**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？

函数是没有类型的，声明中在函数前面添加类型，是为了识别返回值的类型，以便编译程序把返回值按照指定类型来处理。所以，函数中返回值的类型只能与函数声明时的类型一致。

2. 本实验中主函数调用函数时采用的是何种传递方式？

利用参数传递

**四、算法分析，程序结果**

**项目1：**

#include<iostream>

using namespace std;

int yue(int a, int b) {

int c;

if (a < b) {

c = a;

a = b;

b = c;

}

while (b != 0) {

c = a % b;

a = b;

b = c;

}

return a;

}

int bei(int a, int b) {

int d;

int e;

int f;

d = yue(a, b);

e = a \* b;

f = e / d;

return f;

}

int main() {

int a;

int b;

cin >> a >> b;

cout<<"最大公约数为：" << yue(a, b)<<endl;

cout << "最小公倍数为：" << bei(a, b);

}

输入24,56观察结果

**项目2：**

#include<iostream>

using namespace std;

bool is\_prime(int m)

{

int n;

for (n = 2; n < m; n++)

{

if (m % n == 0)

{

return false;

}

}

return true;

}

void fun(int s=200)

{

int c = 1, d = 2;

while (c <= s)

{

if (is\_prime(d) == 1)

{

if (c % 10 == 0)

{

cout << '\t' << d << endl;

}

else

{

cout << '\t' << d;

}

c++;

}

d++;

}

}

int main()

{

int x, y;

cout << "请输入数字:";

cin >> x;

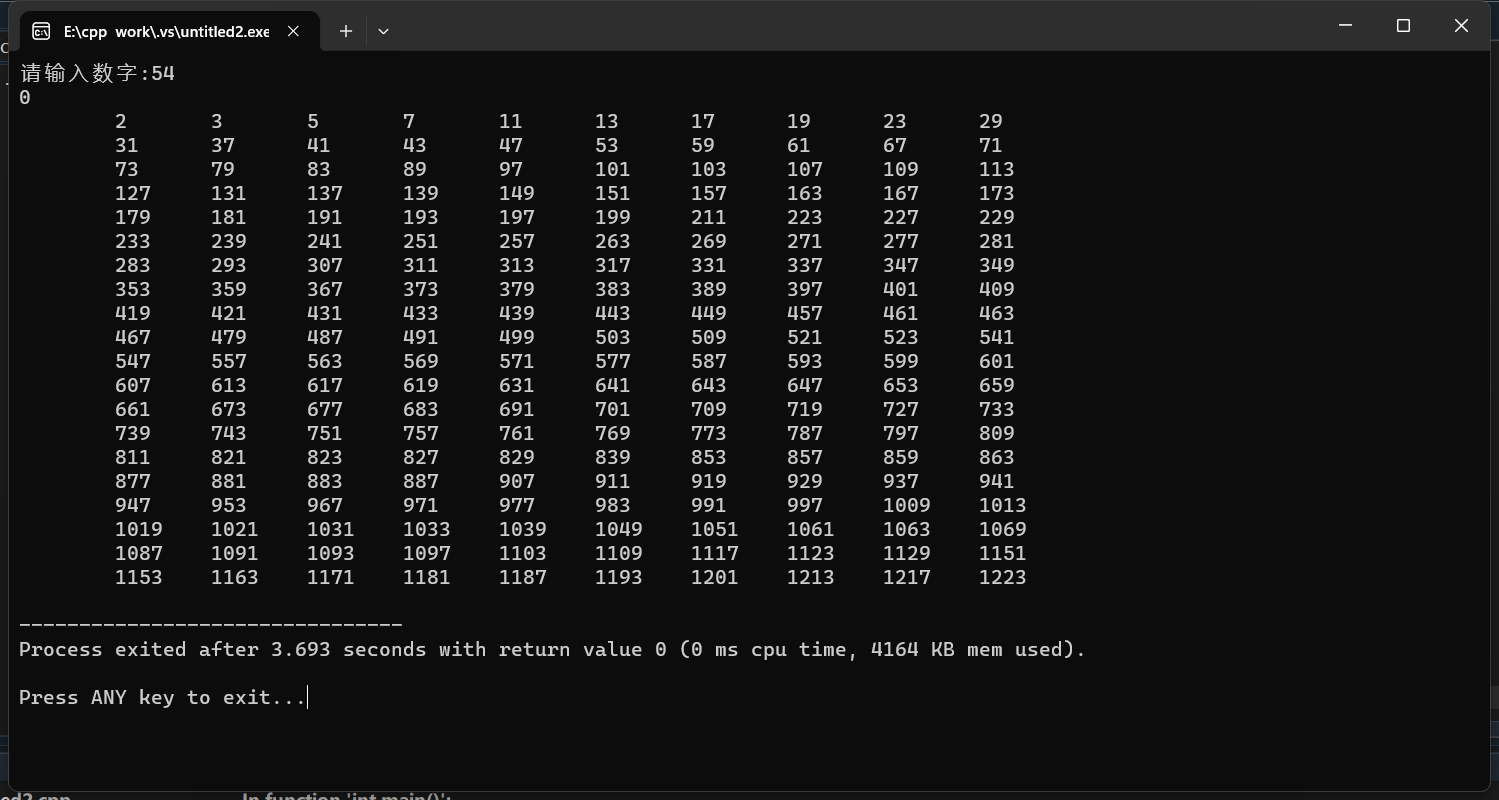
cout << is\_prime(x) << endl;

fun();

return 0;

}

**以输入54为例**



**项目3：**

**头文件：**

double celsius\_to\_fah(double cel);

double fahrenheit\_to\_cels(double fah);

**源文件：**

#include<iostream>

double celsius\_to\_fah(double cel) {

double a;

a = (cel \* 9) / 5 + 32;

return a;

}

double fahrenheit\_to\_cels(double fah) {

double b;

b = (fah - 32) \* 5 / 9;

return b;

}

**测试程序：**

#include<iostream>

#include<iomanip>

#include"mytemperature.h"

using namespace std;

int main() {

double a, b;

cout << "cel to fah:";

cin >> a;

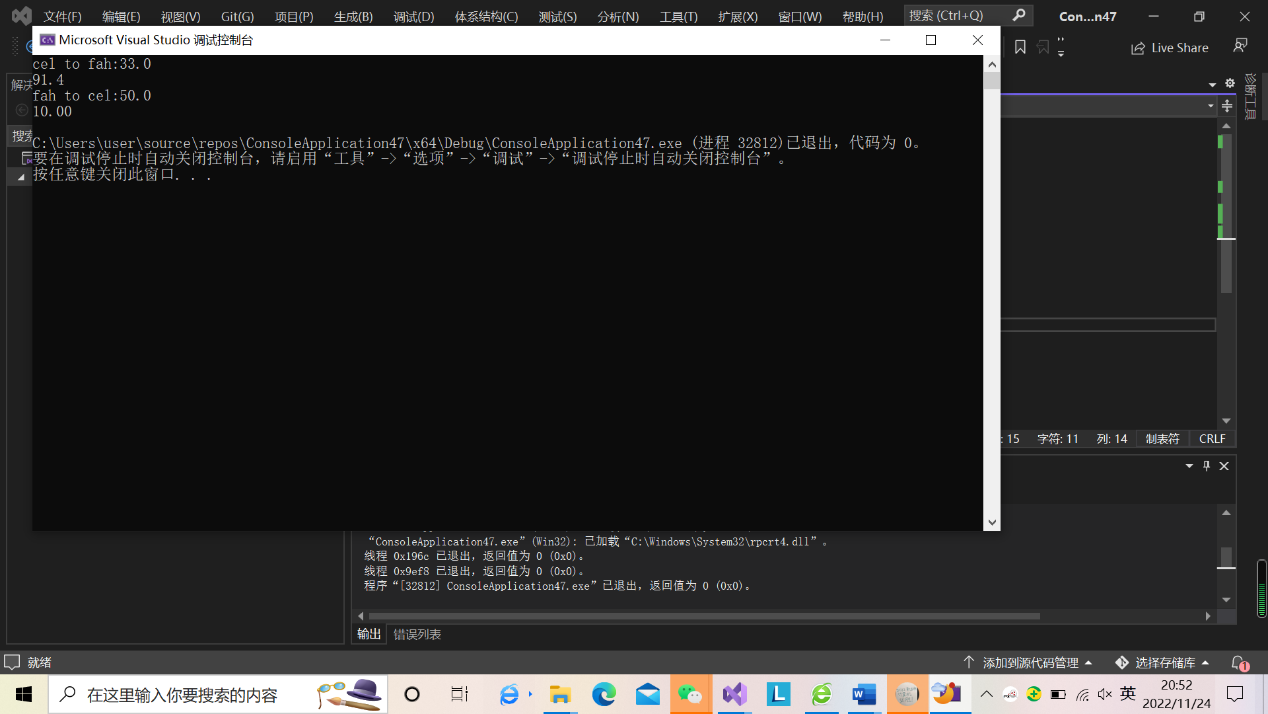
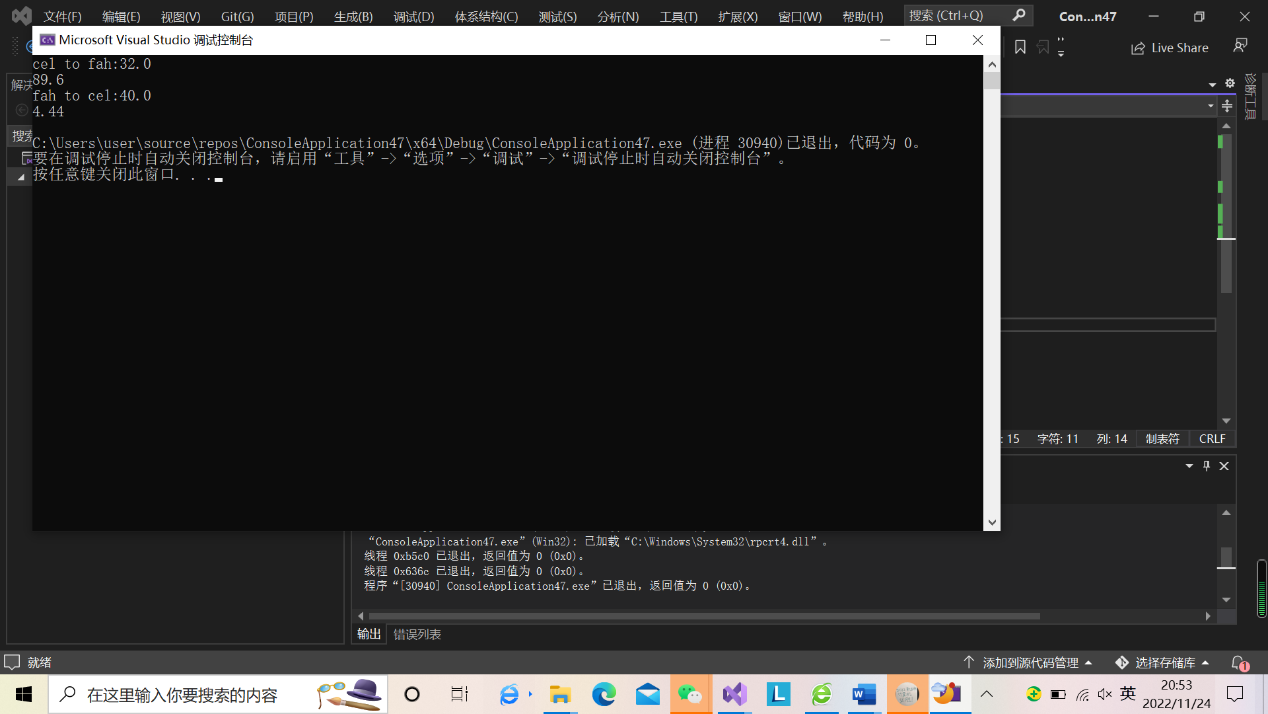
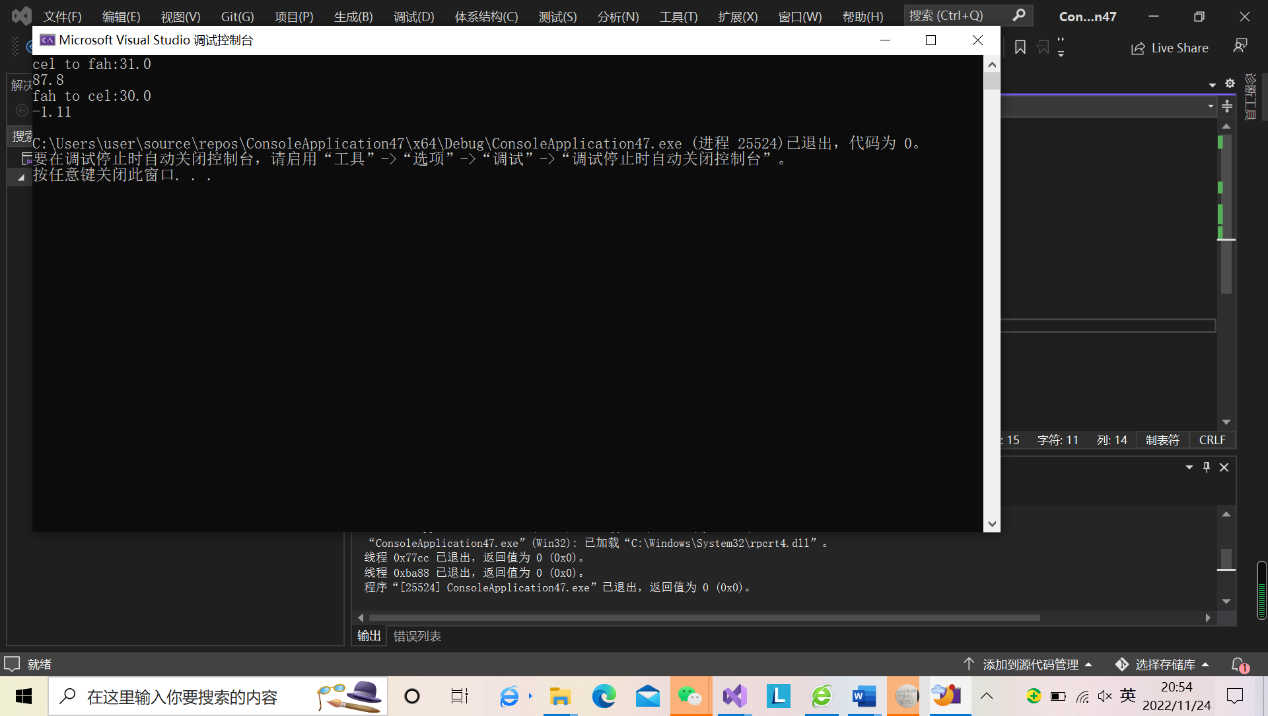
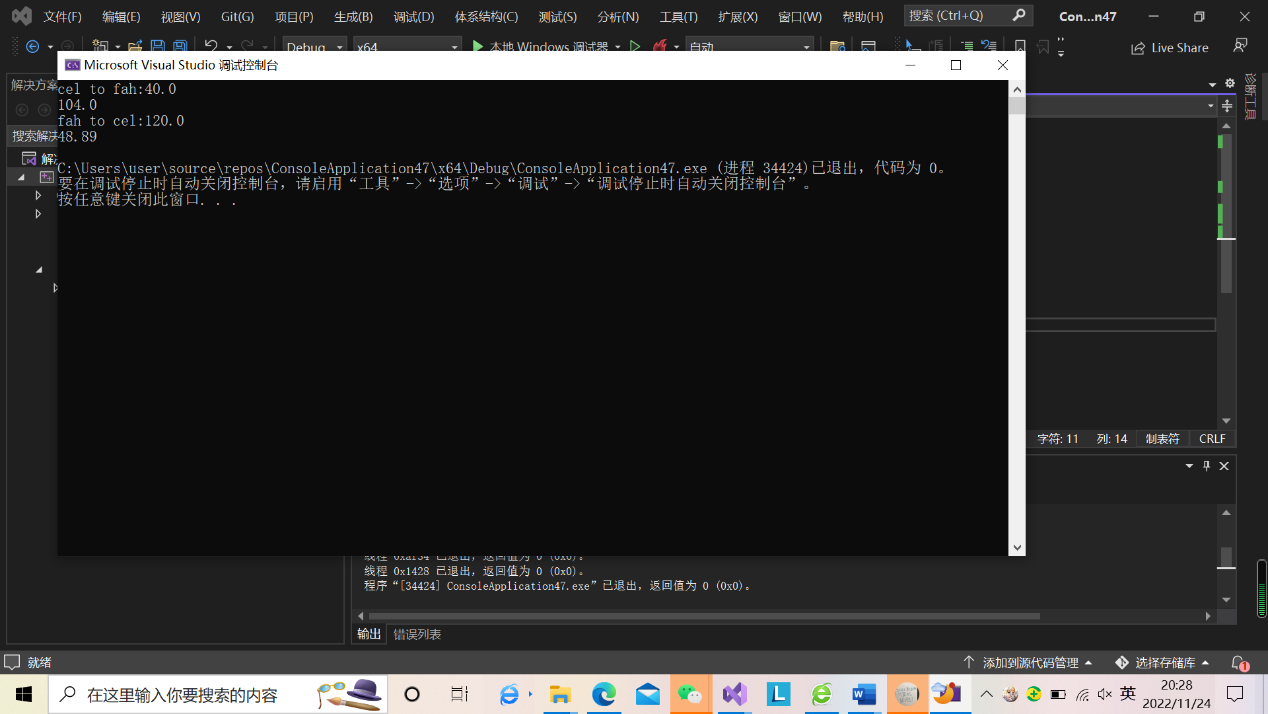
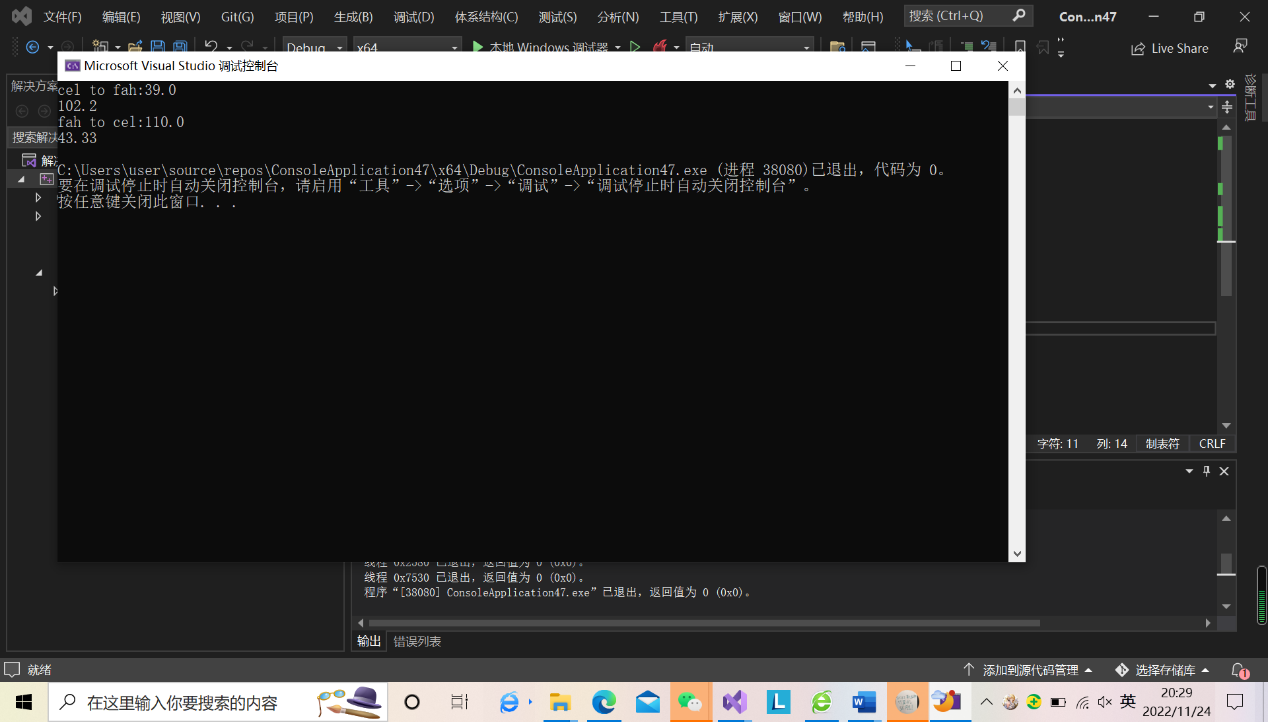
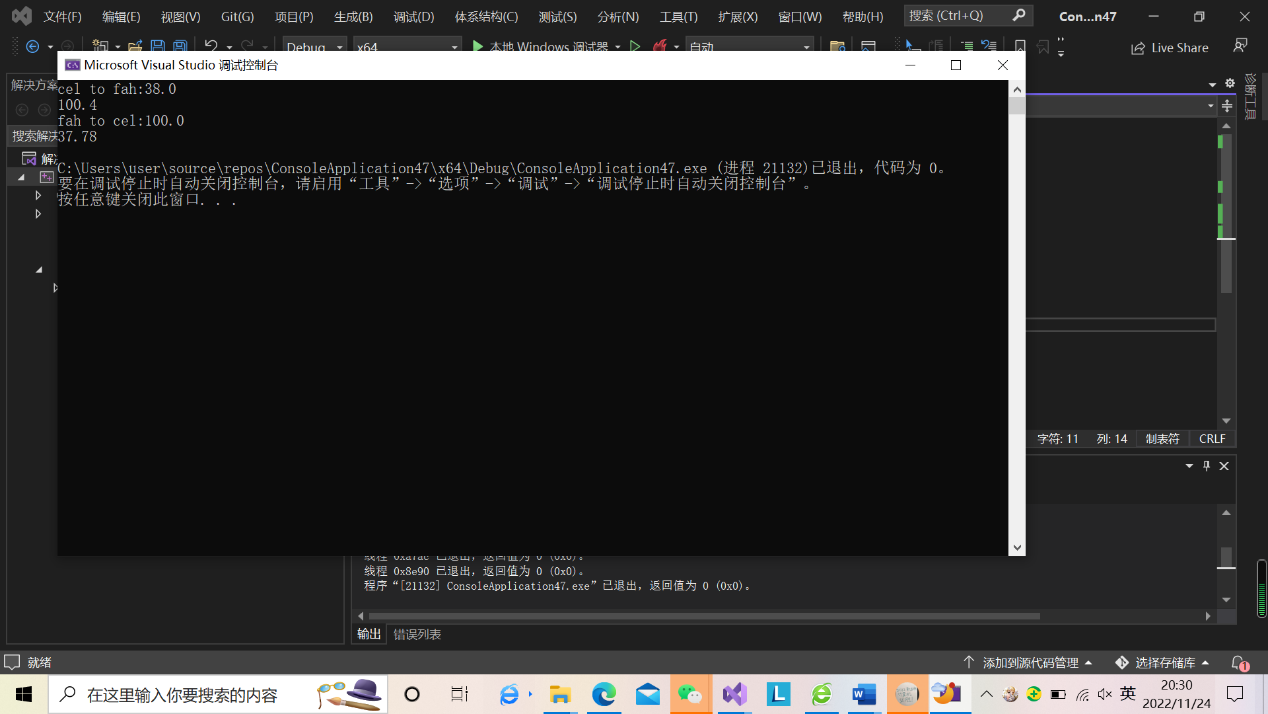
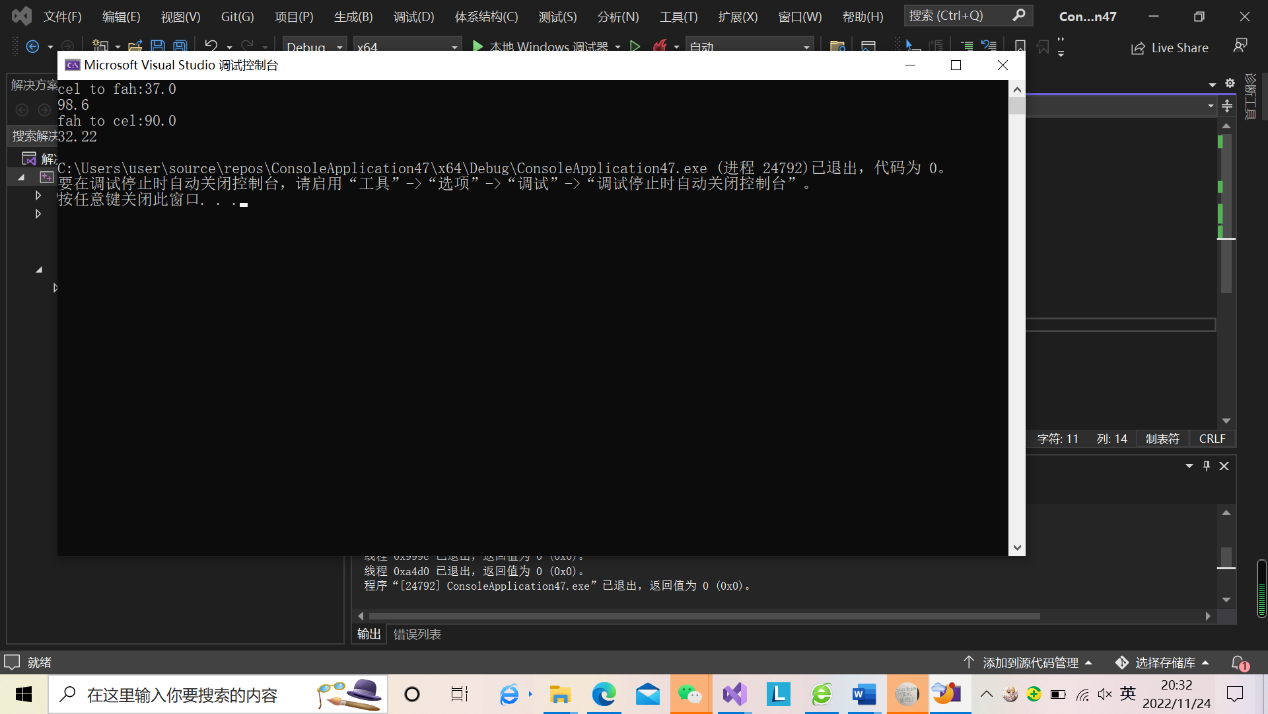
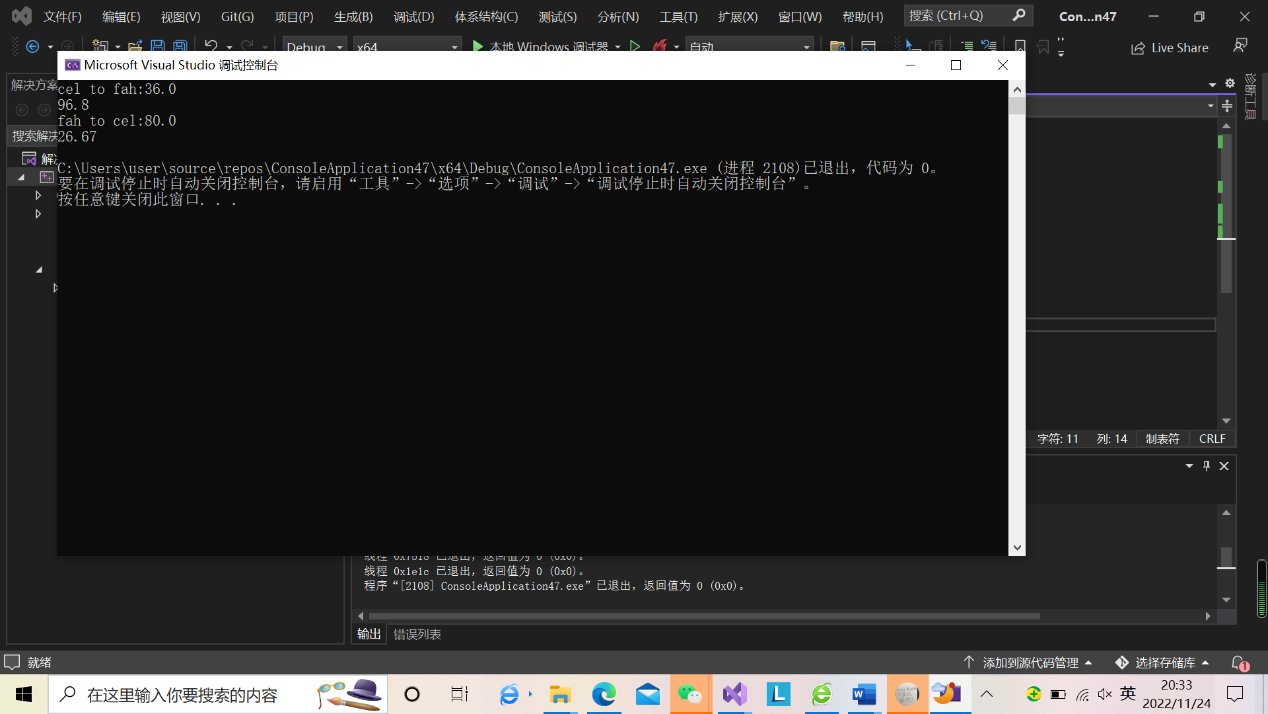
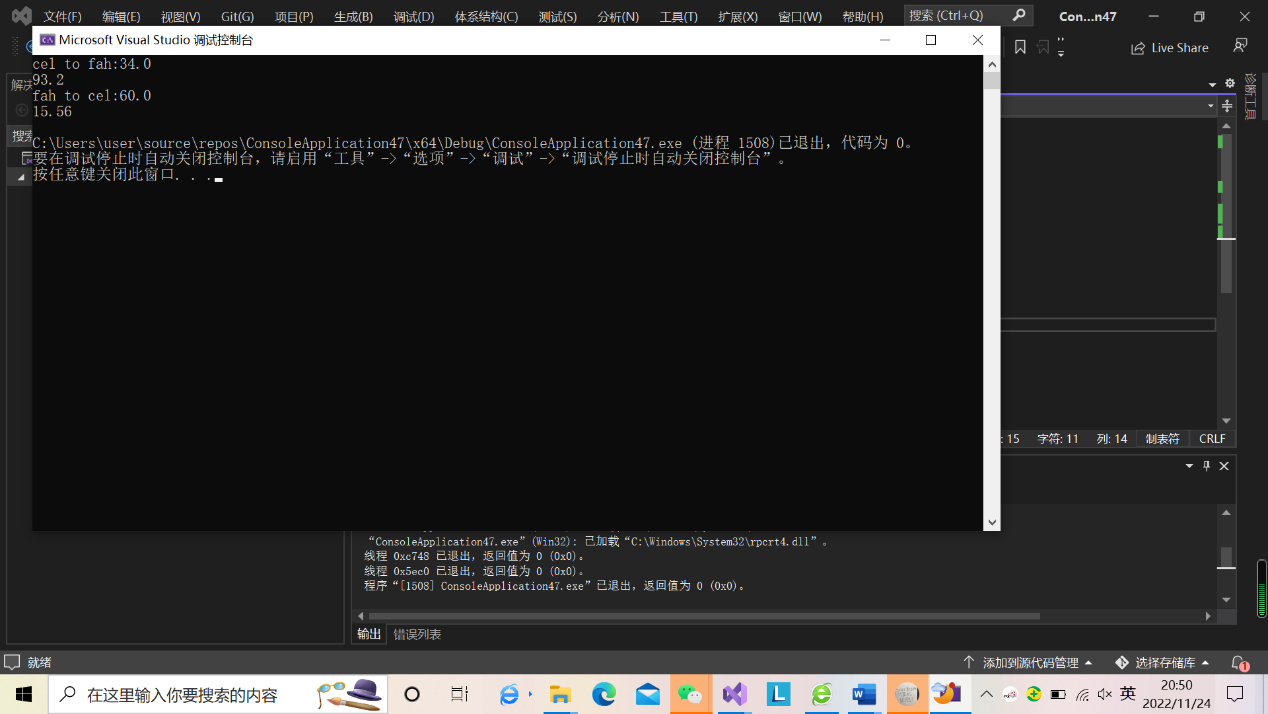
cout << fixed << setprecision(1) << celsius\_to\_fah(a) << endl;

cout << "fah to cel:";

cin >> b;

cout << fixed << setprecision(2) << fahrenheit\_to\_cels(b) << endl;

return 0;

}

**项目5：**

#include<iostream>

using namespace std;

int main() {

int a = 1;

for (int i = 0; i < 9; i++) {

int b;

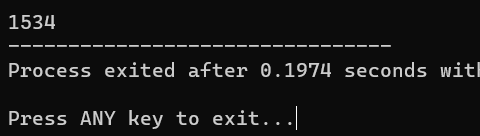
b = 2 \* (a + 1);

a = b;

}

cout << a;

}



**五、遇到的问题与解决方法**

在编写第三个程序的时候，读完题目要求是要有三个文件，分别是头文件.h，源文件.cpp和测试文件含main函数的.cpp。但是我误当成了两个文件，将源文件的内容也写在了头文件中，经过同学的提醒才分成三个文件。在引用mytemperature.h头文件时将””用成了<>，经过向同学询问后修改了过来。

**六、体会**

程序如何编写并非难点，但如何想到用这样的方式实现，是一个很难也很重要的问题。有了算法思路，才有可能通过代码实现。在写求最大公约数的时候，有同学跟我讲解了3种不同的办法，让我深受启发，程序的编写需要广泛的阅读和练习。也让我意识到我们要积极与同学或者其他优秀的人交流，这也是学习编程的重要一环，我们可以从他们那学习到宝贵的经验和见解，未来改进自己的程序。

**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**

**三、算法分析，程序结果**

**（一）数组**

**项目1：**

include<iostream>

using namespace std;

int main() {

int a[10];

cout<< "please input ten numbers:";

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

cin >> a[i];

}

cout << a[0];

for (int i = 1; i < 10; i++)

{

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

{

if (a[p] != a[i])

{

if (p == i - 1)

{

cout << a[i];

}

}

else

{

break;

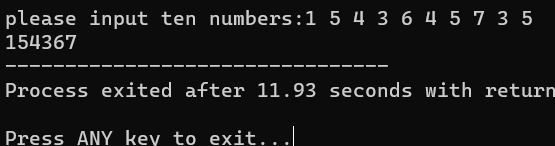
}

}

}

return 0;

}

输入1 5 4 3 6 4 5 7 3 5，观察结果：

项目2：

#include<iostream>

using namespace std;

void sort(int n, double m[])

{

bool changed = true;

int a;

do

{

changed = false;

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

{

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

{

a = m[j + 1];

m[j + 1] = m[j];

m[j] = a;

changed = true;

}

}

} while (changed == true);

}

int main()

{

double a[10];

int i;

cout << "please input ten numbers:" << endl;

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

{

cin >> a[i];

}

sort(10, a);

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

{

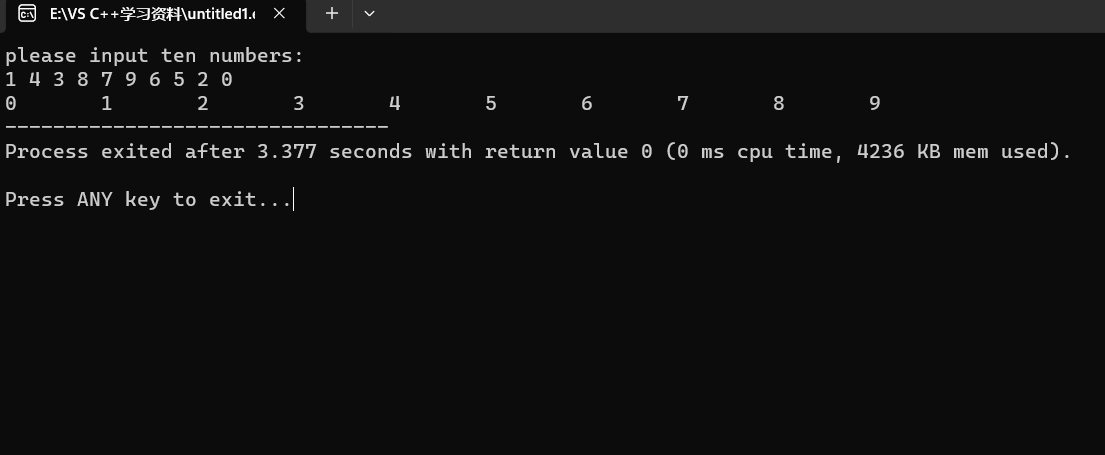
cout << a[i] << '\t';

}

return 0;

}

输入1 4 3 8 7 9 6 5 2 0，观察输出结果：



**项目3：**

#include<iostream>

using namespace std;

int main()

{

bool a[100];

const int s = 100;

int y;

for (int z = 0; z < 100; z++)

{

a[z] = 0;

}

for (int i = 1; i <= s; i++)

{

for (int x = i - 1; x < 100; x = x + i)

{

if (a[x] == 0)

{

a[x] = 1;

}

else

{

a[x] = 0;

}

}

}

for (y = 0; y < 100; y++)

{

if (a[y] == 1)

{

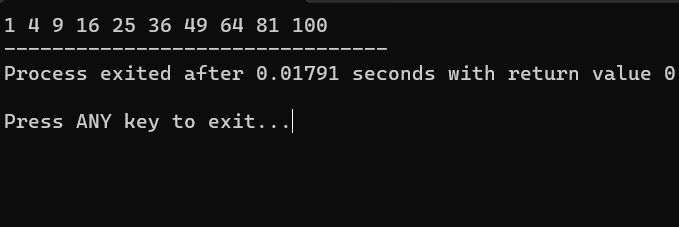
cout << y + 1 << " ";

}

}

return 0;

}



**项目4：**

#include<iostream>

using namespace std;

void merge(const int l1[], int s1, const int l2[], int s2, int l3[])

{

int i, j;

int k;

int\* p = &i;

for (i = 0; i < s1; i++)

{

l3[i] = l1[\*p];

}

for (i = s1; i < s1 + s2; i++)

{

l3[i] = l2[\*p - s1];

}

for (i = 0; i < s1 + s2; i++)

{

for (j = 0; j < s1 + s2; j++)

{

if (l3[i] < l3[j])

{

k = l3[i];

l3[i] = l3[j];

l3[j] = k;

}

}

}

for (int i = 0; i < s1 + s2; i++)

{

cout << l3[i] << " ";

}

}

int main()

{

int s1, s2;

cout << "how many numbers are there in the list1:";

cin >> s1;

int\* l1 = new int[s1];

cout << "please input list1:";

for (int i = 0; i < s1; i++)

{

cin >> l1[i];

}

cout << "how many numbers are there in the list2:";

cin >> s2;

int\* l2 = new int[s2];

cout << "please input list1:";

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

{

cin >> l2[i];

}

int\* l3 = new int[s1 + s2];

merge(l1, s1, l2, s2, l3);

delete[]l1;

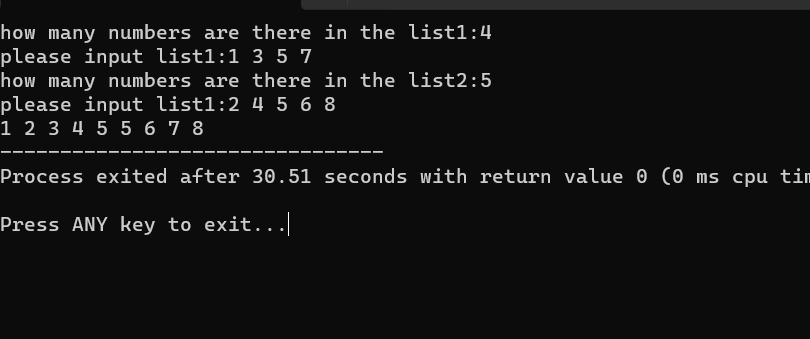
delete[]l2;

delete[]l3;

return 0;

}

数组一输入长度4，内容为1 3 5 7；数组二输入长度5，内容为2 4 5 6 8



**项目5：**

#include<iostream>

#include<cstring>

using namespace std;

int indexOf(const char str1[], const char str2[])

{

int a, b;

a = strlen(str1);

b = strlen(str2);

if (a > b)

{

return -1;

}

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

{

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

{

if (q == a - 1 && str1[q] == str2[i + q]) {

return i;

}

if (str1[q] != str2[i + q])

{

break;

}

}

}

return -1;

}

int main()

{

const int size = 999;

char str1[size], str2[size];

cout << "Enter the first string:" << endl;

cin.getline(str1, size);

cout << "Enter the second string:" << endl;

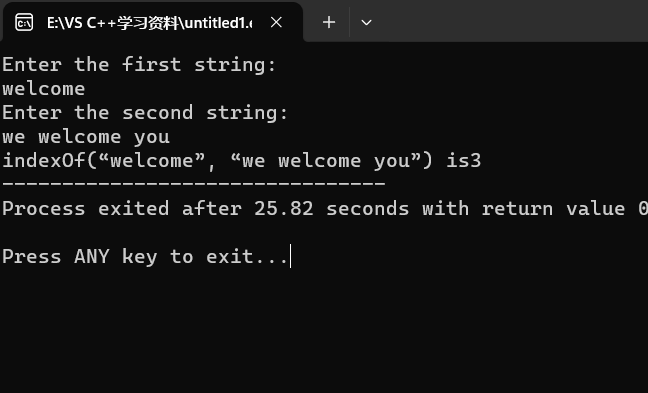
cin.getline(str2, size);

cout << "indexOf(“" << str1 << "”, “" << str2 << "”) is" << indexOf(str1, str2);

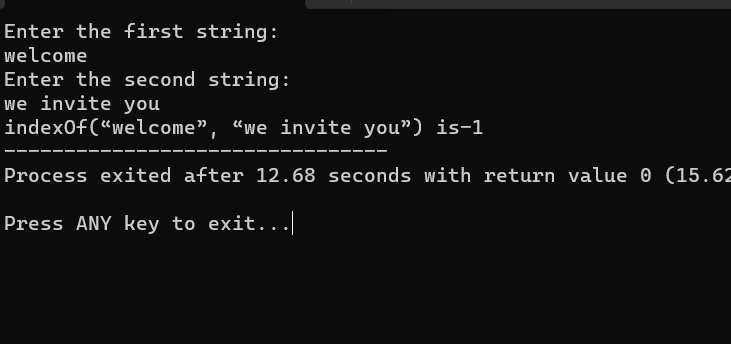
return 0;

}

字符串一输入welcome，字符串二输入we welcome you，观察输出结果：



字符串一输入welcome，字符串二输入we invite you，观察输出结果：



**项目6：**

#include<iostream>

using namespace std;

const int a = 999;

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

{

int k = 0, j = 0;

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

{

counts[i] = 0;

}

for (k = 0; k < a; k++)

{

if ((s[k] <= 90) && (s[k] >= 65))

{

counts[s[k] - 65] += 1;

}

else if ((s[k] <= 122) && (s[k] >= 97))

{

counts[s[k] - 97] += 1;

}

}

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

{

if (counts[j] != 0)

{

int m = j + 97;

cout << char(m) << ":" << counts[j] << "times" << endl;

}

}

}

int main()

{

char s[a];

cout << "Enter a string:" << endl;

cin.getline(s, a);

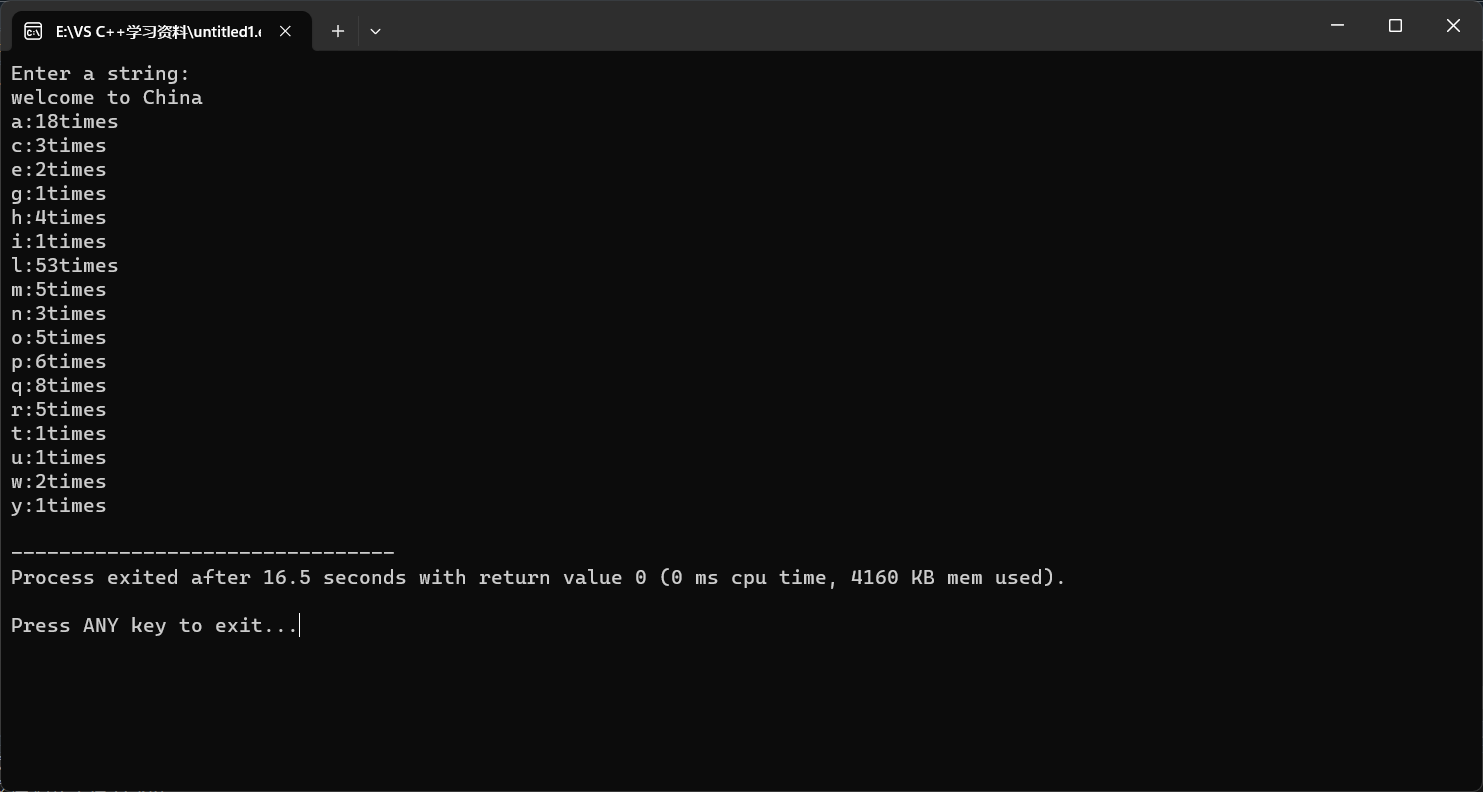
int counts[26];

count(s, counts);

return 0;

}

输入welcome to China，观察输出结果



**（二）指针：**

**项目1:**

#include<iostream>

#include<cstring>

using namespace std;

int indexof(const char\* s1, const char\* s2)

{

int a, b;

a = strlen(s1);

b = strlen(s2);

if (a > b)

{

return -1;

}

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

{

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

{

if (p == a - 1 && s1[p] == s2[i + p])

{

return i;

}

if (s1[p] != s2[i + p])

{

break;

}

}

}

return -1;

}

int main()

{

const int size = 999;

char l1[size], l2[size];

cout << "enter the first string:";

cin.getline(l1, size);

cout << "enter the second string:";

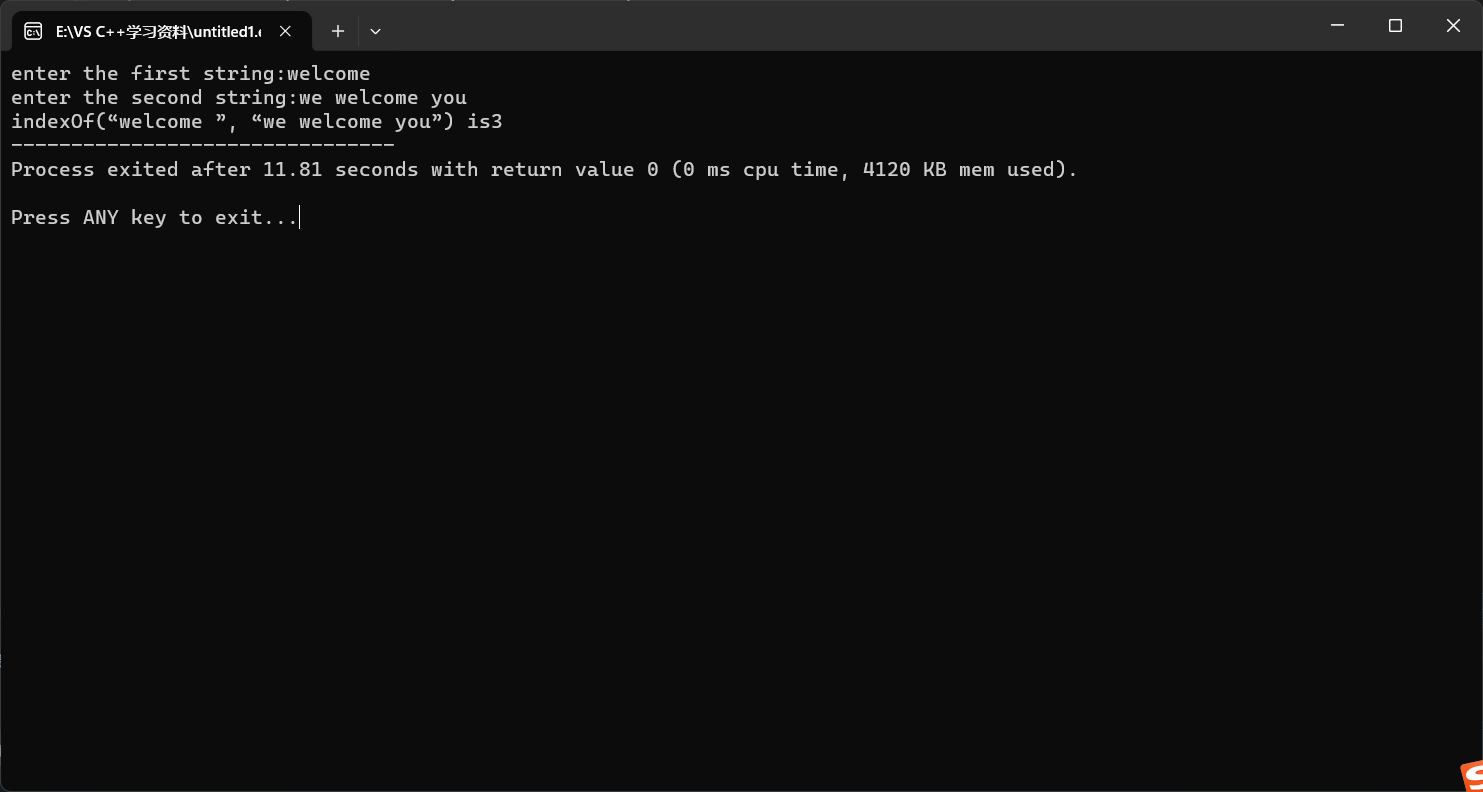
cin.getline(l2, size);

cout << "indexOf(“" << l1 << "”, “" << l2 << "”) is" << indexof(l1, l2);

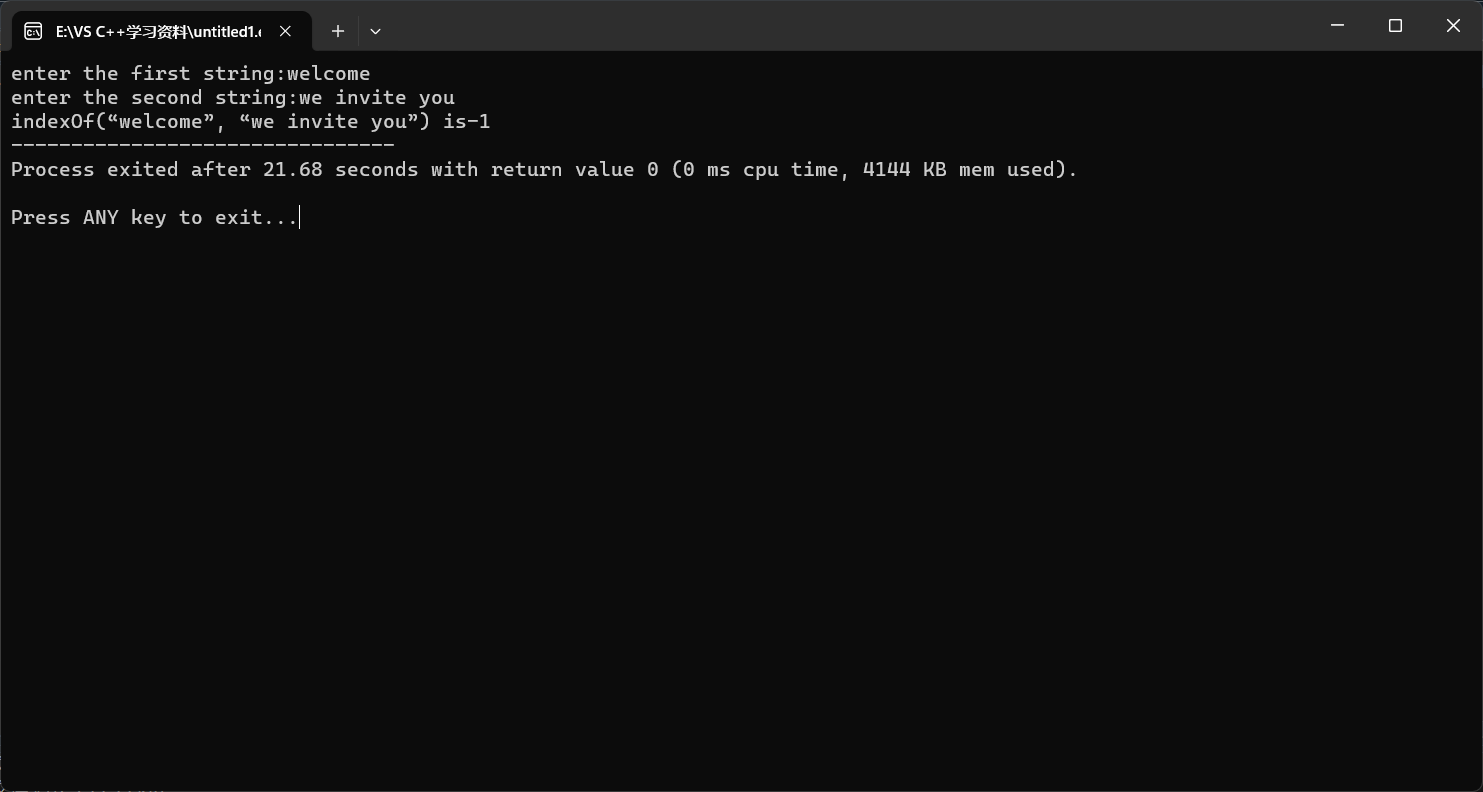
return 0;

}

输入welcome，数组二输入we welcome you，观察输出结果：



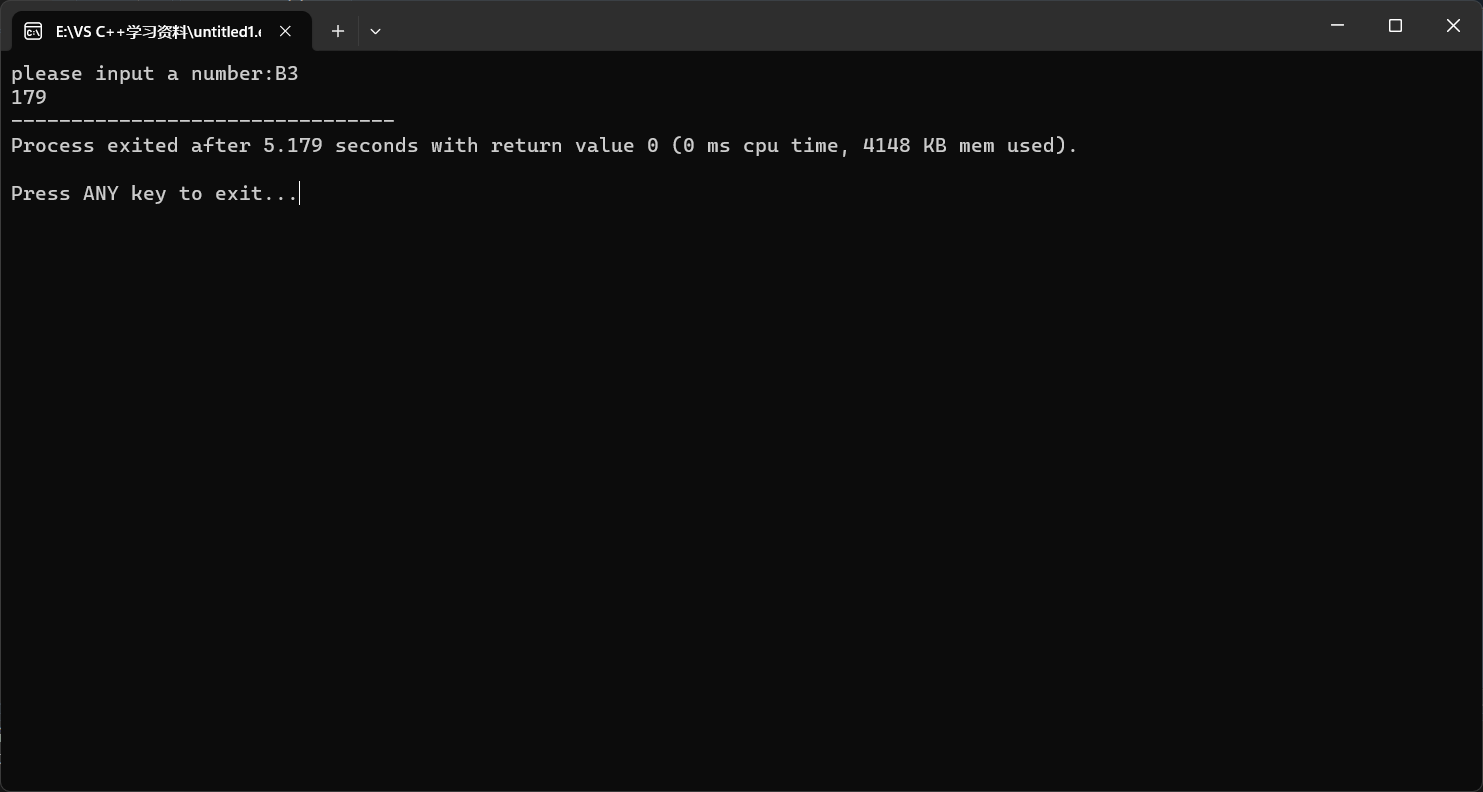
输入welcome，数组二输入we invite you，观察输出结果：



**项目2：**

#include**<**iostream**>**#include**<**cstring**>**#include**<**math**.**h**>  
using** **namespace** **std;  
int** parseHex**(const** **char\*** **const** hexString**)  
{**    **int** a**;**    a **=** strlen**(**hexString**);**    **int** sum **=** 0**,** n **=** 0**;**    **for** **(int** i **=** 0**;** i **<** a**;** i**++)**    **{**        **if** **(**hexString**[**i**]** **>=** 65 **&&** hexString**[**i**]** **<=** 70**)**        **{**            n **=** **(**hexString**[**i**]** **-** 55**)** **\*** pow**(**16**,** a **-** i **-** 1**);**        **}**        **else**        **{**            n **=** **(**hexString**[**i**]** **-** 48**)** **\*** pow**(**16**,** a **-** i **-** 1**);**        **}**        sum **=** sum **+** n**;**    **}**    **return** sum**;  
}  
int** main**()  
{**    **const** **int** size **=** 999**;**    **char** s**[**size**];**    cout **<<** **"please input a number:";**    cin**.**getline**(**s**,** size**);**    cout **<<** parseHex**(**s**);**    **return** 0**;  
}**

输入B3，观察输出结果：



**项目3：**

#include<iostream>

using namespace std;

int a;

void fun(int s[])

{

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

{

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

{

if (s[k] < s[h])

{

int t;

t = s[h];

s[h] = s[k];

s[k] = t;

}

}

}

}

int main()

{

int b;

cout << "please input the size of the array:";

cin >> a;

int\* arr = new int[a];

cout << "please input the details of the array:";

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

{

cin >> \*(arr + i);

}

cout << "请选择你要查看数组中的项目：";

cin >> b;

cout << arr[b - 1] << endl;

fun(arr);

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

{

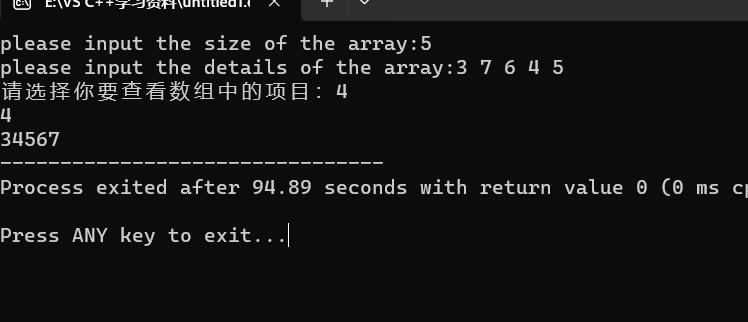
cout << \*(arr + m);

}

delete[]arr;

return 0;

}



**四、遇到的问题与解决方法**

在对用const修饰过的数组赋值时，一开始直接使用cin总是报错，后来上网搜索相关内容才知道，需要使用动态数组new。经过上网的搜索，我找到了一个可以以‘\n’为终止符的输入流函数getline，把它编入程序中。

**五、体会**

再引入数组和指针之后，c++的编写变得更为严格和繁琐，有诸多细节需要我们注意，使得编写难度升高，但是却能简化程序，节省运算时间，提高效率，因而我们要多尝试编写时利用数组或者指针，以求熟练掌握这一技巧。对于指针，它可谓目前学到的最繁琐和抽象的内容，在写指针相关的题时我在书和网上都搜索了大量资料，了解了很多自己从前不知道的知识，最终得以完成实验。在学习编程的过程中，学会搜集相关资料的重要性可见一斑。