**计算机程序设计基础（C++)**

**实验报告**

专业班级： 软工2301班

学 号： 8209230115

姓 名： 李雨顺

**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？

2. 本实验中主函数调用函数时采用的是何种传递方式？

1答：函数本身没有类型，是为了识别返回值的类型，才在函数前加入这个类型，以便编译程序把返回值以这个特定类型来处理。

2 答：值传递

**四、算法分析，程序结果**

1、（1）代码：

#include<iostream>

using namespace std;

int maxgongyueshu(int a, int b)//定义求公约数的函数

{

int z;

z=(a<b) ? a : b;

for(;;z--)//利用for循环求最大公约数

{

if ((a % z == 0) && (b % z == 0))break;

}

return z;

}

int main()

{

cout << "请输入两个自然数：";

int m, n;

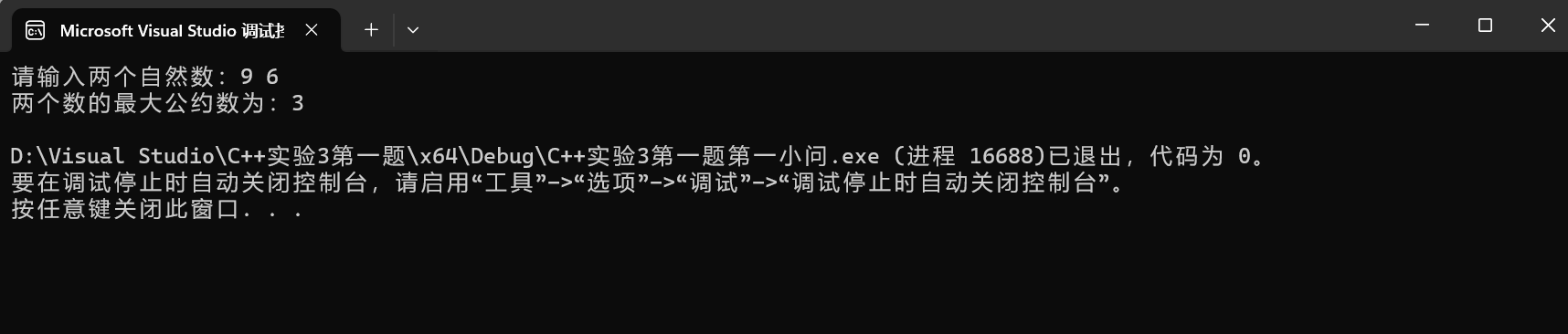
cin >> m >> n;//输入两个数

if (m == 0 || n == 0)cout << "不存在最大公约数。"<<endl;//0不存在最大公约数，故单列出来一种情况

else cout << "两个数的最大公约数为：" << maxgongyueshu(m, n) << endl;//输出最大公约数

return 0;

}

结果截图：

1、（2）代码：

#include<iostream>

using namespace std;

void maxandmin(int a, int b, int& m, int& n)//利用引用形参

{

m = (a < b) ? a : b;

for (;; m--)

{

if ((a % m == 0) && (b % m == 0))break;

}//利用for循环求出最大公约数。

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

{

if ((i % a == 0) && (i % b == 0))break;

n = i + 1;

}//利用for循环求出最小公倍数。

}

int main()

{

int m, n, x, y;

cout << "请输入两个自然数：";

cin >> m >> n;//输入两个数

if ((m == 0) || (n == 0))cout << "没有最大公约数和最小公倍数。";//因为0没有最大公约数与最小公倍数，故单列一个情况出来

else

{

maxandmin(m, n, x, y);

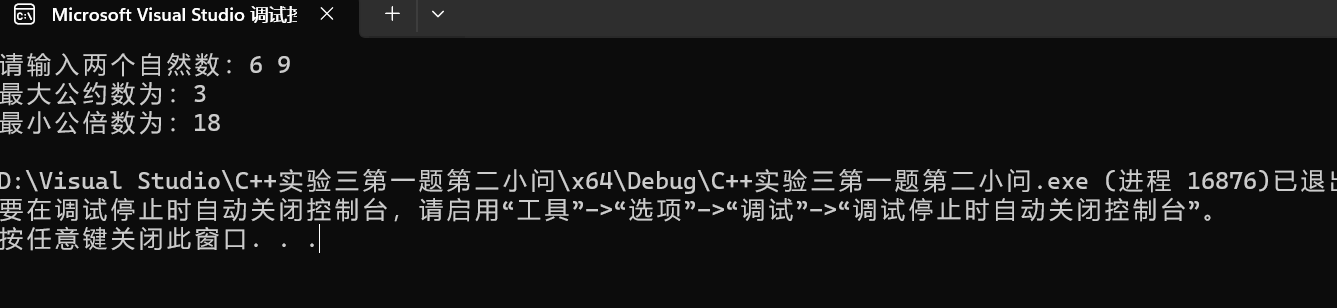
cout << "最大公约数为：" << x << endl;//输出最大公约数

cout << "最小公倍数为：" << y << endl;//输出最小公倍数

}

return 0;

}

结果截图：

2、代码：

#include<iostream>

using namespace std;

bool is\_prime(int a);//声明函数

bool is\_prime(int a)//定义函数判断一个数是否是素数

{

int b = 0;

for(int i=2;i<a;i++)//通过for循环找出除1和它本身的因数，若有，则不是素数

{

if (a % i == 0)

{

b = i;

break;

}

}

if (b)return false;

if (a <= 1) return false;//0和1都不是素数，故单列一种情况

else return true;

}

int main()

{

int num1 = 0;

for(int i=0;num1<200;i++)//利用for循环找出前两百的素数

{

if (is\_prime(i))

{

num1++;

cout << i << '\t';

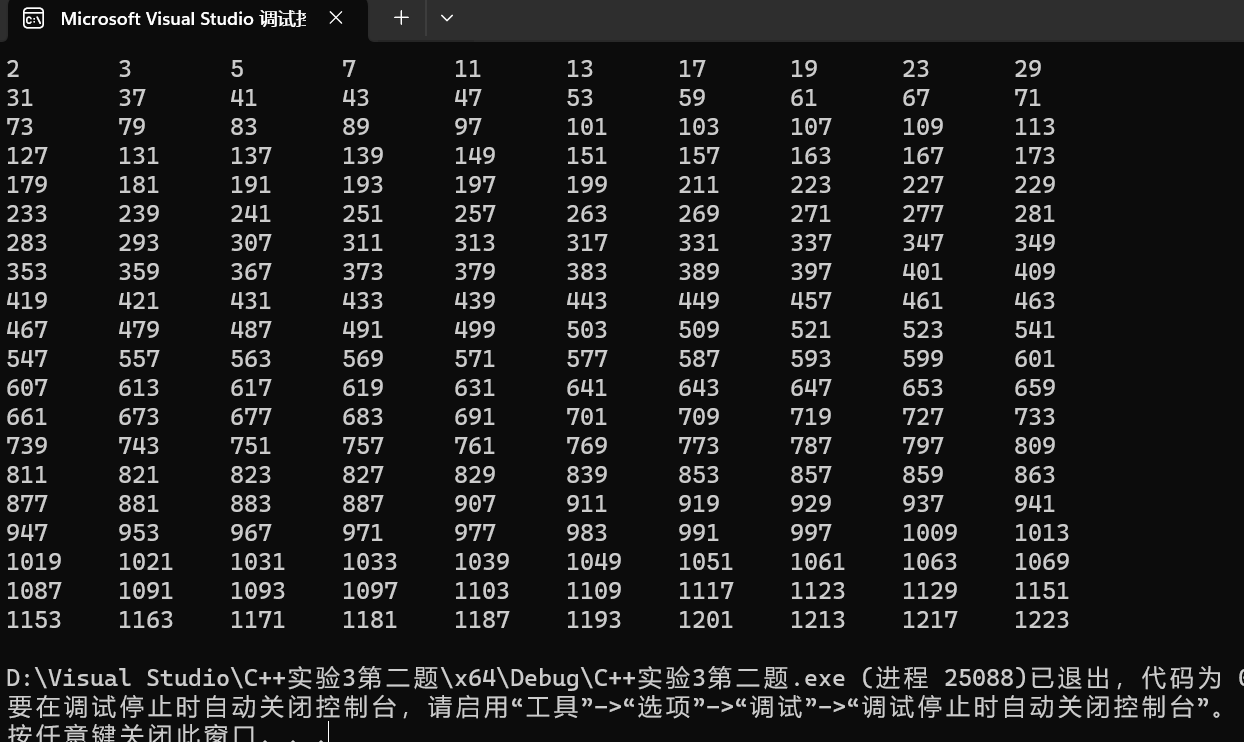
if (num1 % 10 == 0)cout << endl;//做到每行十个输出

}

}

return 0;

}

结果截图：

4、代码：

Cpp文件：

#include<iostream>

#include"mytriangle.h"

using namespace std;

bool is\_valid(double side1, double side2, double side3)//定义判断输入是否合法的函数

{

if ((side1 + side2 > side3) && (side3 + side2 > side1) && (side1 + side3 > side2))return true;

else return false;

}

double area(double side1, double side2, double side3)//定义求面积的函数

{

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

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

return s\_area;

}

int main()

{

double a, b, c;

cout << "请输入三角形的三条边：";//读取三边长

cin >> a >> b >> c;

if (is\_valid(a, b, c))cout <<"面积为：" << area(a, b, c) << endl;//判断出合法，求面积并输出

else cout << "输入不合法，请重试";//判断出不合法，输出不合法

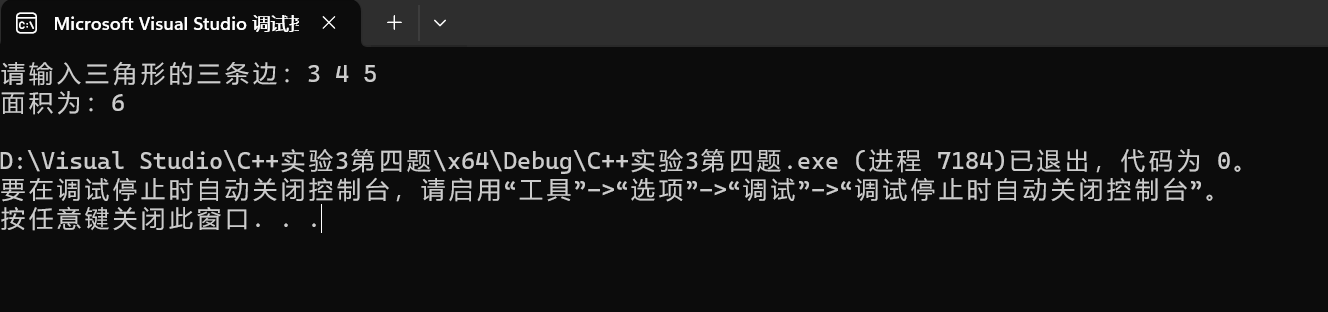
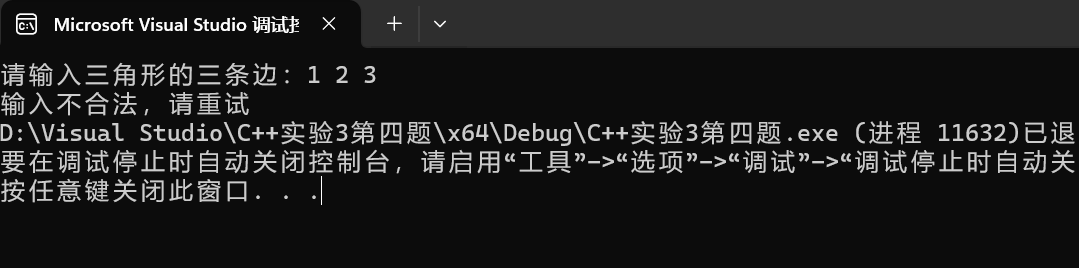
return 0;

}

头文件：

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

double area(double side1, double side2, double side3);//两个函数的声明

结果截图：

5、代码：

#include<iostream>

using namespace std;

int taozi(int n)

{

int num;

if (n == 1)

num = 1;

else

num = (taozi(n - 1) + 1)\*2;//利用函数的递归求出总桃子数

return num;

}

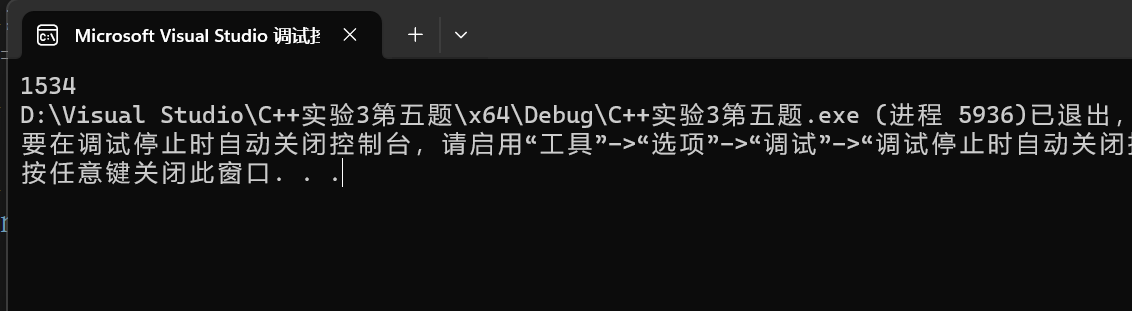
int main()

{

cout << taozi(10);

return 0;

}

结果截图：

**五、遇到的问题与解决方法**

1、函数的递归不熟悉。通过书本寻找后了解要用if语句实现。

2、多文件编程不了解，询问老师以及通过搜索引擎了解。

**六、体会**

1、编程时编写的时间往往比修改的时间少的多，要学会在遇到bug时分区块调试，比如加上cout输出一下看看是不是得到想要的结果。

2、多翻书看书上的代码学习，能有很多灵感。

**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**

（1）、1、代码：

#include<iostream>

using namespace std;

int main()

{

int arr[10] = { 0 };

cout << "Enter ten numbers:";

for (int i = 0; i < 10; i++)//读入十个数

{

cin >> arr[i];

}

int brr[10],x=0,y=0;//再定义一个新的数组brr

brr[0] = arr[0];//第一个数一定要存入新数组

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

{

for (int j = 0; j < i; j++)//判断是否与前面的值重复，即是否是新的值

{

if (arr[i] == arr[j])

{

x++;

y++;

break;

}

}

if (x == 0) //是新的值，将他存入新数组brr中

{

brr[i-y] = arr[i];

}

x = 0;

}

cout << "The distinct numbers are:";

for (int i = 0; i < 10 - y; i++)//将新数组brr中的值依次输出

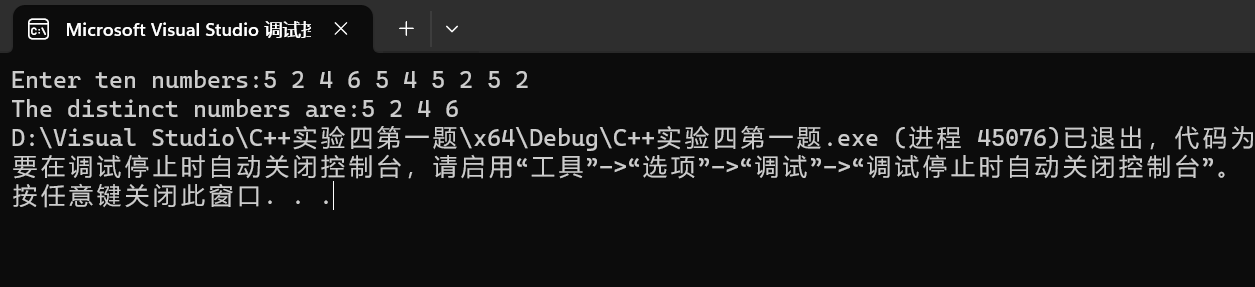
{

cout << brr[i] << '\x20';

}

return 0;

}

结果截图：

（1）、2、代码：

#include<iostream>

using namespace std;

void bubble\_sort(double list[],int listSize)//定义排序函数

{

bool changed = true;

do

{

changed = false;

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

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

{

double t = list[j];

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

list[j + 1] = t;

changed = true;

}

} while (changed);

}

int main()

{

double arr[10] = { 0 };//定义数组

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

{

cin >> arr[i];

}//为数组赋初值

bubble\_sort(arr,10);//调用排序函数

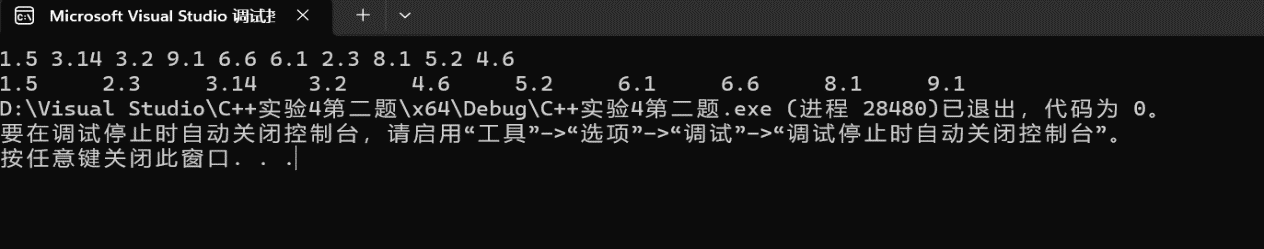
for (int i = 0; i < 10; i++)

{

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

}//输出排序后的数字

return 0;

}

（1）、3、代码：

#include<iostream>

using namespace std;

int main()

{

bool arr[100] = { false };//定义一个bool型数组，元素表示存物柜的开关状态

for(int i=0;i<100;i++)//利用for循环将每个元素都设置为开，此时第一个同学已改变了

{

arr[i] = true;

}

for (int i = 2;i<=100 ; i++)//循环每个学生

{

for(int n=0;(i+(i+1)\*n)<=100;n++)//用for循环找出各个学生要改变状态的柜子

{

if (arr[(i + (i + 1) \* n) - 1]) { arr[(i + (i + 1) \* n) - 1] = false; }

else { arr[(i + (i + 1) \* n) - 1] = true; }//用if语句将柜子改变状态

}

}

for (int i = 0; i < 100; i++)//将柜子为开的柜子号码输出

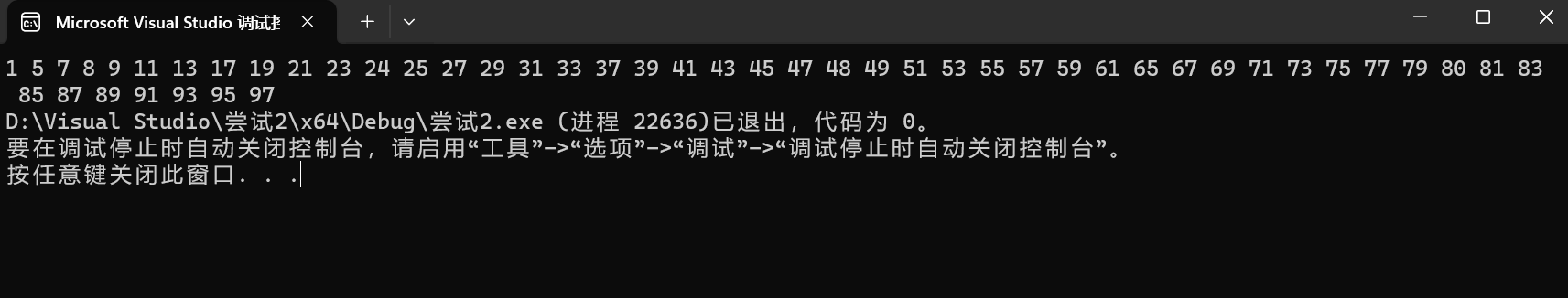
{

if (arr[i])cout << i + 1 << '\x20';

}

return 0;

}

结果截图：

（1）、4、代码：

#include<iostream>

using namespace std;

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

{

int x = 0, y = 0;

int k = 0;

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

{

if(list1[x]<list2[y])//循环size1+size2次，进行size1+size2次比较

{

if ((x < size1)&&(k==0))//防止出现未定义的数组元素。

{

list3[i] = list1[x];

}

x++;

if (x == size1) { x--; list3[i + 1] = list2[y]; y++; k++; }

}

else

{

if ((y < size2)&&(k==0))

{

list3[i] = list2[y];

}

y++;

if (y == size2) { y--; list3[i + 1] = list1[x]; x++; k++;}

}

}

}

int main()

{

int a, b,list1[80],list2[80];

cout << "Enter list1:";

cin >> a; //输入元素数

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

{

cin >> list1[i];//输入第一个排列好的数组

}

cout << "Enter list2:";

cin >> b;//输入元素数

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

{

cin >> list2[i];//输入第二个排列好的数组

}

int list3[80];

merge(list1, a, list2, b, list3);

cout << "The merged list is ";

for (int i = 0; i < a + b; i++)//输出排好的数组

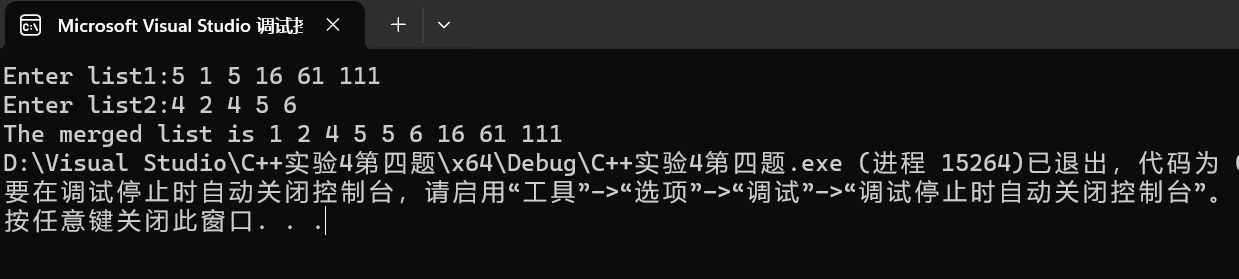
{

cout << list3[i] << '\x20';

}

return 0;

}

结果截图：

（1）、5、代码：

#include<iostream>

#include<cstdio>

using namespace std;

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

{

int x, y, z = 0, p = -1 ,ii;

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

{

x = a;

if (int(s1[a]) == 0)break;

}//将s1数组中赋值的元素个数赋值给x

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

{

y = a;

if (int(s2[a]) == 0)break;

}//将s2数组中赋值的元素个数赋值给y

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

{

ii = i;

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

{

if (int(s2[ii]) == int(s1[j])) { z++; }

ii++;

}

if (z == x)

{

p = i;

break;

}

else z = 0;

}//利用两个for循环嵌套找到s1在s2中的下标并赋值给p,否则p的值为-1

return p;

}

int main()

{

#define maxlenth 100

char s1[maxlenth];

cout << "Enter the first string:";

cin.getline(s1, maxlenth);//读入第一个字符串

char s2[maxlenth];

cout << "Enter the second string:";

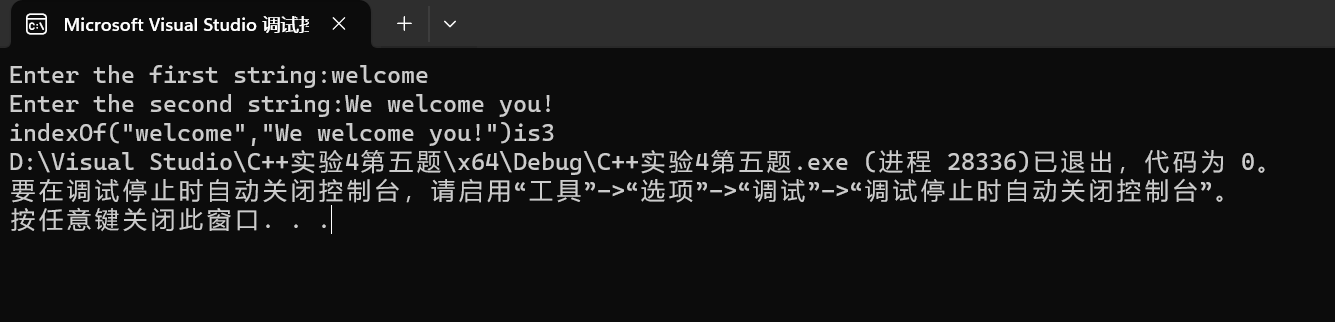
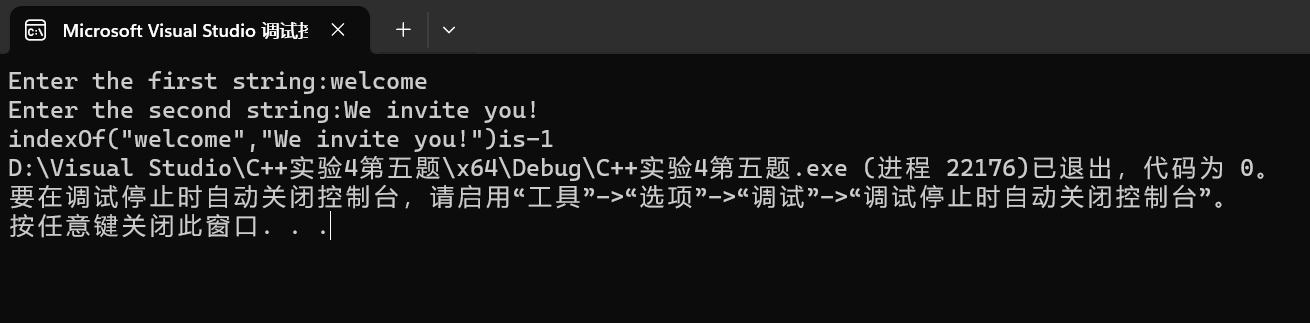
cin.getline(s2, maxlenth);//读入第二个字符串

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

cout << indexOf(s1, s2);//调用函数

return 0;

}

结果截图：

（1）、6、代码：

#include<iostream>

#include<cstdio>

using namespace std;

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

{

for (int i = 0; i < 26; i++)//对26个字母循环

{

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

{

if ((int(s[j]) == i + 97) || (int(s[j] == i + 65)))

{

counts[i]++;

}

if (int(s[j]) == 0)break;

}

}

}

int main()

{

const int maxlenth = 50;

char s\_s[maxlenth];

char ss[27] = "abcdefghijklmnopqrstuvwxyz";

int s\_counts[26] = { 0 };

cout << "Enter a string:";

cin.getline(s\_s,maxlenth);//读入字符串

count(s\_s, s\_counts);//调用函数

for (int i = 0; i < 26; i++)//显示各字母非零的次数

{

if (s\_counts[i] != 0)cout << ss[i] << ":" << s\_counts[i] << " times" << endl;

}

}

结果截图：

（2）、1、代码：

#include<iostream>

#include<cstdio>

using namespace std;

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

{

int x, y, z = 0, p = -1, ii;

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

{

x = a;

if (int(\*(s1+a)) == 0)break;

}//将s2字符串的元素个数赋值给y

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

{

y = a;

if (int(\*(s2+a)) == 0)break;

}//将s2字符串的元素个数赋值给y

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

{

ii = i;

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

{

if (int(\*(s2+ii)) == int(\*(s1+j))) { z++; }

ii++;

}

if (z == x)

{

p = i;

break;

}

else z = 0;

}//利用两个for循环嵌套找到s1在s2中的下标并赋值给p,否则p的值为-1

return p;

}

int main()

{

const int maxlenth = 100;

char s\_s1[maxlenth] = { 0 };

char s\_s2[maxlenth] = { 0 };

cout << "请输入字符串s1：";

cin.getline(s\_s1, maxlenth);//输入第一个字符串

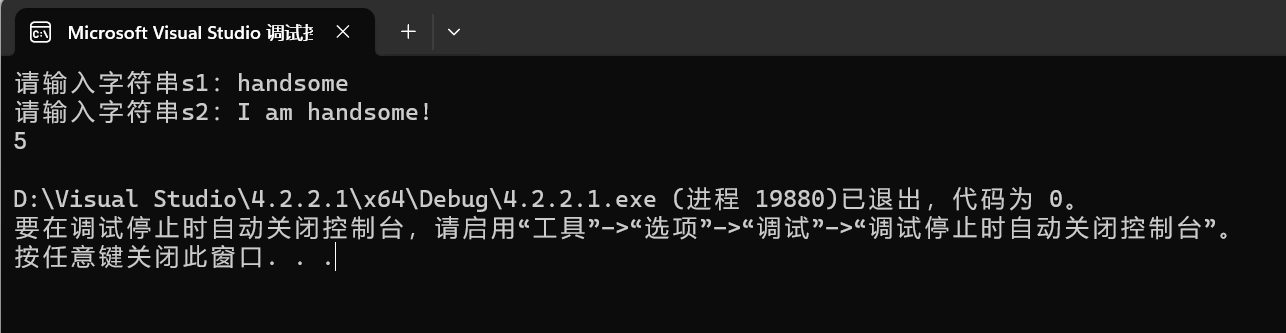
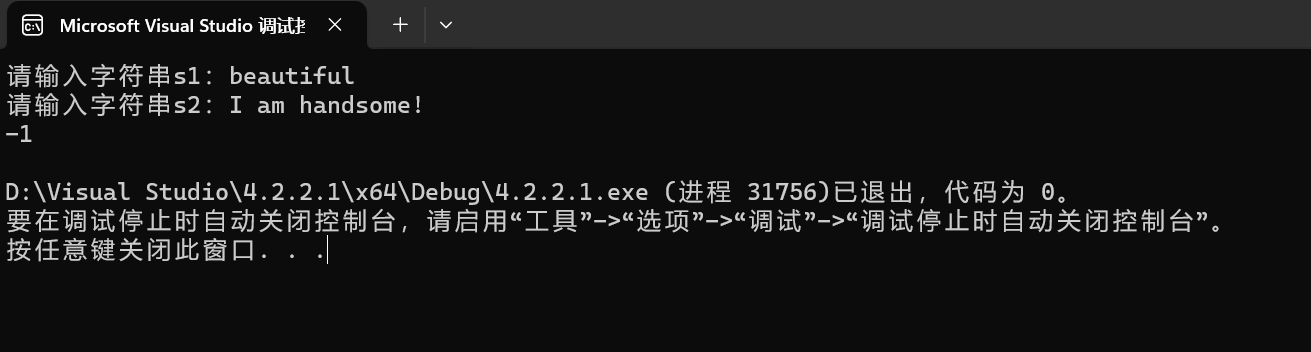
cout << "请输入字符串s2：";

cin.getline(s\_s2, maxlenth);//输入第二个字符串

cout << indexof(s\_s1, s\_s2) << endl;//调用函数

return 0;

}

结果截图：

（2）、2代码：

#include<iostream>

using namespace std;

int parseHex(const char\* const hexString)

{

int decnum = 0, k = 0, x = 0;

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

{

k = i;

if (int(\*(hexString + i)) == 0)break;

}//找到有几位十六进制数。

for (int i = k-1; i >= 0; i--)

{

if(k-1==i){

if (int(\*(hexString + i)) >= 65) x = int(\*(hexString + i)) - 65 + 10;

else x = int(\*(hexString + i)) -49 +1;

}

else {

if (int(\*(hexString + i)) >= 65) x = int(\*(hexString + i)) - 65 + 10;

else x = int(\*(hexString + i)) -49 + 1;

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

{

x \*= 16;

}

}

decnum += x;

}

return decnum;

}

int main()

{

const int maxlenth = 100;

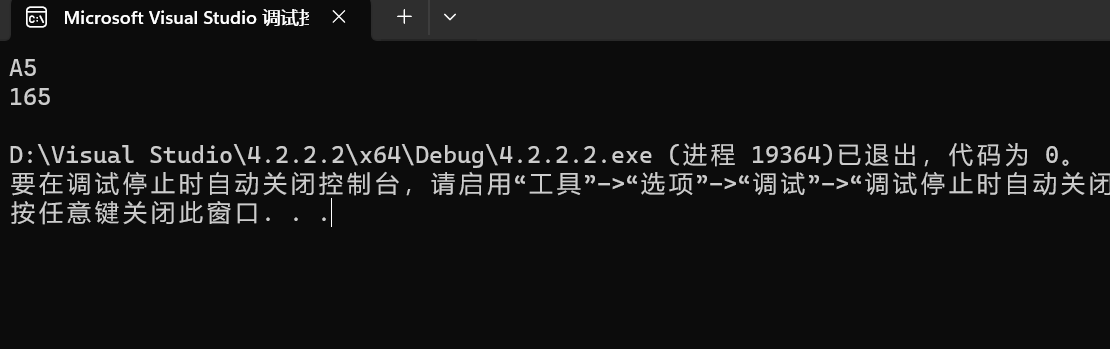
char s[maxlenth];

cin >> s;

cout << parseHex(s) << endl;

return 0;

}

结果截图：

（2）、3、代码：

#include<iostream>

using namespace std;

void bubble\_sort(int list[], int listSize)//定义排序函数

{

bool changed = true;

do

{

changed = false;

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

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

{

int t = list[j];

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

list[j + 1] = t;

changed = true;

}

} while (changed);

}

int main()

{

int i;

cout << "请分别输入个数及数组元素:";

cin >> i;//输入元素个数

int\* p = new int[i];

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

{

cin >> \*(p + j);//输入数组元素

}

cout << p << '\t' << \*p << endl;//观察指针与指针指向的内容

bubble\_sort(p, i);

cout << "排序后数组元素为:";

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

{

cout << \*(p + j) << '\t';//用指针方式输出数组元素

}

if (p != NULL)

{

delete[]p;

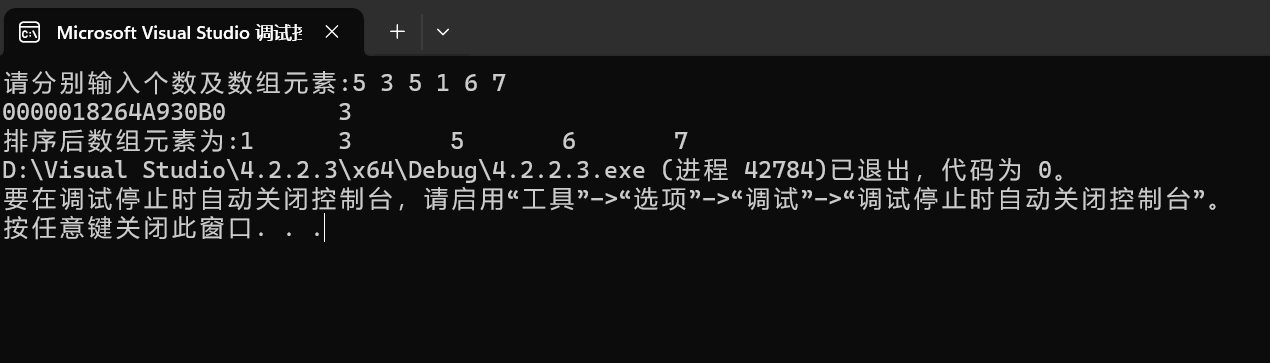
p = NULL;

}//释放数组内存。

return 0;

}

结果截图：



**四、遇到的问题与解决方法**

1、赋值号=与判断是否相等的符号==是两个不同的符号，千万不要混用。

2、用字符为字符数组赋初值只要cin>>数组名。输入时直接输入字符串。

但是不能带有空格。

3、以数字为外表的字符取int值后不是该数字，而是从49开始。

**五、体会**

1、书本上的例题很重要，能为我们提供很多思路以及知识指引。

2、在做实验之前要做到有大致思路之后再去做，不要边做边想。做的过程中要想一想每一步的目的以及该步骤的作用，有一个区块意识。