## 实验八 Pointers and Pointer-based Strings

## Objectives

## 学号: 09017423 姓名：杨彬

### 实验目的

Objectives

To understand the concept of pointer, master pointer-based indirect organization method of data.

To master the basic of declare and use pointer in C++, the basic operation of pointer.

To master the relationship between pointer and array, the pointer-based array operation.

To master the relationship between pointer and string, the pointer-based string operation.

To master the use of standard string library fuction.

To master the relationship between pointer and function, the pointer-based function operation

（目标

为了理解指针的概念，基于主指针的间接数据组织方法。

掌握申报和C++使用指针的基本指针的基本操作。

掌握指针与数组的关系，基于指针的数组运算。

为了掌握指针和字符串之间的关系，基于指针的字符串操作。

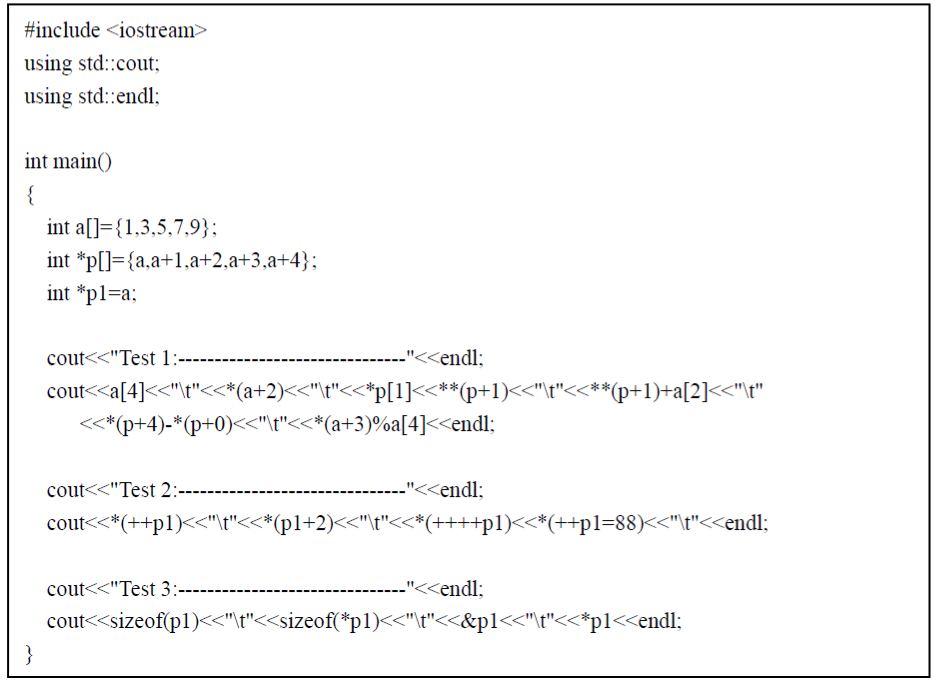
掌握标准字符串库函数的用法。

掌握指针与函数的关系，基于指针的函数运算）

### 实验内容

Exp1

Experiments 1、To run the following program and analysis its result.



Exp2

To declare an array and input 10 integers into it. Then sort and output the array by pointer-based operation.

(声明一个数组并输入10个整数。然后通过指针操作对数组进行排序和输出。)

Exp3

To declare a function to swap two integer as following every prototype, and analysis carefully their executing. void swap(int a, int b);

// declare a local identifier temp void swap(int a, int b);

// declare a local identifier temp\* void swap(int\* a, int\* b);

// declare a local identifier temp void swap(int\* a, int\* b);

// declare a local identifier temp\* void swap(int& a, int& b);

// declare a local identifier temp

(声明一个函数来交换两个整数作为每个原型，并仔细分析它们的执行情况。空交换（int，int）；

/声明一个本地标识符临时替换（int，int）；

/声明一个本地标识符空值交换（int \*，int \*）；

/声明一个本地标识符临时替换（int \*，int \*）；

/声明一个本地标识符空值交换（int，a，int和b）；

/声明本地标识符温度

)

EXP4

To declare a function to get the sum of two integer as following every prototype, and analysis carefully their executing. int getsum(int a, int b);

// declare a local identifier temp, return temp int\* swap(int a, int b);

// declare a local identifier temp, return the address of temp void swap(int a, int b);

// declare a local identifier temp\*, return temp void swap(int a, int b);

// declare a local identifier temp\*, return the content of temp void swap(int a, int b);

// declare a local identifier temp&, return temp

Exp5

To write a function strcpy(char\*s1, const char\* s2) that copies string s2 into the character array s1. (写一个函数strcpy（char \* const char \* S1，S2）拷贝到字符数组中的字符串。)

Exp6

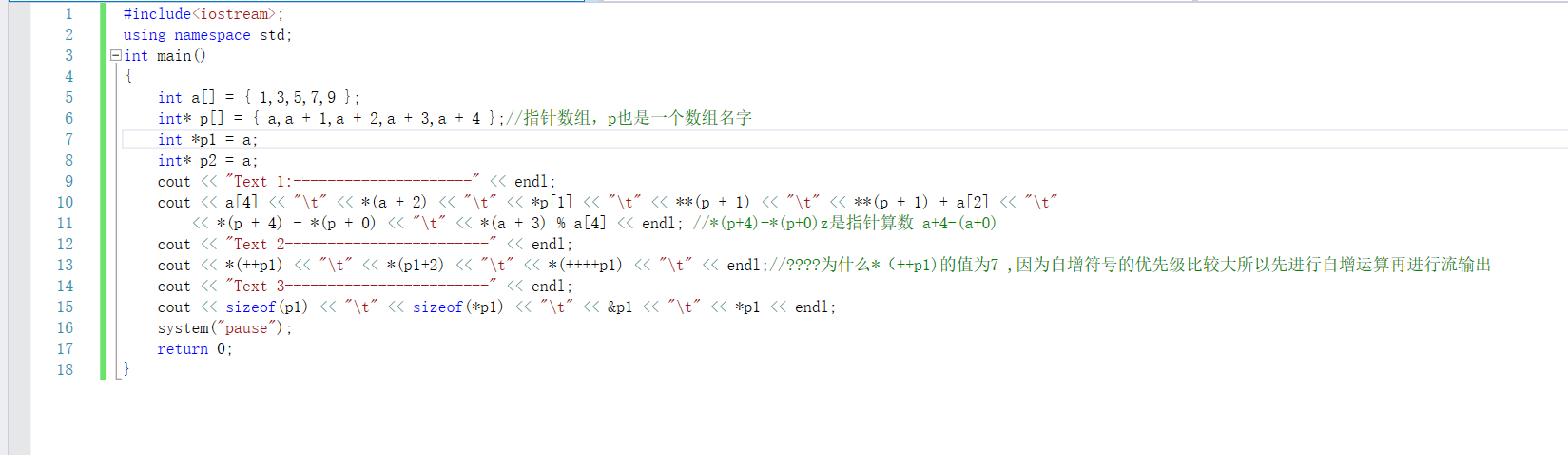
To write a function strcat(char\*s1, const char\* s2) that appends the string s2 to s1.

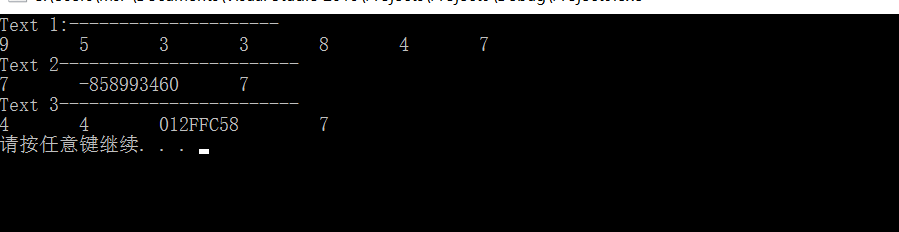
(写一个函数strcat（char \* const char\* S1，S2），将字符串S2到S1。)

Exp7、At first, we declare a 3-by-5 two-dimensional array. Then it and output result by pointer-based operation.

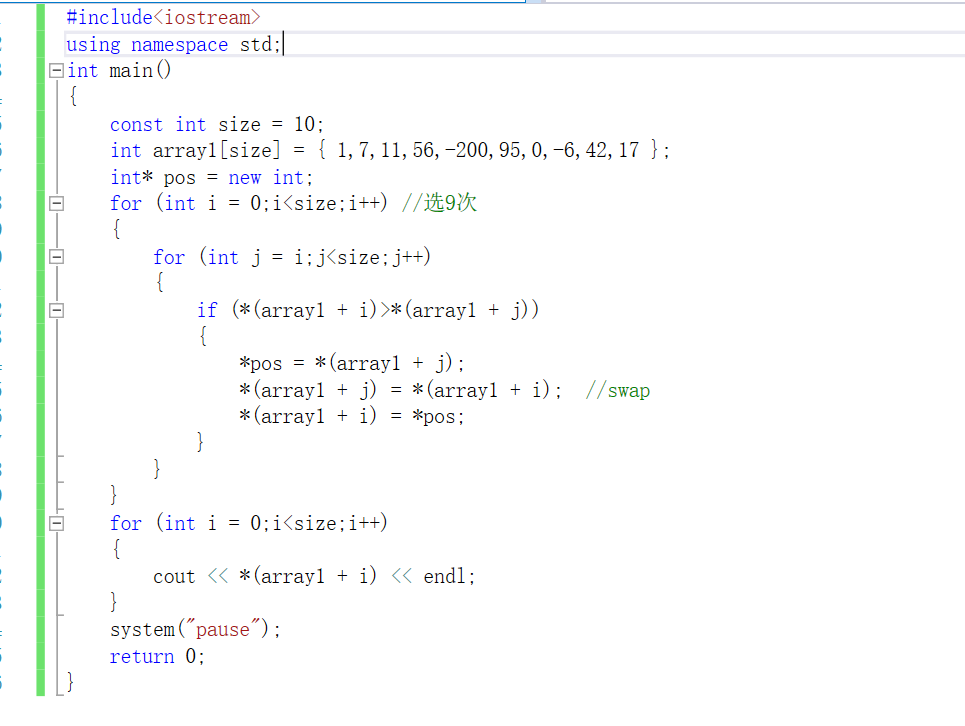
### 实验代码及结果：

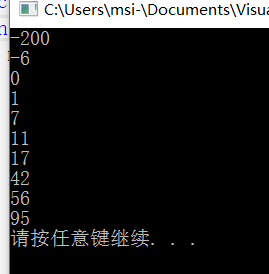
Exp1



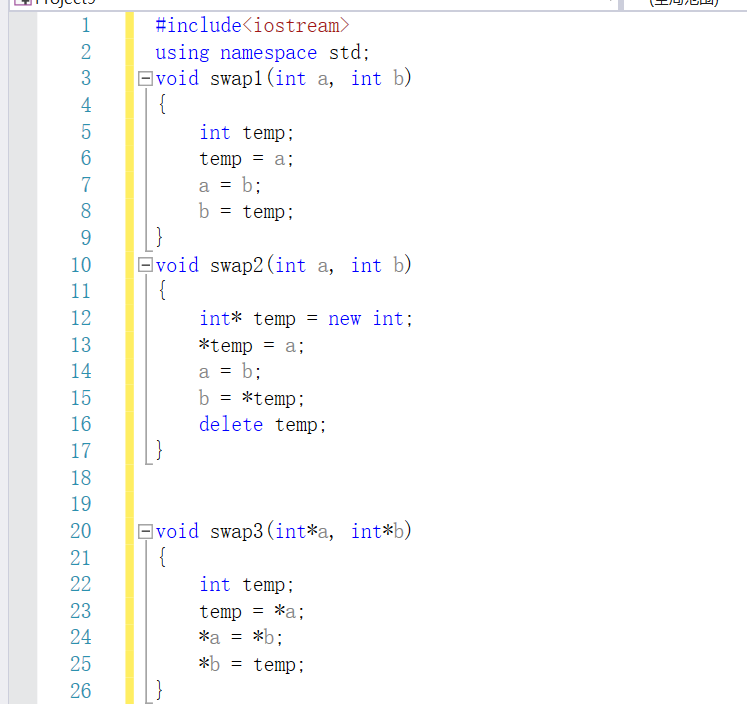
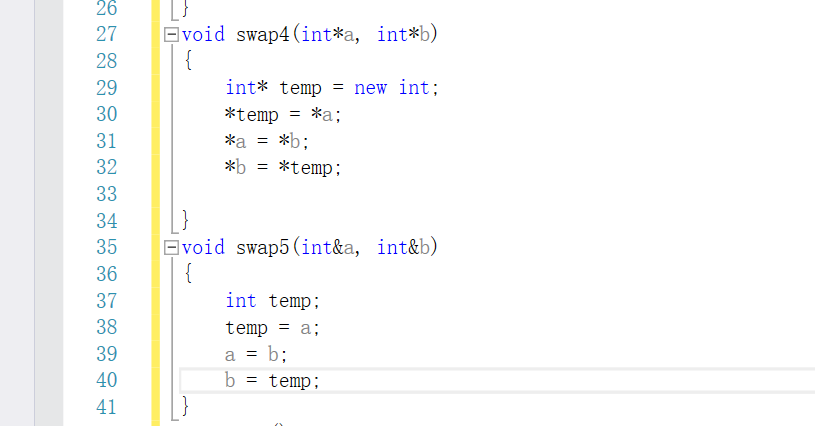


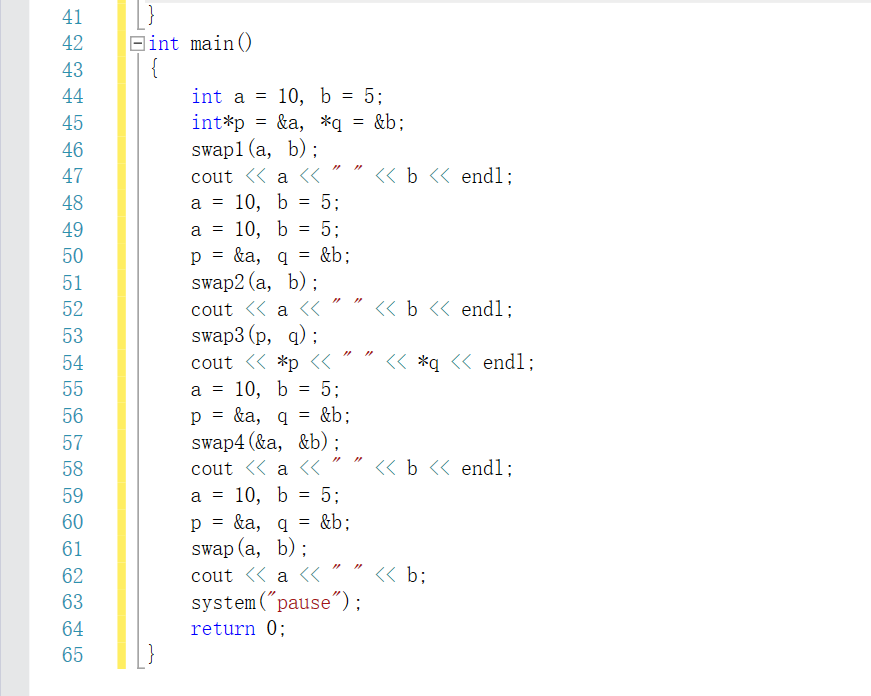
Exp2

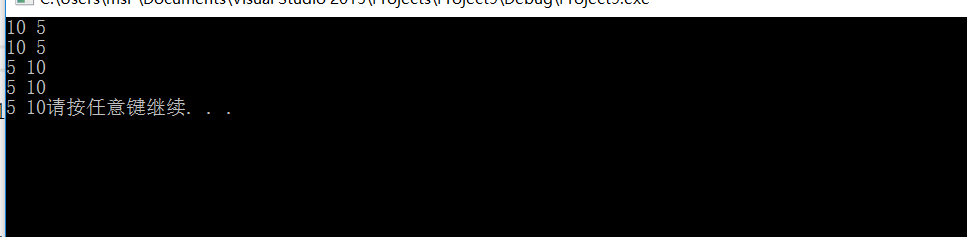




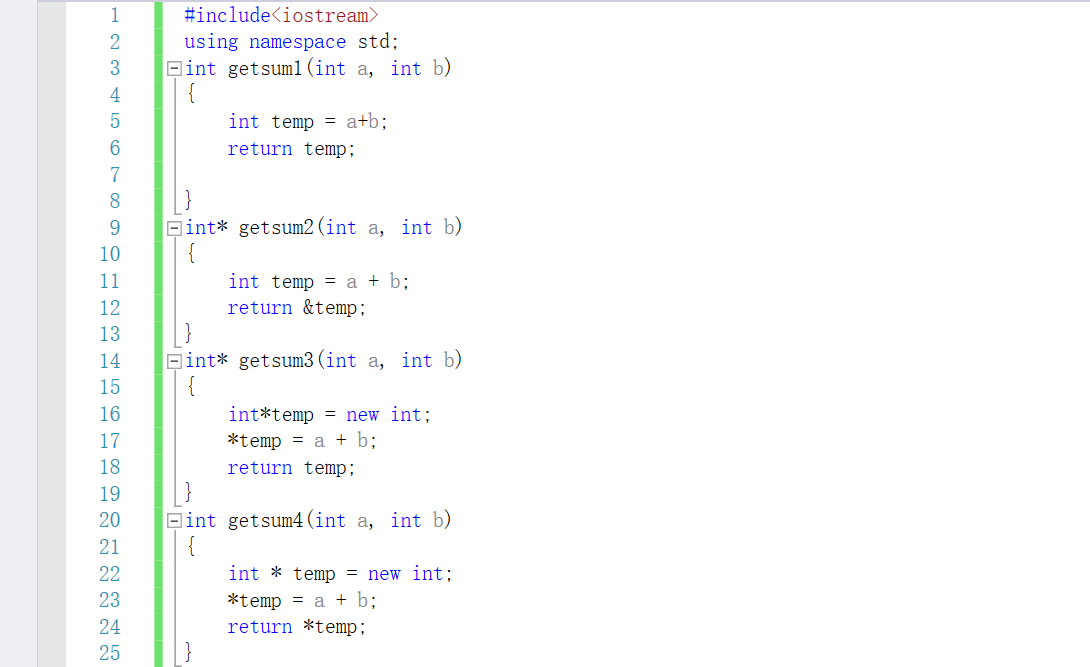
Exp3

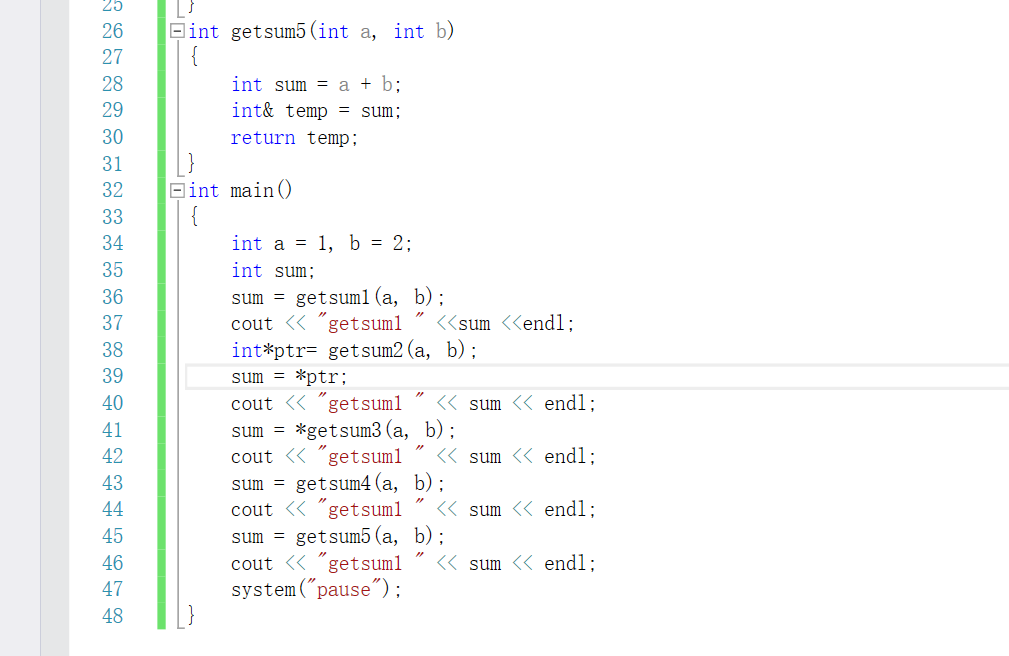
 

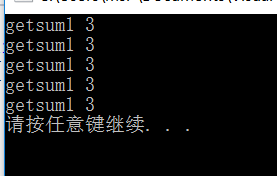




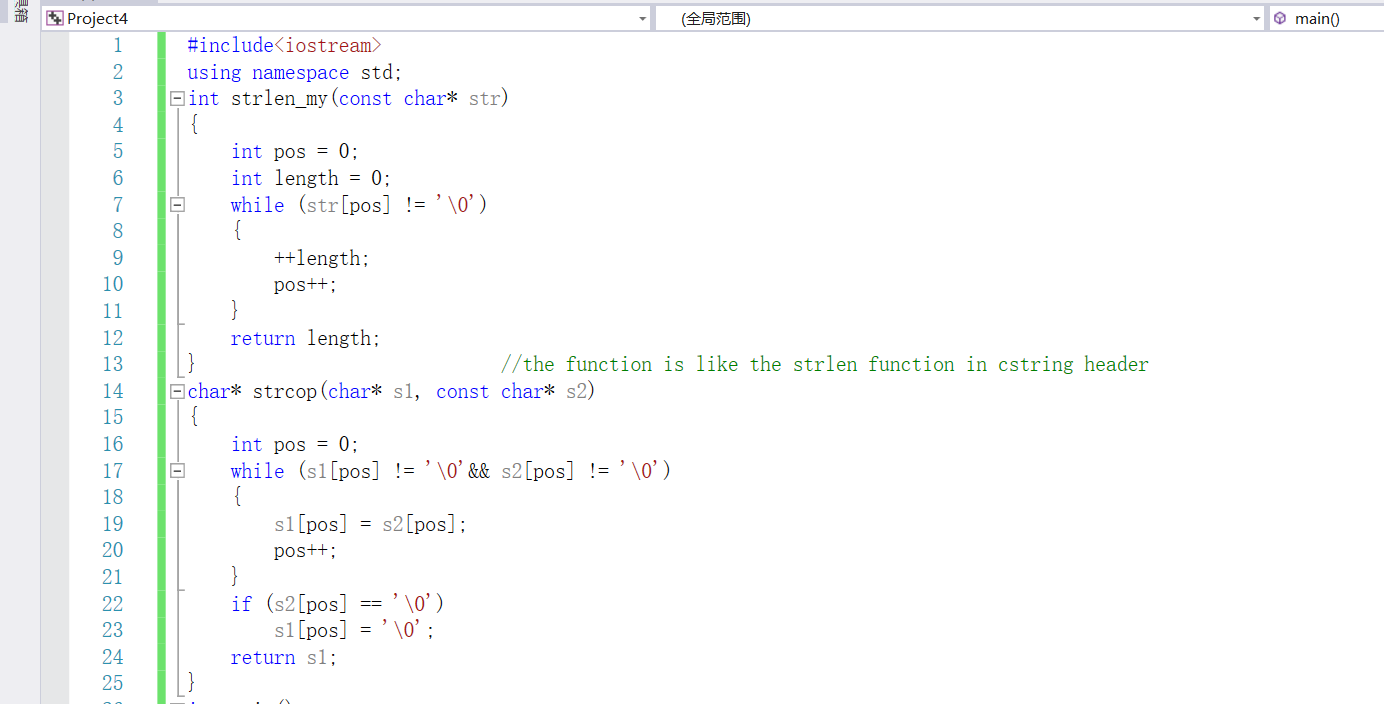
Exp4

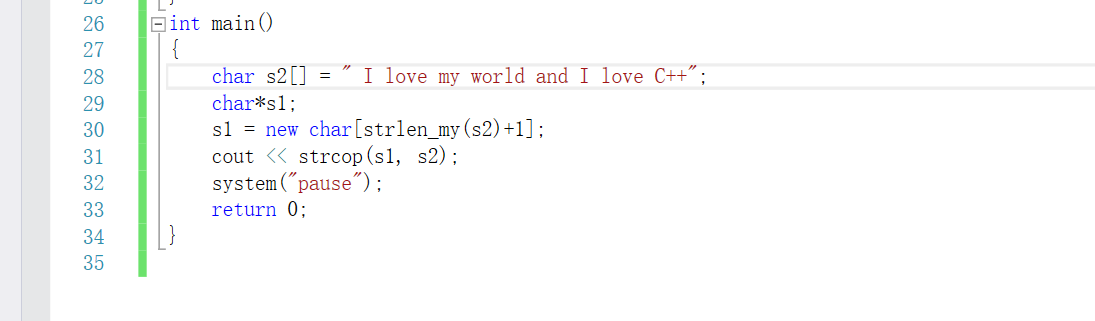


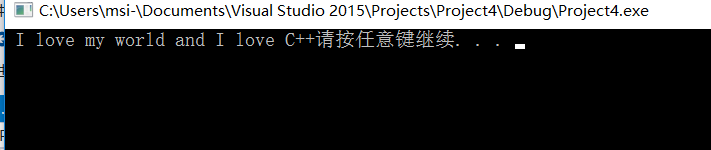




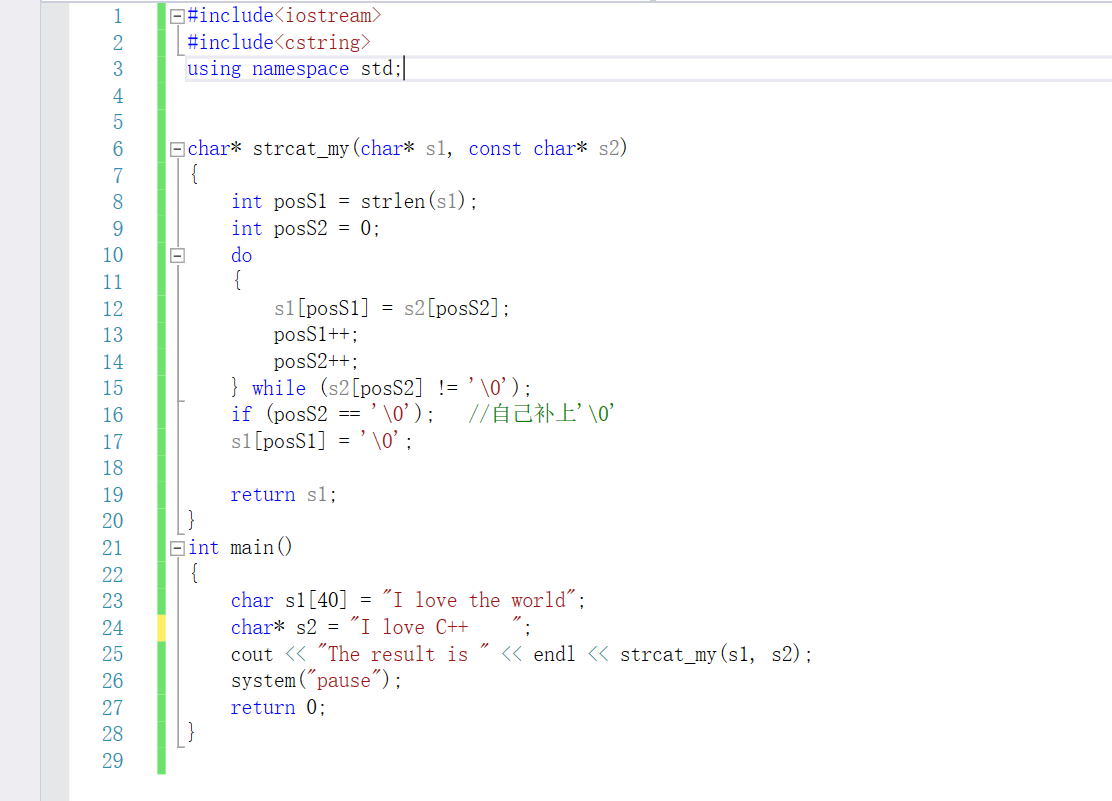
Exp5





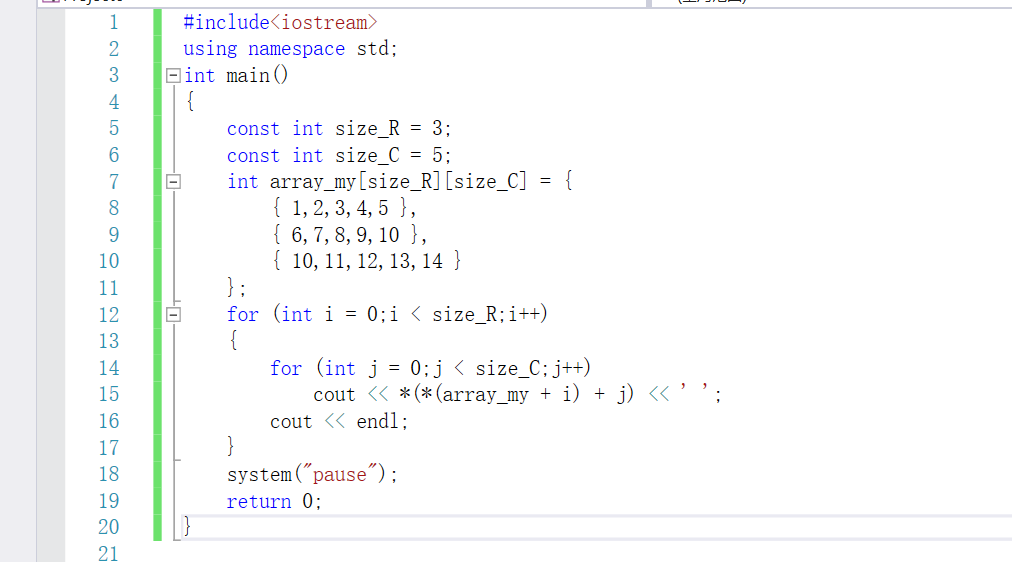


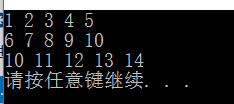
Exp6



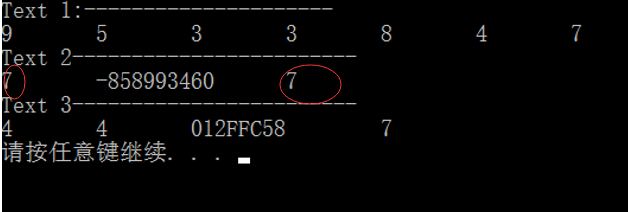


Exp7





### 遇到的问题及解决过程

Exp1 一开始我有一个疑惑

为什么我标注出来的值是7？

询问老师之后，我知道了++运算符的优先级比<<流输出符的优先级大，所以对于这个语句



程序进行全部的自增之后才输出结果。而第二个值出现的原因是因为数据溢出了。

Exp2 :nothing

Exp3:nothing

Exp4:为什么这四个函数都够得到正确的结果？

经过思考得出了结论，因为它们针对地址进行操作，直接给哪一个地址所在的内存块赋值，所以说虽然指针在函数体中被声明，函数结束后指针变量就消亡了，但是因为它是对地址进行操作，仍能得出结果。

Exp5:nothing

Exp6;nothing

Exp7 nothing

### 实验体会

Exp1

通过这个实验，我加深了对于指针的理解。具体表现为

1如何使用指针算数，以及如何通过指针进行间接引用（\* ——inreference operator）

2对于指针数组的使用 

3指针变量的大小是4个字节 

同时也知道了表达式的重要性，很多时候结果的错误正是因为，没有弄清楚运算符的优先级和结合度导致的。因此有理由重视对表达式的理解。

Exp2:基于指针的运算，在使用上更加的灵活，但是也有其弊端，比如书写上较数组风格的写法，更为复杂，而且容易出错。并且采用指针程序的可读性会变差

Exp3这个程序证明了，传递参数给函数有两种方式 1值传递2地址传递。

但是值传递不能改变参数原来的值 比如函数1， 2

而地址传递可以改变参数的原始值 如函数3、4、5，而且地址传递也有两种传递方法1通过指针传递，然后间接引用。2通过直接引用（别名）

Exp4使用指针和引用，都是直接针对所对应的地址的内存块进行操作，能够直接改变值，与对普通的变量进行操作存在区别。

Exp5\6通过这两个实验，模拟了cstring 库函数中 strcop和strcat 的处理机制，并且自己补上了’\0’

Exp7这个实验掌握了二维数组的初始化方法eg{{}，{}，{}，}同时强化了再上一个实验中遗留下来的，关于用指针来访问二维数组的某个元素的方法

即