**计算机程序设计基础（C++)**

**实验报告**

专业班级： 软工2301班

学 号： 8209230110

姓 名： 肖峰

**实验三 函数**

**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？

2. 本实验中主函数调用函数时采用的是何种传递方式？

**1.函数中返回的值与函数类型一致是为了确保类型的一致性和类型安全。当定义一个函数时，我们需要指定函数的返回类型，这样在函数执行完毕后，返回的值就会被赋给一个与函数类型一致的变量或者表达式。如果返回的值与函数类型不一致，就会导致类型不匹配的错误，这可能会引起程序的不可预测行为或者崩溃。**

**2.引用传递和值传递。**

**四、算法分析，程序结果**

**一．代码：**

#include<iostream>

using namespace std;

int gy(int& a, int& b) {

int tempa = a, tempb = b;

a > b ? a = a,b=b : a = tempb,b=tempa;

while (b != 0) {

int temp = b;

b = a % b;

a = temp;

}

return a;

}//求最大公约数的函数

int gb(int& a, int& b) {

int tempa = a, tempb = b;

a > b ? a = a, b = b : a = tempb, b = tempa;

while (b != 0) {

int temp = b;

b = a % b;

a = temp;

}

return tempa \* tempb / a;

}//求最小公倍数的函数

int main() {

int m, n;

cout << "请输入两个自然数" << endl;

cin >> m >> n;

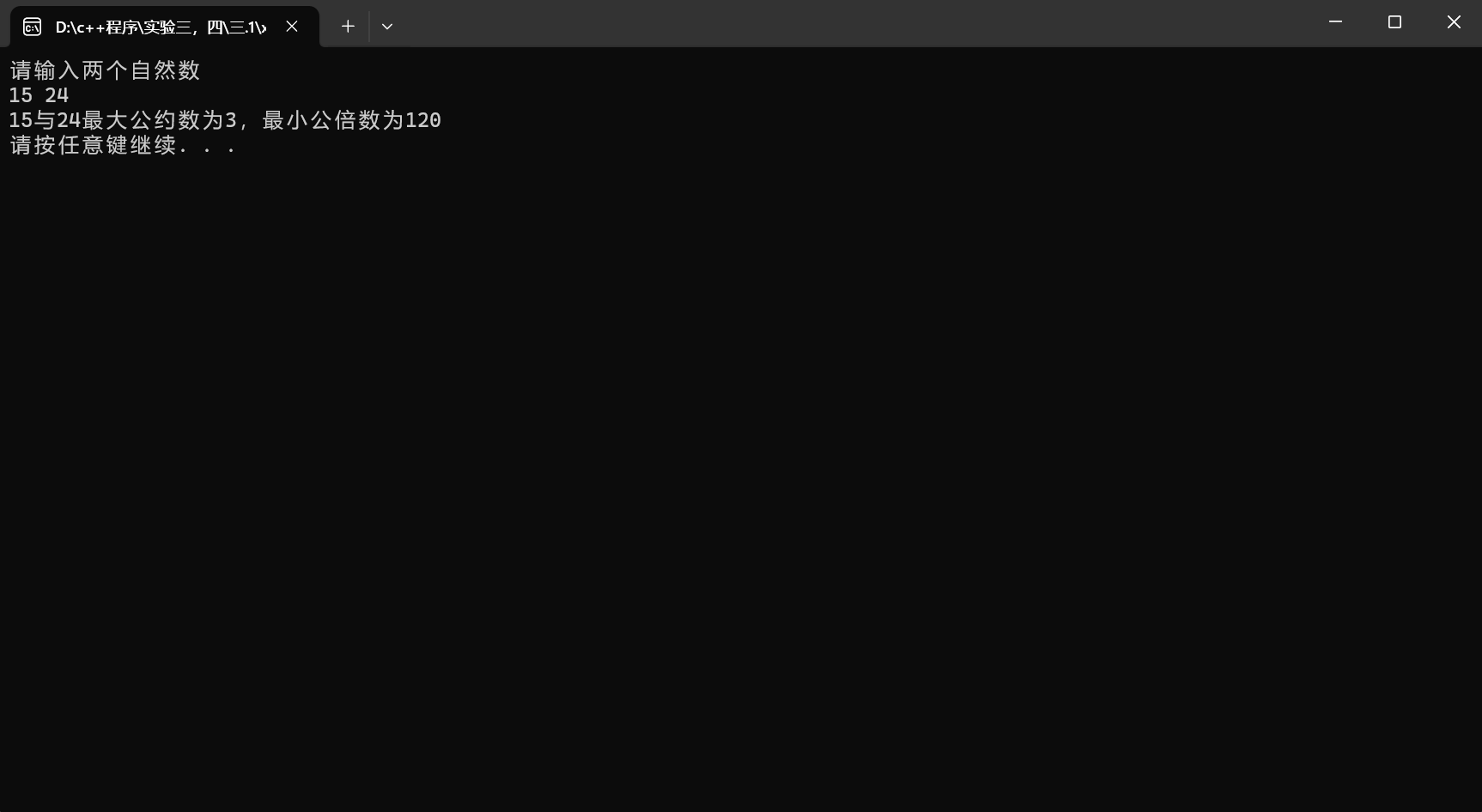
int tempm = m,temp1=m, tempn = n,temp2=n;

cout << m<< "与" << n << "最大公约数为" << gy(tempm, tempn) << "，最小公倍数为" << gb(temp1, temp2) << endl;

system("pause");

return 0;

}

结果：

**二.**

**代码：**

#include<iostream>

using namespace std;

bool is\_prime(int num);

int main() {

int i = 1;

int j = 0;

while (1) {

if (is\_prime(i) == true) {

cout << i << '\t';

++j;

if (j % 10 == 0) {

cout << endl;

};

};

if (j == 200) {

break;

};

++i;

}

system("pause");

return 0;

}

bool is\_prime(int num) {

int i=2;

while (i < num) {

int j;

j = num % i;

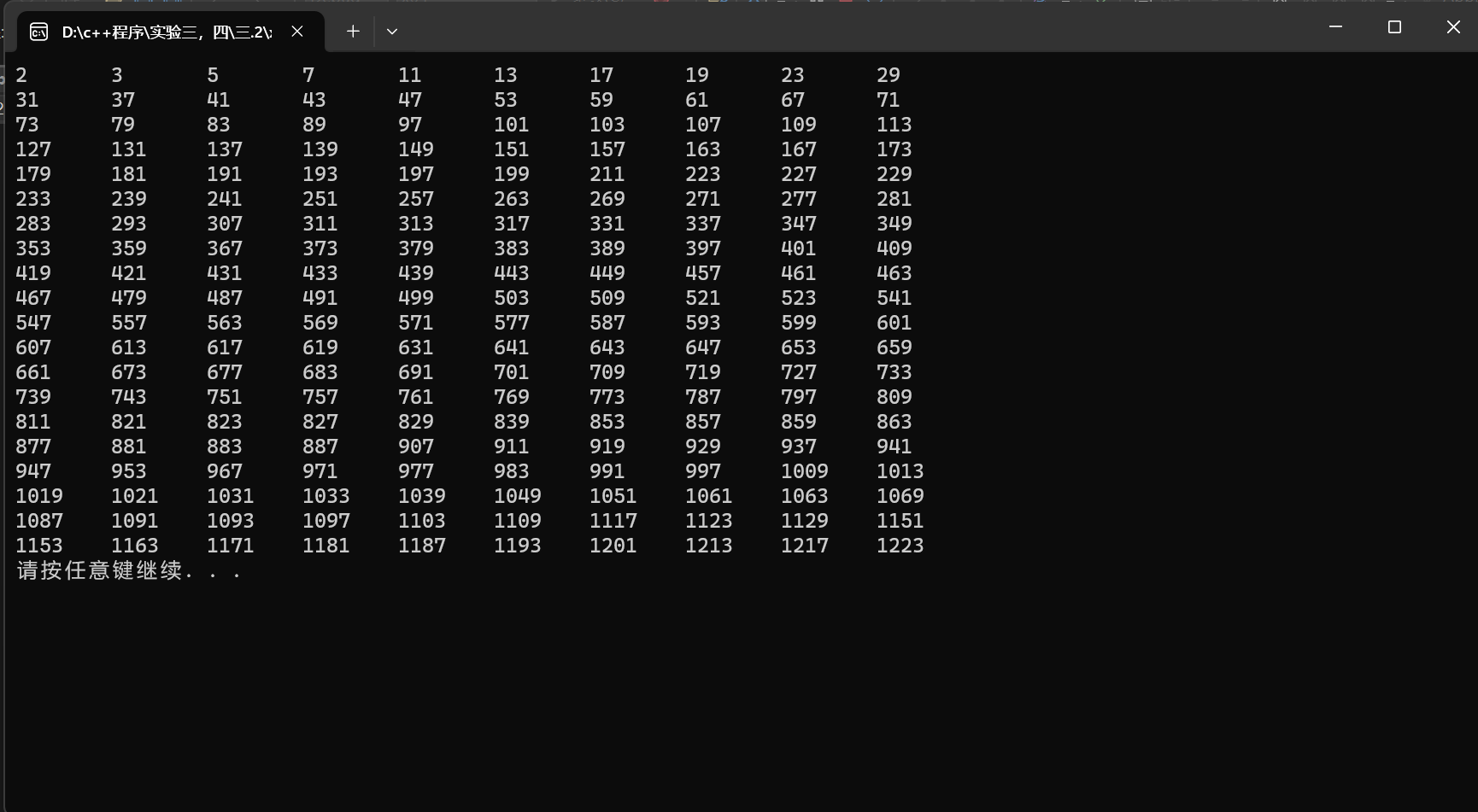
if (j == 0) {

return false;

}

i++;

}

 if (i == num ) {

return true;

}

else {

return false;

}

}

结果：

四．

代码：

1>mytriangle.h:

#pragma once

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

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

2>mytriangle.cpp:

#include<iostream>

#include"mytriangle.h"

using namespace std;

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

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

return true;

}

else {

return false;

}

};

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

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

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

return area;

};

3>#include<iostream>

#include"mytriangle.h"

using namespace std;

int main() {

double side1, side2, side3;

cout << "请输入三角形的三边长" << endl;

while (1) {

cin >> side1 >> side2 >> side3;

if (is\_valid(side1, side2, side3) == true) {

cout << "该三角形的面积为" << \_area(side1, side2, side3) << endl;

break;

}

else {

cout << "该三边无法构成三角形，请重新输入" << endl;

}

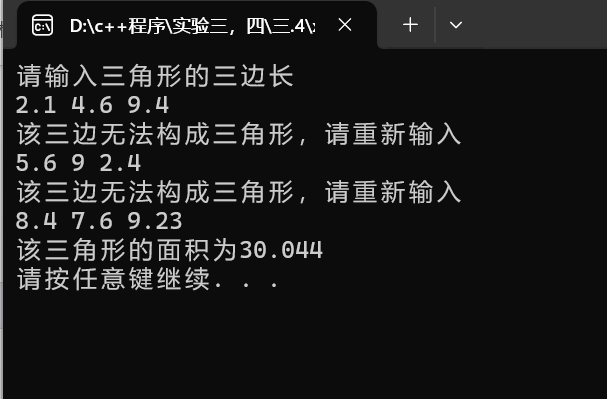
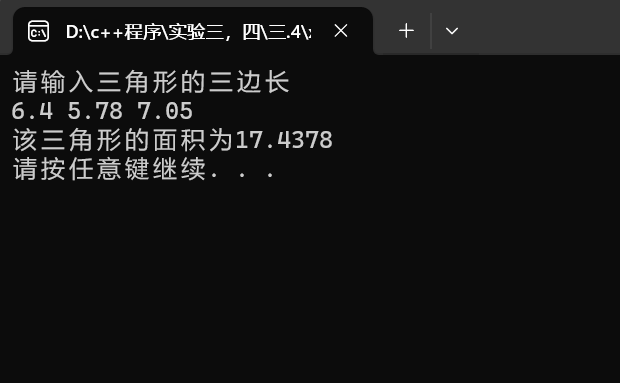
}

system("pause");

return 0;

}

结果：

五：

代码：

#include<iostream>

using namespace std;

int ct(int& a,int&b) {

a = 2 \* (a + 1);

b++;

if (b <= 10) {

ct(a,b);

}

else {

return a;

}

}

int main() {

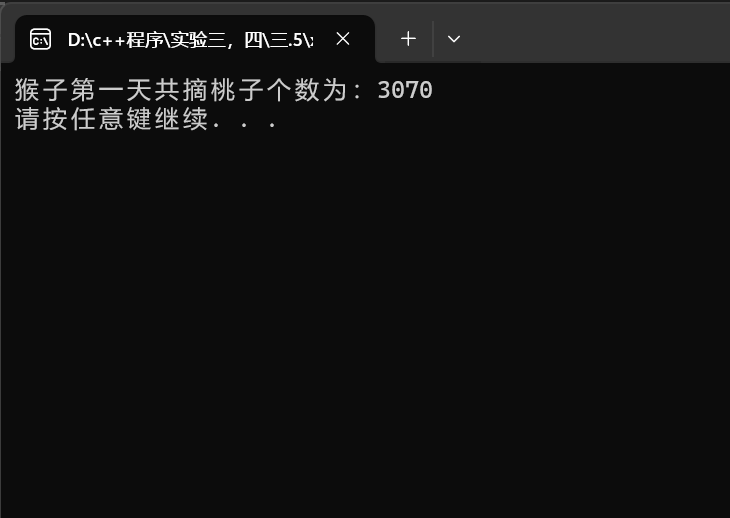
int a = 1, b = 1;//a为桃子数量

cout << "猴子第一天共摘桃子个数为：" << ct(a, b) << endl;

system("pause");

return 0;

}

结果：

**五、遇到的问题与解决方法**

1.怎么用c++语言实现解最大公约数的辗转相除法---用while循环来解决

2.如何使自己写的头文件在代码中可用--main函数上面加一句#include”头文件名称.h”

3.对函数递归的使用

**六、体会**

通过该实验的学习，我掌握了函数的定义、声明的方法，掌握了函数的编写要求，掌握了函数的调用方法，掌握了函数参数的传递方法，掌握了变量的作用域以及掌握了多文件编程方法。这次实验也让我对C++中函数的概念和用法有了更深入的了解。我学会了如何定义和调用函数，以及如何传递参数和返回值。同时，我也学会了如何使用函数来模块化程序，使得程序更易于理解和维护。通过实际的编程练习，我对函数的使用有了更加深刻的认识，并且提高了自己的编程能力和解决问题的能力。

在实验过程中，我遇到了一些问题，例如函数的参数传递和返回值的处理等，但通过查阅资料和进行调试，我最终成功地解决了这些问题。这让我更加熟悉了C++语言的特性和用法，也提高了我的问题解决能力。

总的来说，本次实验让我对C++函数有了更深入的了解，也让我更加熟练地掌握了C++的编程技能。

通**实验四 数组与指针**

**三、算法分析，程序结果**

**一．1：**

**代码：**

#include<iostream>

using namespace std;

int main() {

int a[10],d[10];

int b = 0;

cout << "请输入10个数" << endl;

for (int i = 0; i < 10; ++i) {

cin >> a[i];

}

for (int i = 0; i < 10; i++) {

bool ac = true;

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

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

ac = false;

}

}

if (ac == true) {

d[i - b] = a[i];

}

else {

b++;

}

}

int \*c=new int[10 - b];

for (int i = 0; i < 10 - b; ++i) {

c[i] = d[i];

cout << c[i] << " ";

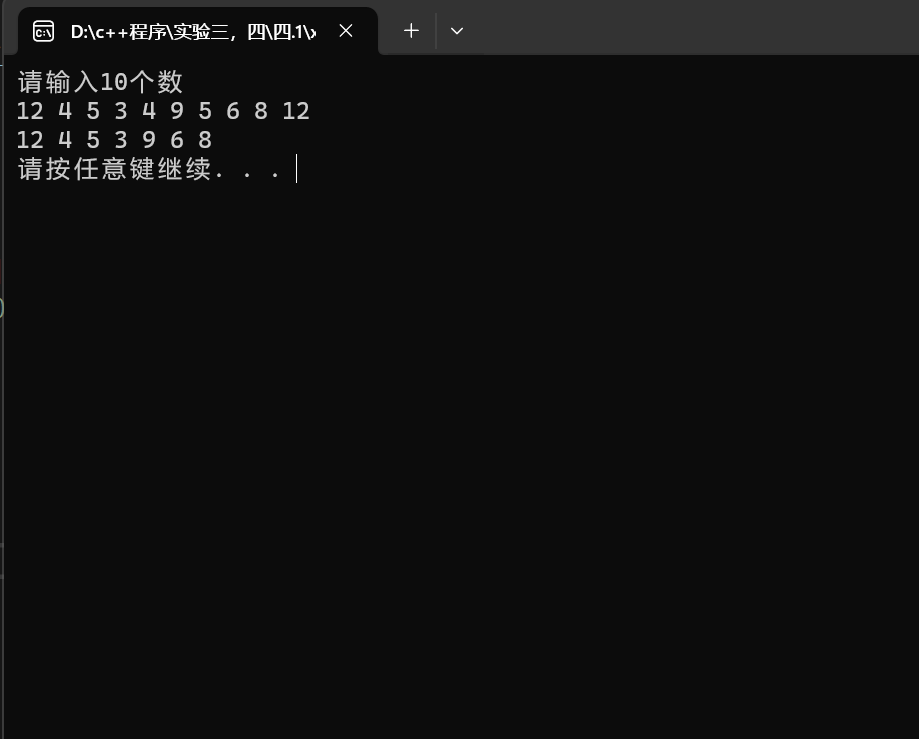
}

cout << endl;

system("pause");

return 0;

}

**结果：**

**一.2：**

**代码：**

#include<iostream>

using namespace std;

void bubble(double\* list, int listsize) {

bool changed = true;

do

{

changed = false;

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

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

{

swap (list[j] , list[j + 1]);

changed = true;

}

} while (changed);

}

int main() {

double a[10];

cout << "请输入10个数字" << endl;

for (int i = 0; i < 10; i++) {

cin >> a[i];

}

bubble(a, 10);

for (int i = 0; i < 10; i++) {

cout << a[i] <<" ";

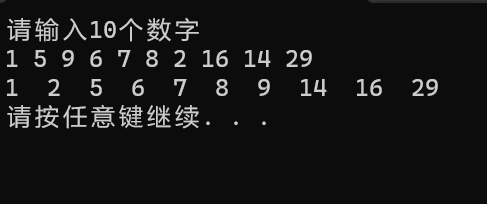
}

cout << endl;

system("pause");

return 0;

}

结果：

**一．3：**

**代码：**

#include<iostream>

using namespace std;

int main() {

bool L[100];

for (int i = 0; i < 100; ++i) {

L[i] = false;

}

for (int i = 0; i < 100; ++i) {

for (int j = i; j < 100; j = j + i+1 ) {

if (L[j] == true) {

L[j] = false;

}

else {

L[j] = true;

}

}

}

for (int i = 0; i < 100; i++) {

if (L[i] == true) {

cout << i+1 << " ";

}

else {

continue;

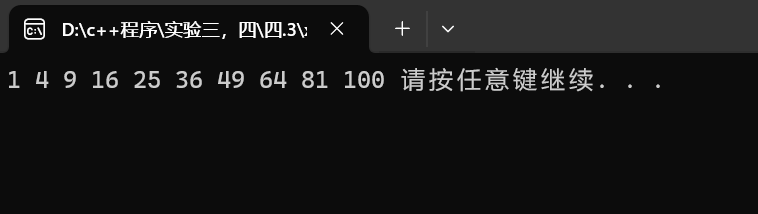
}

}

system("pause");

return 0;

}

结果：

**一.4：**

**代码：**

#include<iostream>

using namespace std;

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

for (int i = 0; i < size1 + size2; ++i) {

if (i < size1) {

list3[i] = list1[i];

}

else {

list3[i] = list2[i - size1];

}

}

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

for (int k = 0; k < size1 + size2-1- j; ++k) {

if (list3[k] > list3[k + 1]) {

int temp = list3[k];

list3[k] = list3[k + 1];

list3[k + 1] = temp;

}

}

}

cout << "The merge list is:";

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

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

}

cout << endl;

};

int main() {

int num1, num2;

cout << "请输入第一个数组元素个数和排列好的数组1（提示：请先输入元数个数）" << endl;

cin >> num1;

int\* list1 = new int[num1];

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

cin >> list1[i];

}

while (1) {

if (num1 > 80) {

cout << "数组大小过大，请重新输入数组1元素个数和各元素" << endl;

cin >> num1;

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

cin >> list1[i];

}

}

else {

cout << "成功，请继续执行下一步" << endl;

break;

}

}

cout << "请输入第二个数组元素个数和排列好的数组2（提示：请先输入元素个数）" << endl;

cin >> num2;

int\* list2 = new int[num2];

for (int i = 0; i < num2; i++) {

cin >> list2[i];

}

while (1) {

if (num2 > 80) {

cout << "数组大小过大，请重新输入数组2元素个数和元素" << endl;

cin >> num2;

for (int i = 0; i < num2; i++) {

cin >> list2[i];

}

}

else {

cout << "输入成功" << endl;

break;

}

}

int size1 = num1, size2 = num2;

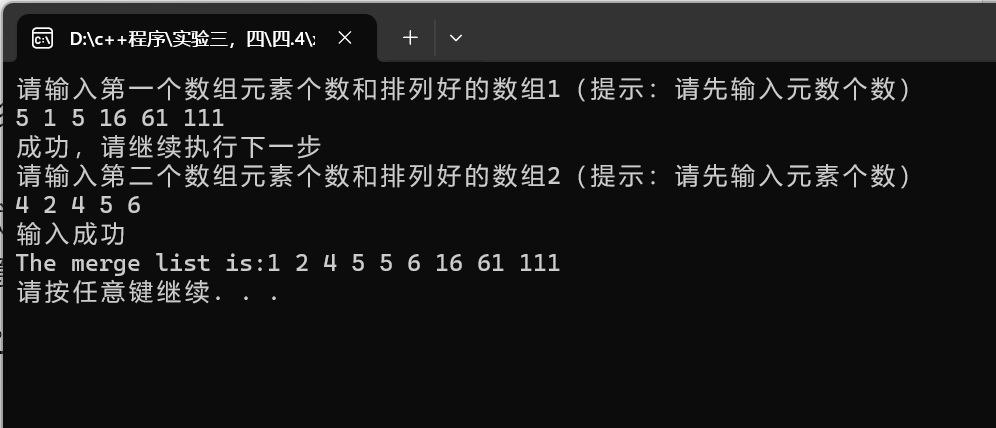
int\* list3 = new int[size1 + size2];

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

system("pause");

return 0;

}

结果：

**一．5：**

**代码：**

#include<iostream>

using namespace std;

#include<string>

#include<cstring>

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

int num = 0,stage=0,num1=0;

for (int i = 0; i<strlen(s2); ++i) {

if (s2[i] == s1[0]) {

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

if (s2[k] == ' ') {

++num1;

}

}

for (int j = 0; j < strlen(s1); ++j) {

if (s1[j] == s2[i+j]) {

++num;

}

else {

break;

}

}

if (num == strlen(s1)) {

stage = i+1-num1;

}

else {

num = 0;

}

}

}

if (num ==strlen(s1)) {

return stage;

}

else {

return -1;

}

}

int main() {

string str1,str2;

cout << "请输入字符串1" << endl;

getline(cin,str1);

char\* s1 = new char[str1.size()+1];

for (int i = 0; i < str1.size(); ++i) {

s1[i] = str1[i];

}

s1[str1.size()] = '\0';

cout << "请输入字符串2" << endl;

getline(cin, str2);

char\* s2 = new char[str2.size()+1];

for (int i = 0; i < str2.size(); ++i) {

s2[i] = str2[i];

}

s2[str2.size()] = '\0';

int ret=indexOf(s1, s2);

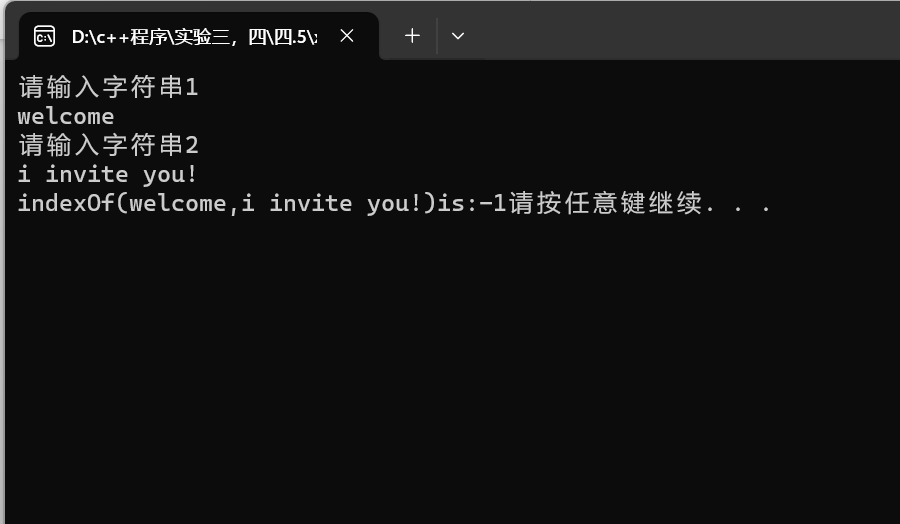
cout << "indexOf(" << str1 << "," << str2 << ")is:" << ret;

system("pause");

return 0;

}

结果：



**一.6：**

**代码：**

#include<iostream>

using namespace std;

#include<string>

#include<cstring>

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

string s1 = "abcdefghijklmnopqrstuvwxyz";

for (int i = 0; i < strlen(s); ++i) {

for (int j = 0; j < 26; j++) {

if (s[i] == s1[j]||tolower(s[i])==s1[j]) {

counts[j]++;

}

}

}

for (int i = 0; i < 26; ++i) {

if (counts[i] > 0) {

cout << s1[i] << ":" << counts[i] << "times" << endl;

}

}

}

int main() {

string str;

cout << "请输入一个字符串:" << endl;

getline(cin, str);

char\* s = new char[str.size()+1];

for (int i = 0; i < str.size(); ++i) {

s[i] = str[i];

}

s[str.size()]='\0';

int counts[26];

for (int i = 0; i < 26; i++) {

counts[i] = 0;

}

count(s, counts);

system("pause");

return 0;

}

结果：



**二．1**

**代码：**

#include<iostream>

using namespace std;

#include<string>

#include<cstring>

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

int num = 0, stage = 0, num1 = 0;

for (int i = 0; i < strlen(s2); ++i) {

if (s2[i] == s1[0]) {

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

if (s2[k] == ' ') {

++num1;

}

}

for (int j = 0; j < strlen(s1); ++j) {

if (s1[j] == s2[i + j]) {

++num;

}

else {

break;

}

}

if (num == strlen(s1)) {

stage = i + 1 - num1;

}

else {

num = 0;

}

}

}

if (num == strlen(s1)) {

return stage;

}

else {

return -1;

}

}

int main() {

string str1, str2;

cout << "请输入字符串1" << endl;

getline(cin, str1);

char\* s1 = new char[str1.size() + 1];

for (int i = 0; i < str1.size(); ++i) {

s1[i] = str1[i];

}

s1[str1.size()] = '\0';

cout << "请输入字符串2" << endl;

getline(cin, str2);

char\* s2 = new char[str2.size() + 1];

for (int i = 0; i < str2.size(); ++i) {

s2[i] = str2[i];

}

s2[str2.size()] = '\0';

int ret = indexOf(s1, s2);

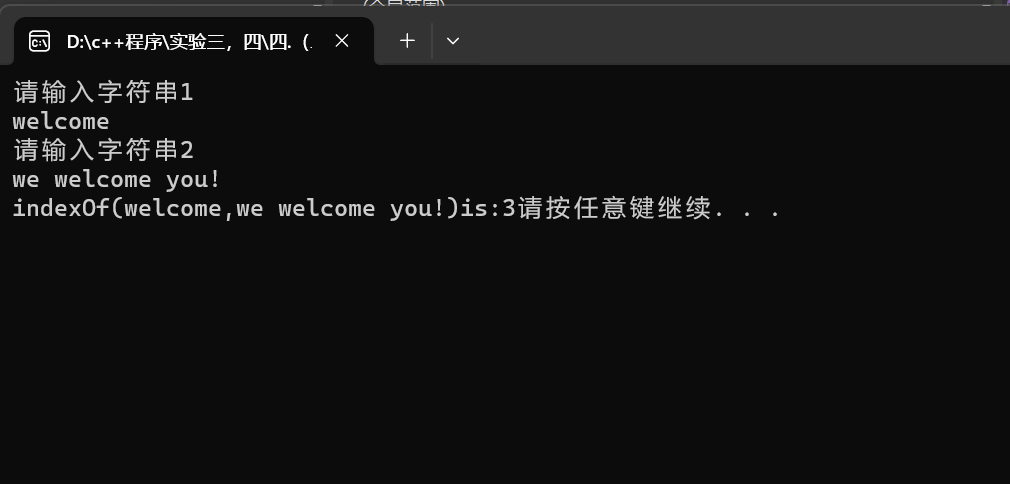
cout << "indexOf(" << str1 << "," << str2 << ")is:" << ret;

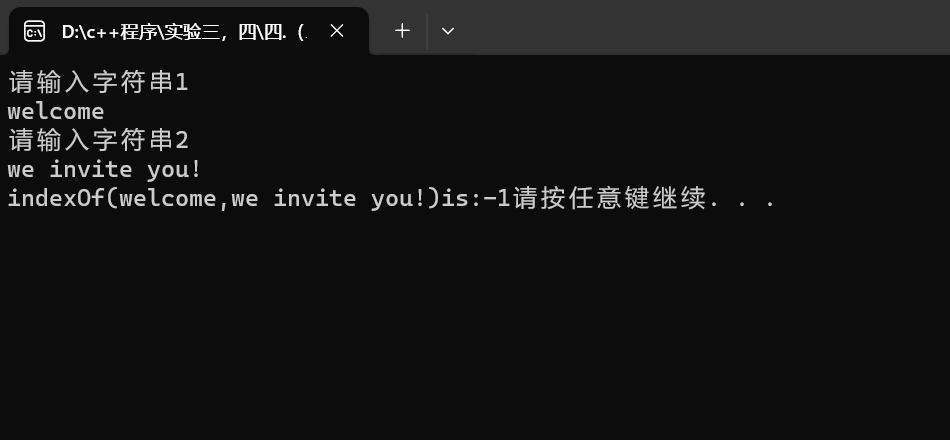
system("pause");

return 0;

}

结果:





**二．2：**

**代码：**

#include<iostream>

#include<string>

#include<cstring>

#include<cmath>

using namespace std;

int parseHex(const char\* const hexString) {

int\* decstring = new int[strlen(hexString)];

for (int i = 0; i < strlen(hexString); ++i) {

switch (hexString[i])

{

case 'A':decstring[i] = pow(16, strlen(hexString) - i-1) \* 10; break;

case 'B':decstring[i] = pow(16, strlen(hexString)-i-1) \* 11;break;

case 'C':decstring[i] = pow(16, strlen(hexString)-i-1) \* 12;break;

case 'D':decstring[i] = pow(16, strlen(hexString)-i-1) \* 13;break;

case 'E':decstring[i] = pow(16, strlen(hexString)-i-1) \* 14;break;

case 'F':decstring[i] = pow(16, strlen(hexString)-i-1) \* 15;break;

case '1':decstring[i] = pow(16, strlen(hexString)-i-1) \* 1;break;

case '2':decstring[i] = pow(16, strlen(hexString)-i-1) \* 2;break;

case '3':decstring[i] = pow(16, strlen(hexString)-i-1) \* 3;break;

case '4':decstring[i] = pow(16, strlen(hexString)-i-1) \* 4;break;

case '5':decstring[i] = pow(16, strlen(hexString)-i-1) \* 5;break;

case '6':decstring[i] = pow(16, strlen(hexString)-i-1) \* 6;break;

case '7':decstring[i] = pow(16, strlen(hexString)-i-1) \* 7;break;

case '8':decstring[i] = pow(16, strlen(hexString)-i-1) \* 8;break;

case '9':decstring[i] = pow(16, strlen(hexString)-i-1) \* 9;break;

}

}

int sum=0;

for (int i = 0; i < strlen(hexString); ++i) {

sum += decstring[i];

}

return sum;

}

int main() {

string str;

cout << "请输入一个16进制数的字符串" << endl;

cin >> str;

char\* hexString = new char[str.size() + 1];

for (int i = 0; i < str.size(); ++i) {

hexString[i] = str[i];

}

hexString[str.size()] = '\0';

int ret = parseHex(hexString);

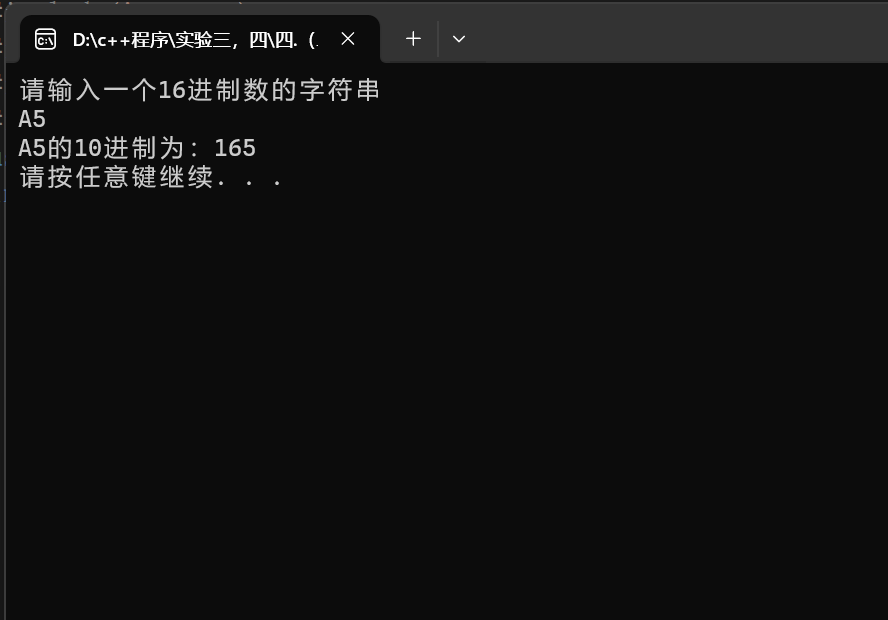
cout << str << "的10进制为：" << ret << endl;

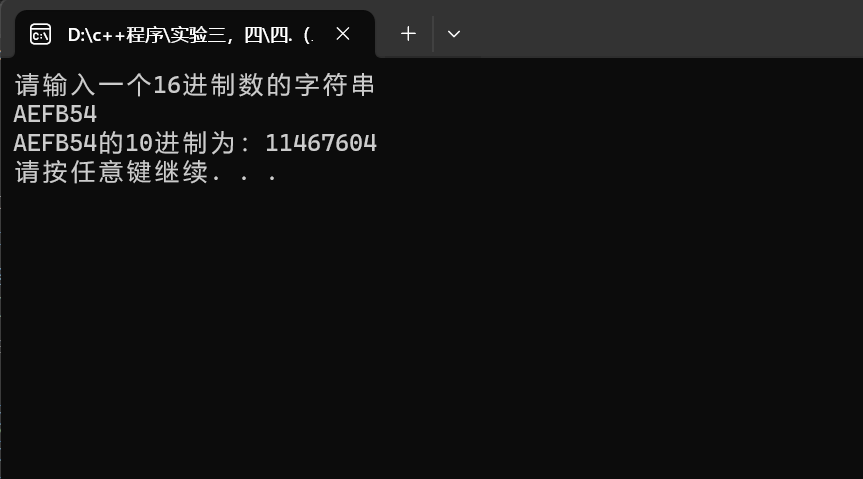
system("pause");

return 0;

}

结果：





**二．3：**

**代码：**

#include<iostream>

using namespace std;

void comparison(int\* s1, int num) {

for (int i = 0; i < num-1; ++i) {

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

if (s1[j] > s1[j + 1]) {

int temp = s1[j];

s1[j] = s1[j + 1];

s1[j + 1] = temp;

}

}

}

}

int main() {

int num;

cout << "请输入数组元素个数及各元素的值" << endl;

cin >> num;

int\* s1 = new int[num];

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

cin >> s1[i];

}

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

cout << \*s1 << " " << s1 << endl;

s1++;

}

s1 = s1 - num;

comparison(s1, num);

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

cout << \*(s1+i) << " ";

}

cout << endl;

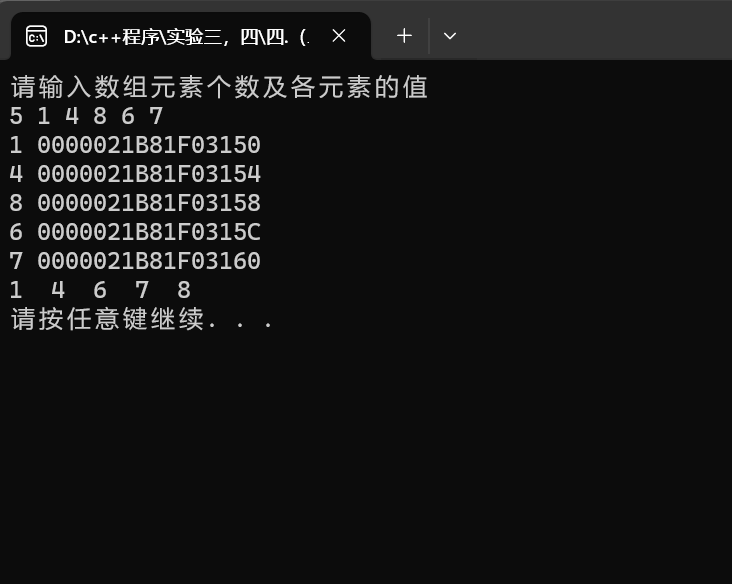
delete[]s1;

system("pause");

return 0;

}

结果：



**四、遇到的问题与解决方法**

**五、体会**