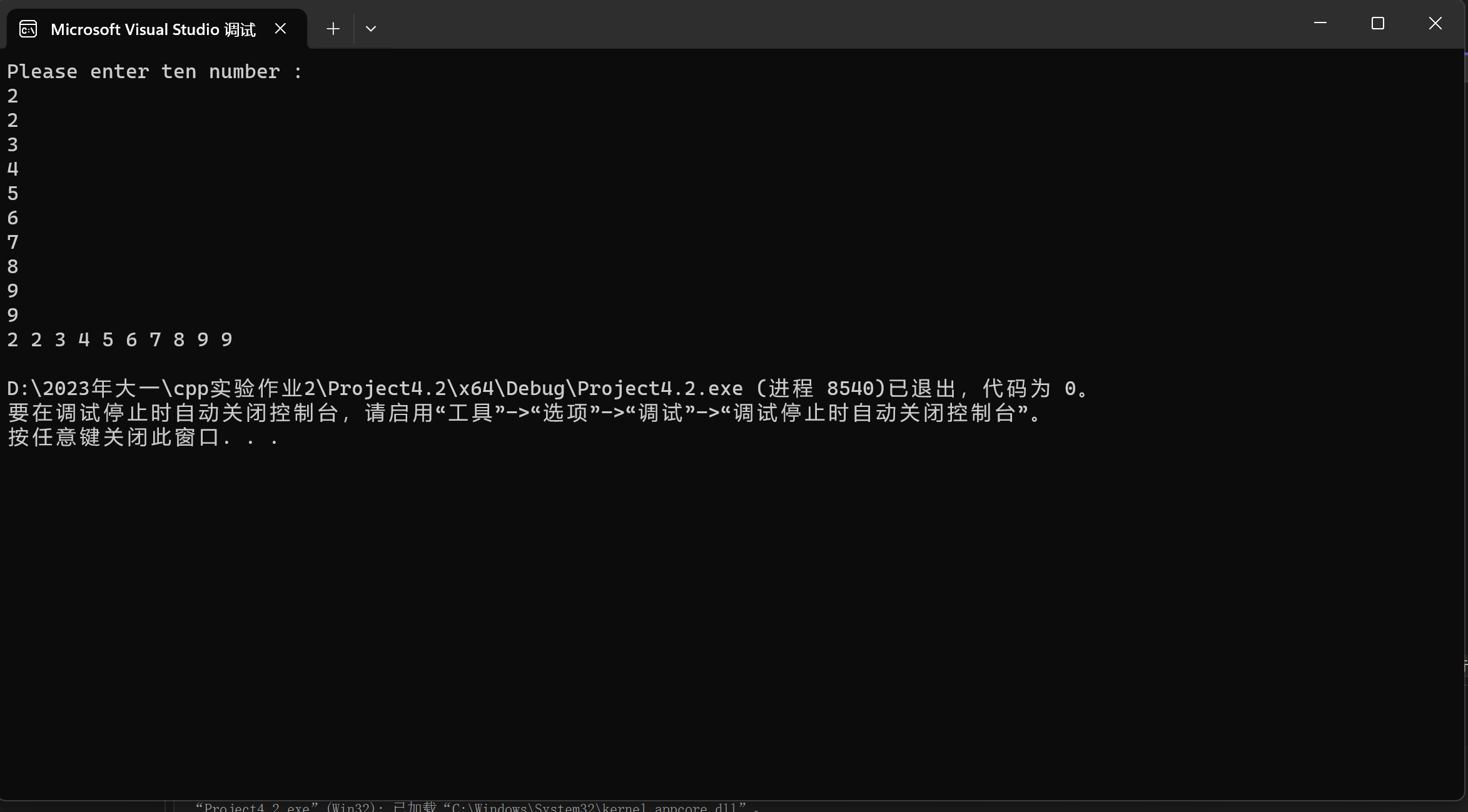
for (int k = 0;k < C;k++)

cout << list[k] << " ";

cout << endl;

return 0;

}



4.3

#include<iostream>

using namespace std;

int main()

{

const int C = 100;

bool list[C];

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

{

list[i] = false;

}

for (int j = 0;j < C;j++)

{

for (int k = j;k < C;k += j + 1)

{

if (list[k])

list[k] = false;

else

list[k] = true;

}

}

for (int m = 0;m < C;m++)

{

if (list[m])

cout << m + 1 << " ";

}

cout << endl;

return 0;

}



4.4

using namespace std;

void arrange(const int list[], int size);

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

int main()

{

int size1, size2, size3;

cout << "Enter size1: ";

cin >> size1;

cout << "Enter list1: ";

int\* list1=new int[size1];

for (int i = 0;i < size1;i++)

cin >> list1[i];

cout << "Enter size2: ";

cin >> size2;

cout << "Enter list2: ";

int\* list2 = new int[size2];

for (int i = 0;i < size2;i++)

cin >> list2[i];

size3 = size1 + size2;

int\* list3 = new int[size3];

merge(list1, size1, list2, size2, list3);

cout << endl;

return 0;

}

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

{

int temp;

for (int j = 0;j < size1;j++)

list3[j] = list1[j];

for (int k = 0;k < size2;k++)

list3[size1 + k] = list2[k];

cout << "The merged list is : ";

for (int m = 0;m < (size1 + size2) - 1;m++)

for (int b = 0;b < (size1 + size2) - 1;b++)

if (list3[b] > list3[b + 1])

{

temp = list3[b];

list3[b] = list3[b + 1];

list3[b + 1] = temp;

}

for (int a = 0;a < (size1 + size2);a++)

cout << list3[a] << " ";

}

void arrange(int list3[], int size)

{

int temp;

for (int m = 0;m < size - 1;m++)

for (int b = 0;b < size - 1;b++)

if (list3[b] > list3[b + 1])

{

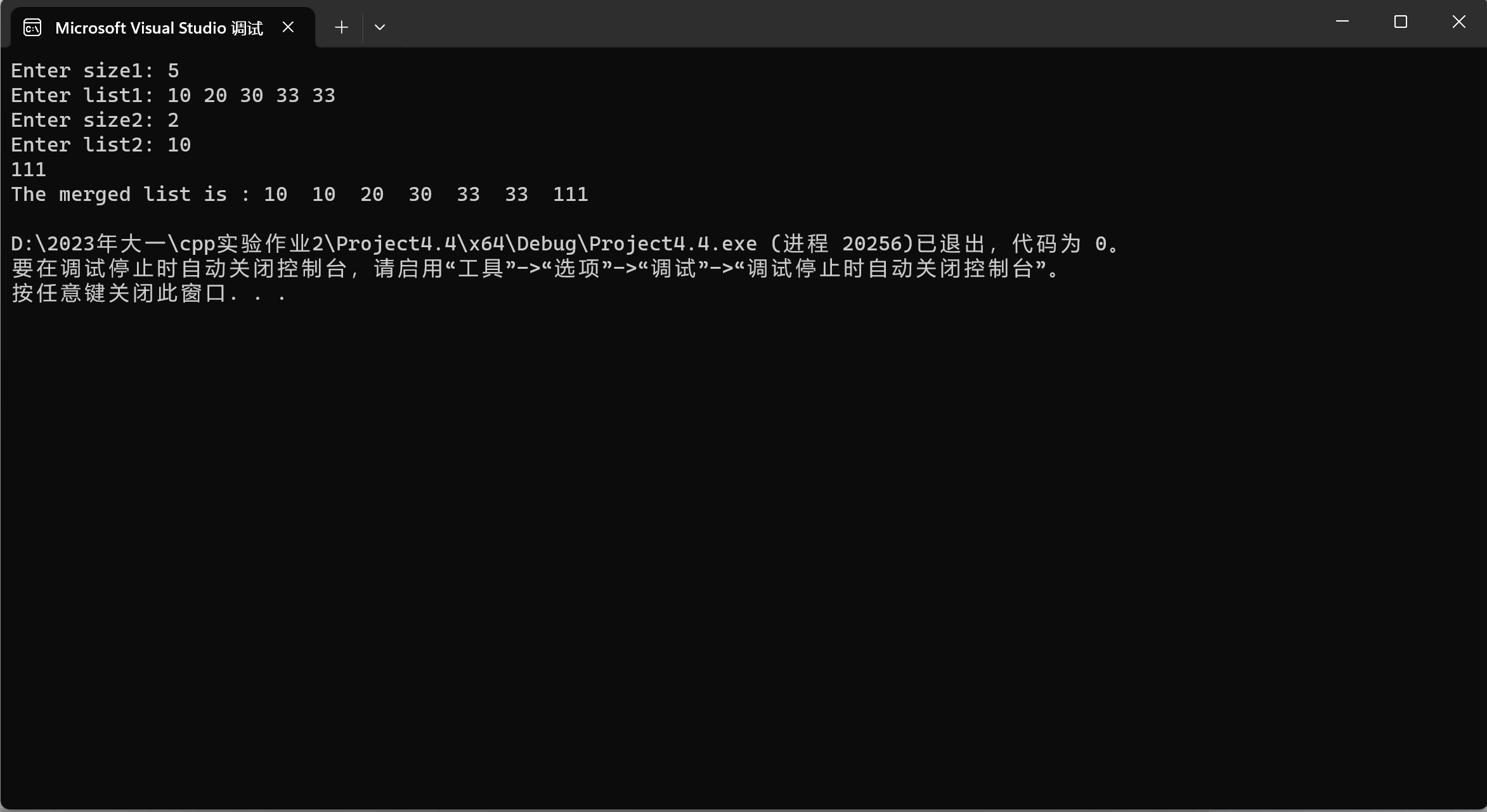
temp = list3[b];

list3[b] = list3[b + 1];

list3[b + 1] = temp;

}

}



4.5

#include<iostream>

#include<cstring>

using namespace std;

int indexOf(char\* s1, char\* s2);

int main()

{

const int C = 999;

char s1[C], s2[C];

cout << "Enter the first string : ";

cin.getline(s1, C);

cout << "Enter the second string: ";

cin.getline(s2, C);

cout << "indexOf(\"" << s1 << "\", \"" << s2 << "\") is " << indexOf(s1, s2) << endl;

return 0;

}

int indexOf(char\* s1, char\* s2)

{

int a, b, index, j, k;

a = strlen(s1);

b = strlen(s2);

bool\* s3=new bool[a];

for (int m = 0;m < a;m++)

s3[m] = false;

for (int i = 0;i < b - a + 1;i++)

{

for (j = 0, k = i;(j < a) && (k < i + a);j++, k++)

if (s1[j] == s2[k])

{

index = i;

s3[j] = true;

}

int h = 0;

for (int n = 0;n < a;n++)

{

if (s3[n] == false && i == b - a)

{

return -1;break;

}

if (s3[n])

h++;

}

if (h == a)

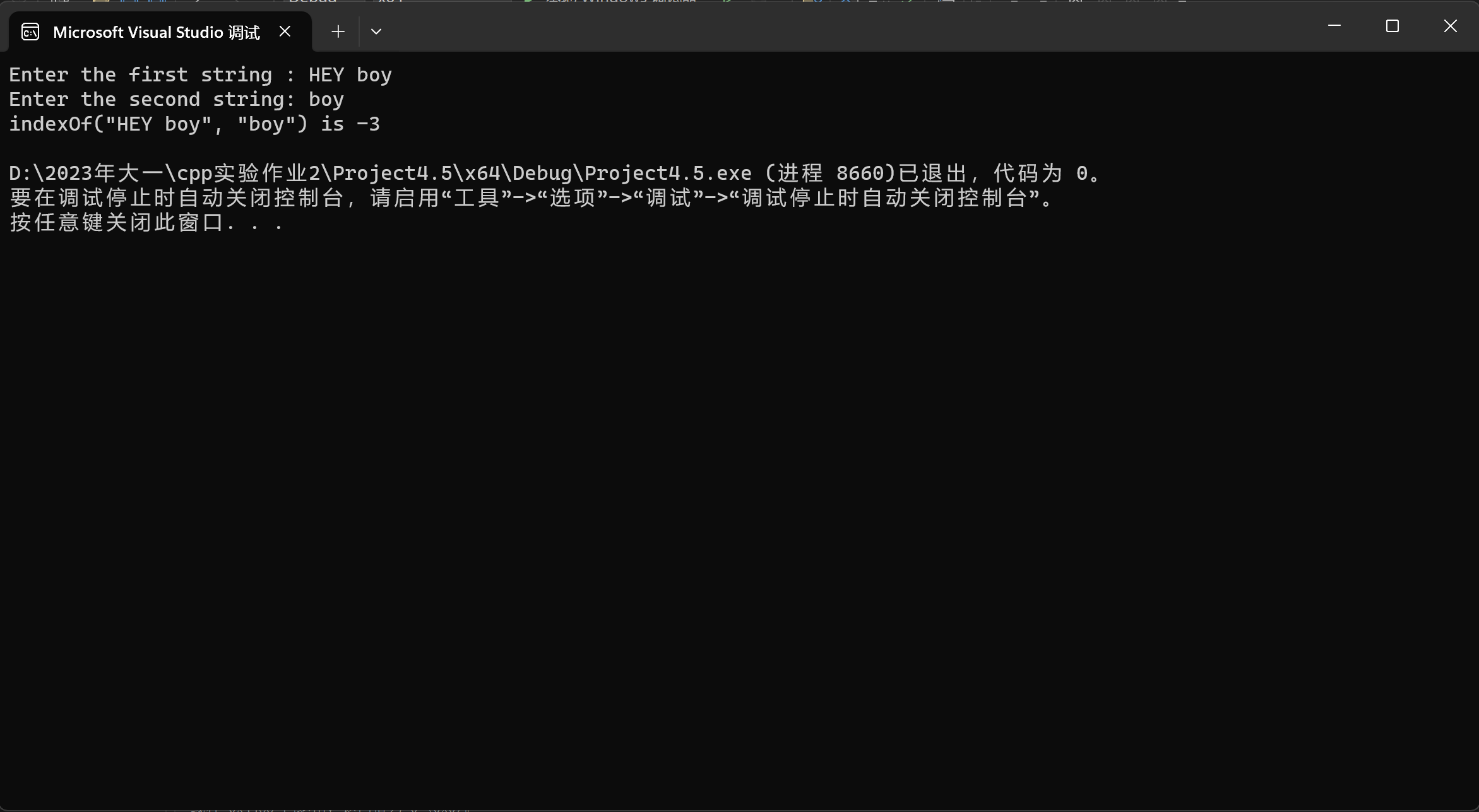
{

return index;

}

}

}



4.6

#include <iostream>

using namespace std;

const int num1 = 26; //num1=letters

const int num2 = 99; //num2=string

void countLetters(const char list[], int counts[])

{

for (int i = 0;i < num1;i++)

counts[i] = 0;

for (int j = 0;j < num2;j++)

{

counts[list[j] - 'a']++;

counts[list[j] - 'A']++;

}

}

void displayCounts(const int counts[])

{

for (int i = 0;i < num1;i++) {

if (counts[i] != 0)

cout << static\_cast<char>(i + 'a') << " : " << counts[i] << " times " << endl;

}

}

int main()

{

char list[num2];

int counts[num1];

cout << "Enter a string : ";

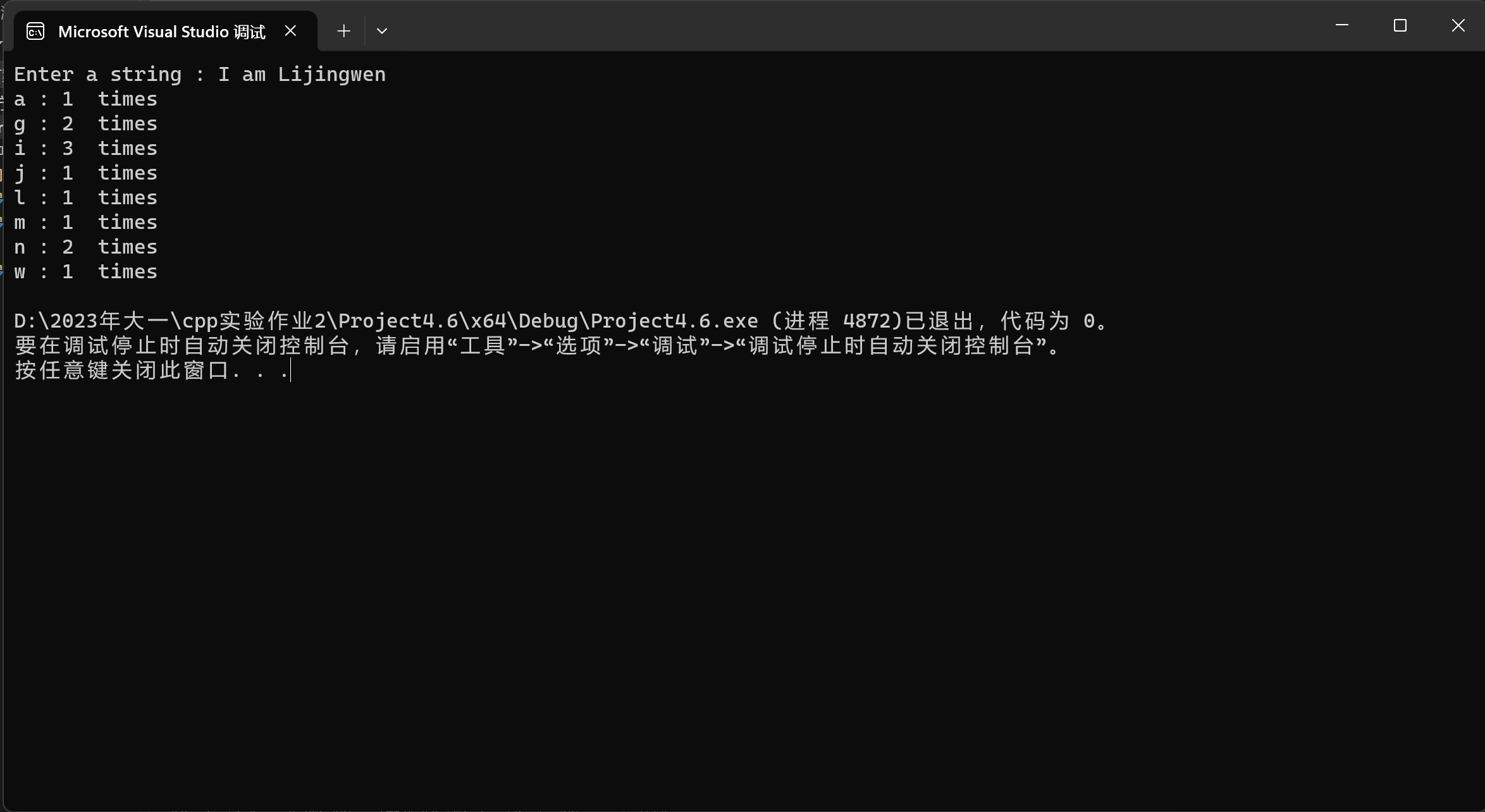
cin.getline(list, num2);

countLetters(list, counts);

displayCounts(counts);

return 0;

}



4.7（指针）

#include<iostream>

#include<cstring>

using namespace std;

int indexOf(char\* s1, char\* s2);

int main()

{

const int C = 999;

char s1[C], s2[C];

cout << "Enter the first string : ";

cin.getline(s1, C);

cout << "Enter the second string: ";

cin.getline(s2, C);

cout << "indexOf(\"" << s1 << "\", \"" << s2 << "\") is " << indexOf(s1, s2) << endl;

return 0;

}

int indexOf(char\* s1, char\* s2)

{

int a, b, index, j, k;

a = strlen(s1);

b = strlen(s2);

bool\* s3=new bool[a];

for (int m = 0;m < a;m++)

s3[m] = false;

for (int i = 0;i < b - a + 1;i++)

{

for (j = 0, k = i;(j < a) && (k < i + a);j++, k++)

if (s1[j] == s2[k])

{

index = i;

s3[j] = true;

}

int h = 0;

for (int n = 0;n < a;n++)

{

if (s3[n] == false && i == b - a)

{

return -1;break;

}

if (s3[n])

h++;

}

if (h == a)

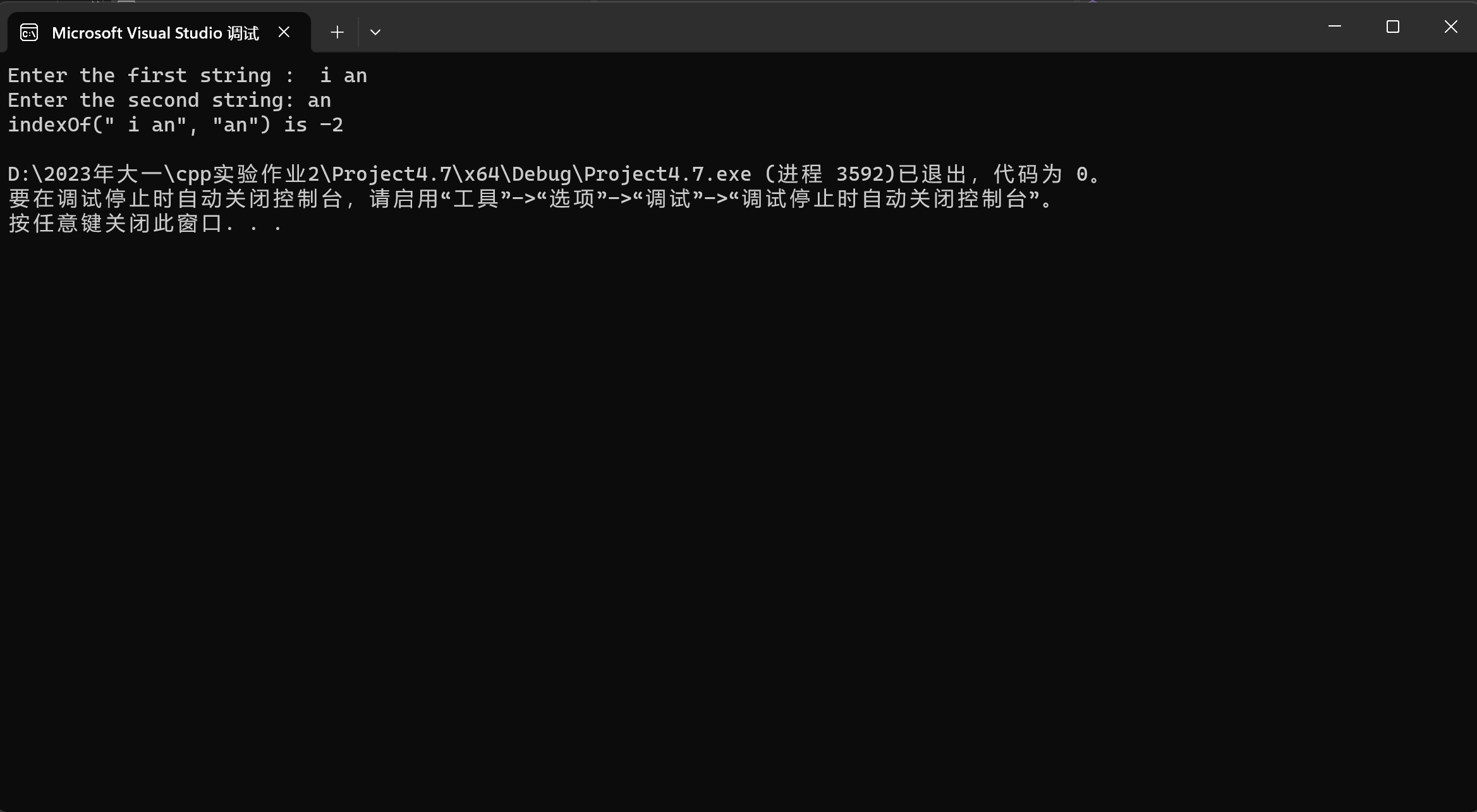
{

return index;

}

}

}



#include<iostream>

#include<cstring>

#include<cmath>

using namespace std;

int parseHex(const char\* const hexString);

int main()

{

int size = 999;

char\* s=new char[size];

cout << "Please enter a number in hexChar : \n";

cin.getline(s, size);

cout << "Tne number " << s << " in hexChar is " << parseHex(s) << " in decimal\n";

return 0;

}

int parseHex(const char\* const hexString)

{

int a, sum = 0;

a = strlen(hexString);

int\* list = new int[a];

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

{

if (hexString[i] >= 'A' && hexString[i] <= 'F')

list[i] = (static\_cast<int>(hexString[i]) - 55) \* (pow(16, a - 1 - i));

else

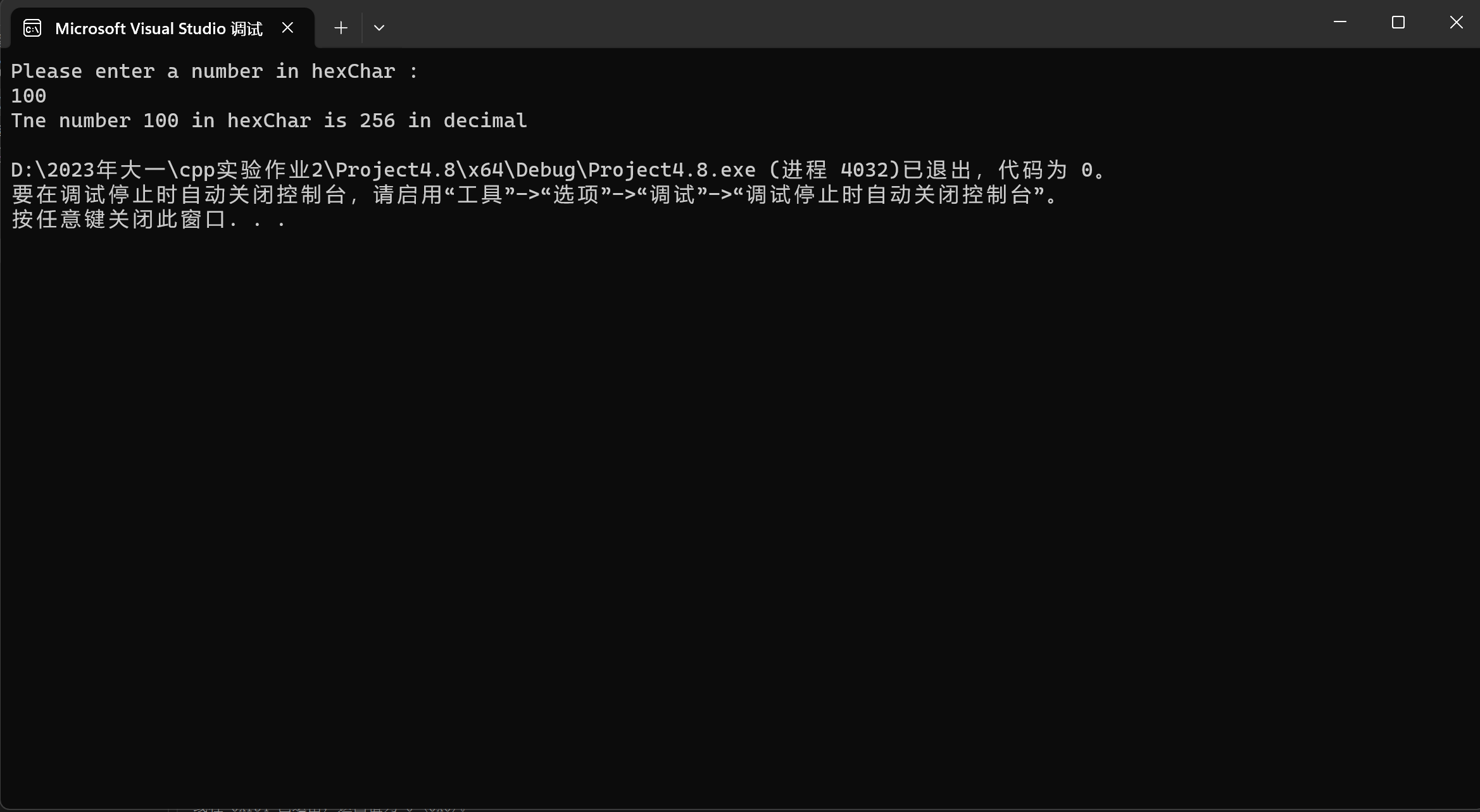
list[i] = (hexString[i] - 48) \* pow(16, a - i - 1);

sum = sum + list[i];

}

return sum;

}



4.9

#include<iostream>

using namespace std;

void arrange(int p[], int size);

int main()

{

int size = 0, a;

cin >> size;

int\* p = new int[size];

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

cin >> \*(p + i);

cout << "动态调试" << endl << "请输入你想要输出的数组" << endl;

cin >> a;

cout << "p[" << a - 1 << "] = " << \*(p + a - 1) << endl;

arrange(p, size);

for (int b = 0;b < size;b++)

cout << \*(p + b) << " ";

cout << endl;

delete p;

system("pause");

return 0;

}

void arrange(int p[], int size)

{

int temp;

for (int n = 0;n < size;n++)

{

for (int m = 0;m < size - 1;m++)

if (\*(p + m) > \*(p + m + 1))

{

temp = \*(p + m);

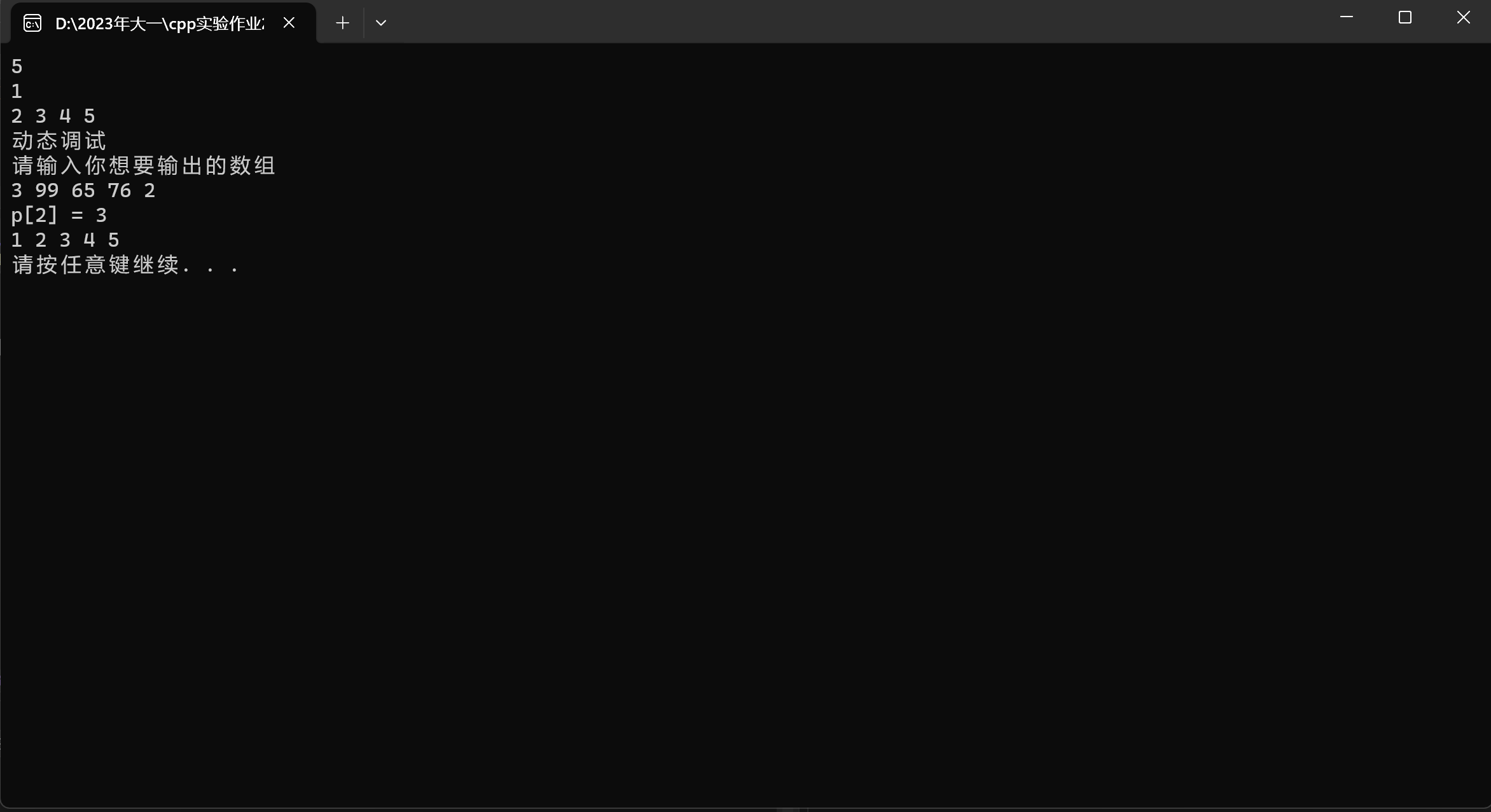
\*(p + m) = \*(p + m + 1);

\*(p + m + 1) = temp;

}

}

}



遇到的问题：

创建动态数组的格式不懂，一些算法不会做。

解决方法：

问师兄！！！