**C++第八次实验报告——实验过程展示**

**程序阅读题**

1、阅读程序，并输出结果

#include <iostream>

int main() {

using namespace std;

double \*p3=new double[3];

p3[0]=0.2; p3[1]=0.5; p3[2]=0.8;

cout<<”p3[1] is”<<p3[1]<<”.\n”;

p3=p3+1;

cout<<”Now p3[0] is”<<p3[0]<<” and”;

cout<<” p3[1] is”<<p3[1]<<” .\n”;

p3=p3-1;

delete [] p3;

return 0;

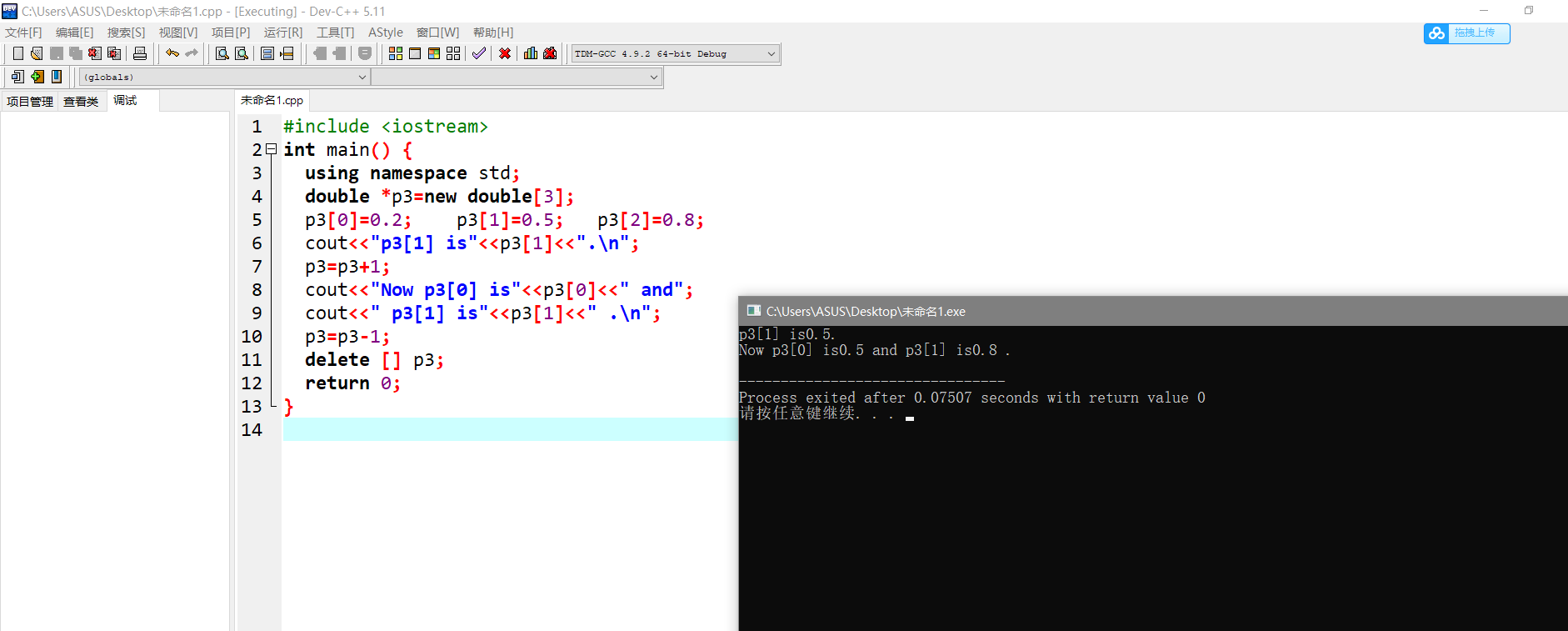
}

**预测输出结果：**

**p3[1] is0.5.**

**Now p3[0] is0.5 and p3[1] is0.8 .**

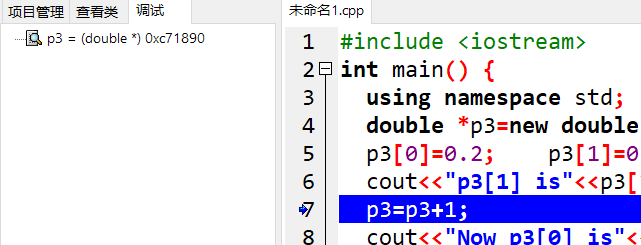
**结果验证：**



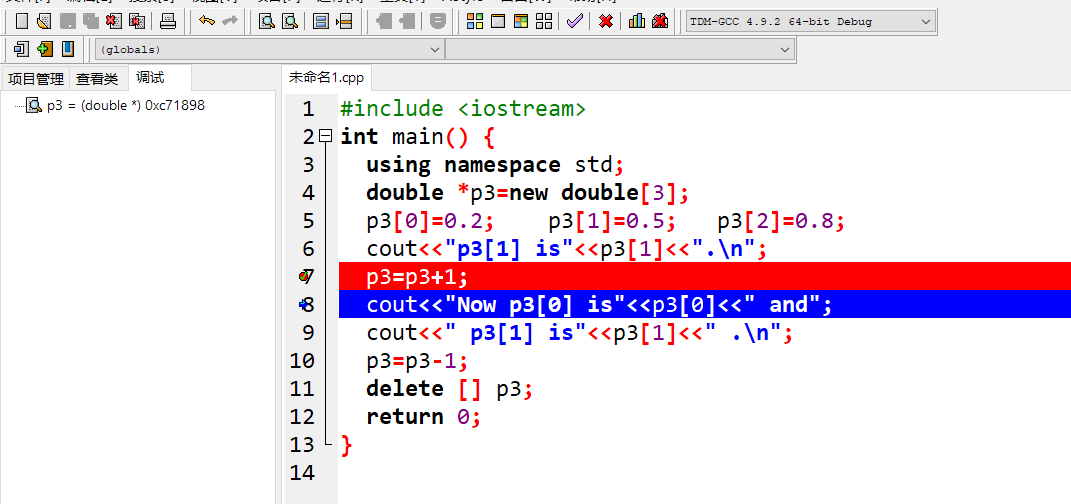
**结果正确。**

**程序调试：**

**断点：**



**在对指针+1之后：指针后移了一个double变量大小的长度。**

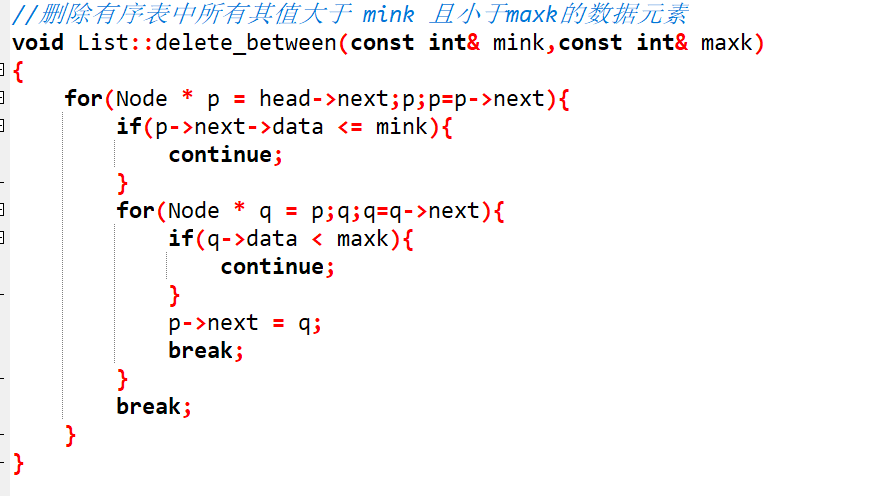


**六、程序题**

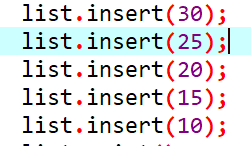
（一）链表应用题

1 已知线性表中的元素以值递增有序，并以单链表作存储结构。试写一程序，删除有序表中所有其值大于 mink 且小于maxk的数据元素。

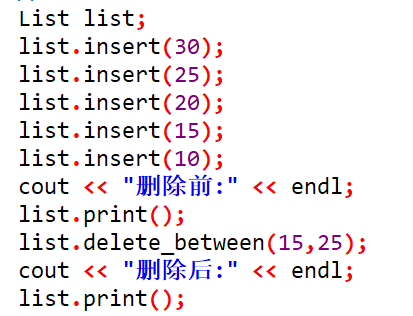
**代码实现：（链表的创建见cpp文件）**



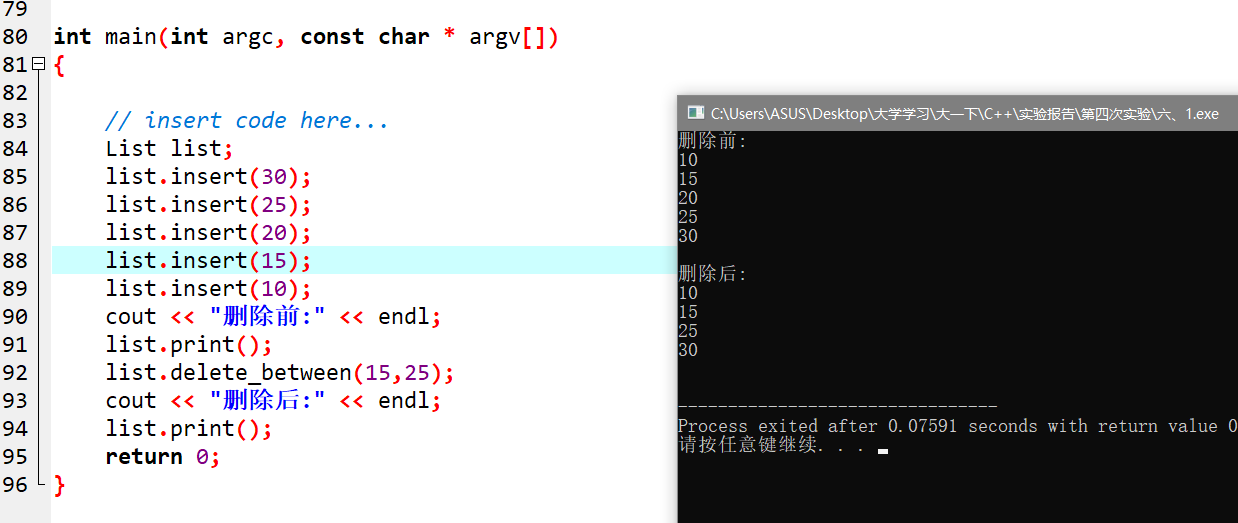
**链表输入：**



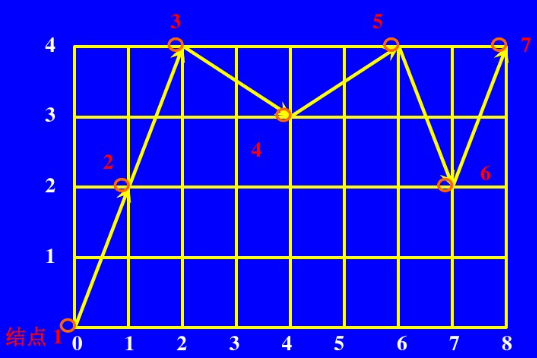
**链表输出：**



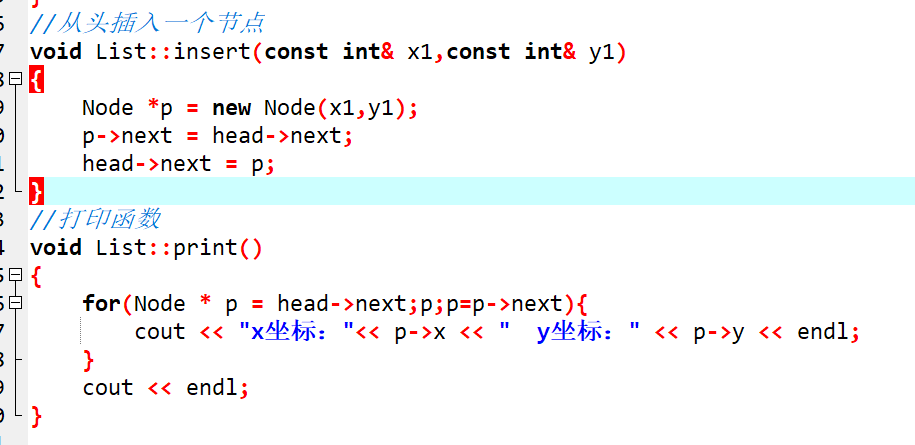
**最终结果：（只有20被删除了）**



2跳马。依下图将每一步跳马之后的位置（x,y）存放到一个“结点”中，再用“链子穿起来”，形成一条链。

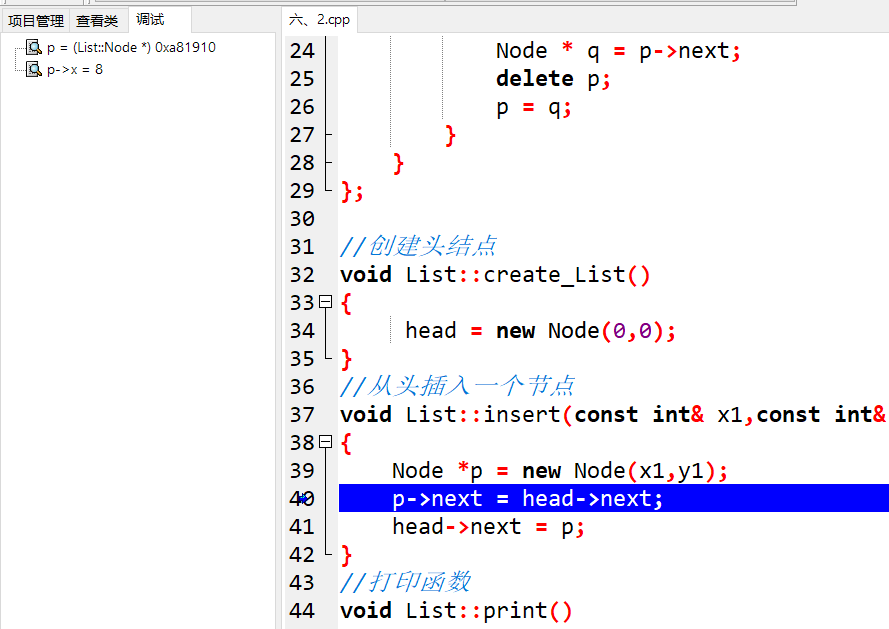


**存储实现：**



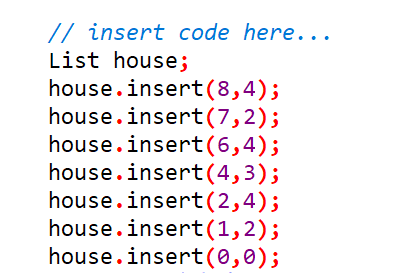
**原理：倒序存储各个坐标，每次从头插入节点**

**程序调试：**

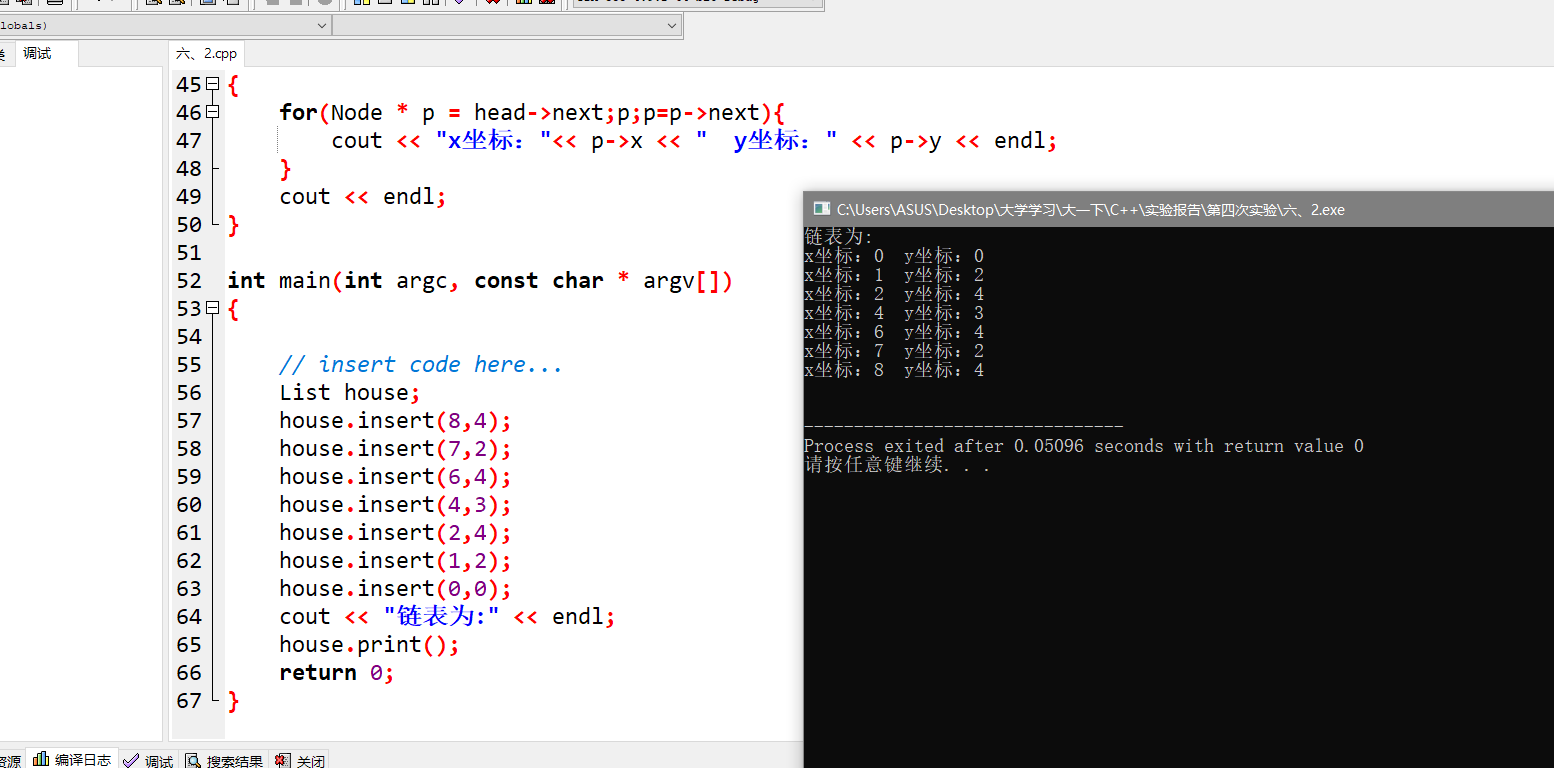


**（查看存储链表的过程）**

**链表输入：**



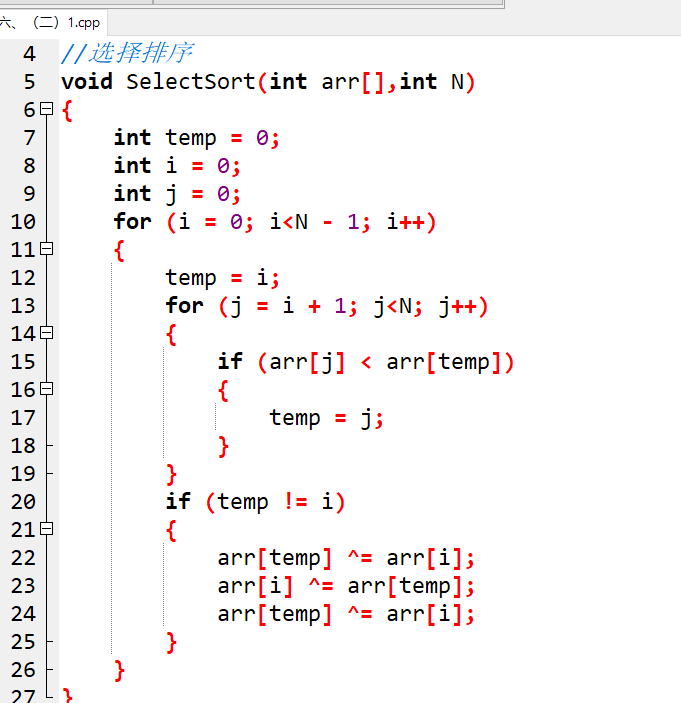
**程序结果：**



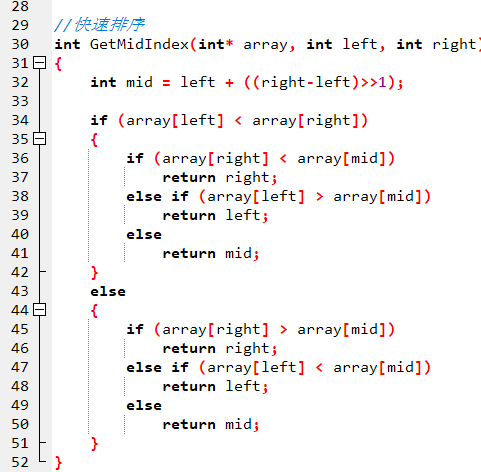
（二）动态数组的应用题

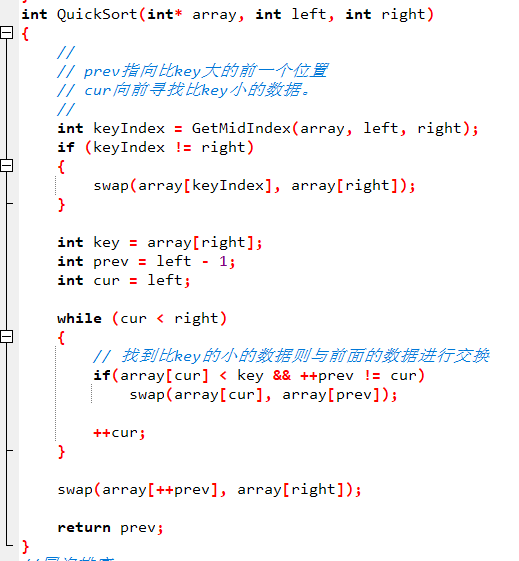
1、用new创建一个数组，从键盘上读取数组长度和数据，分别采用：选择排序、快速排序和冒泡排序，并输出数据。

选择排序：

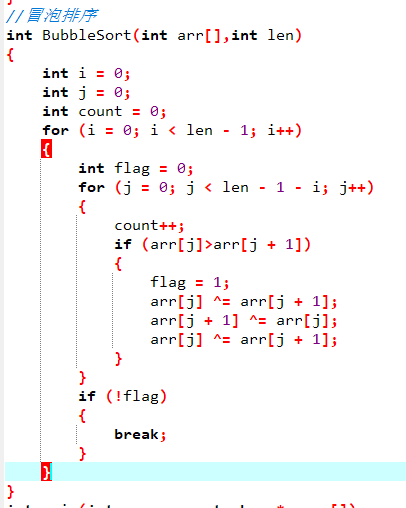


快速排序：

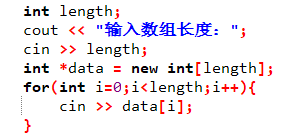




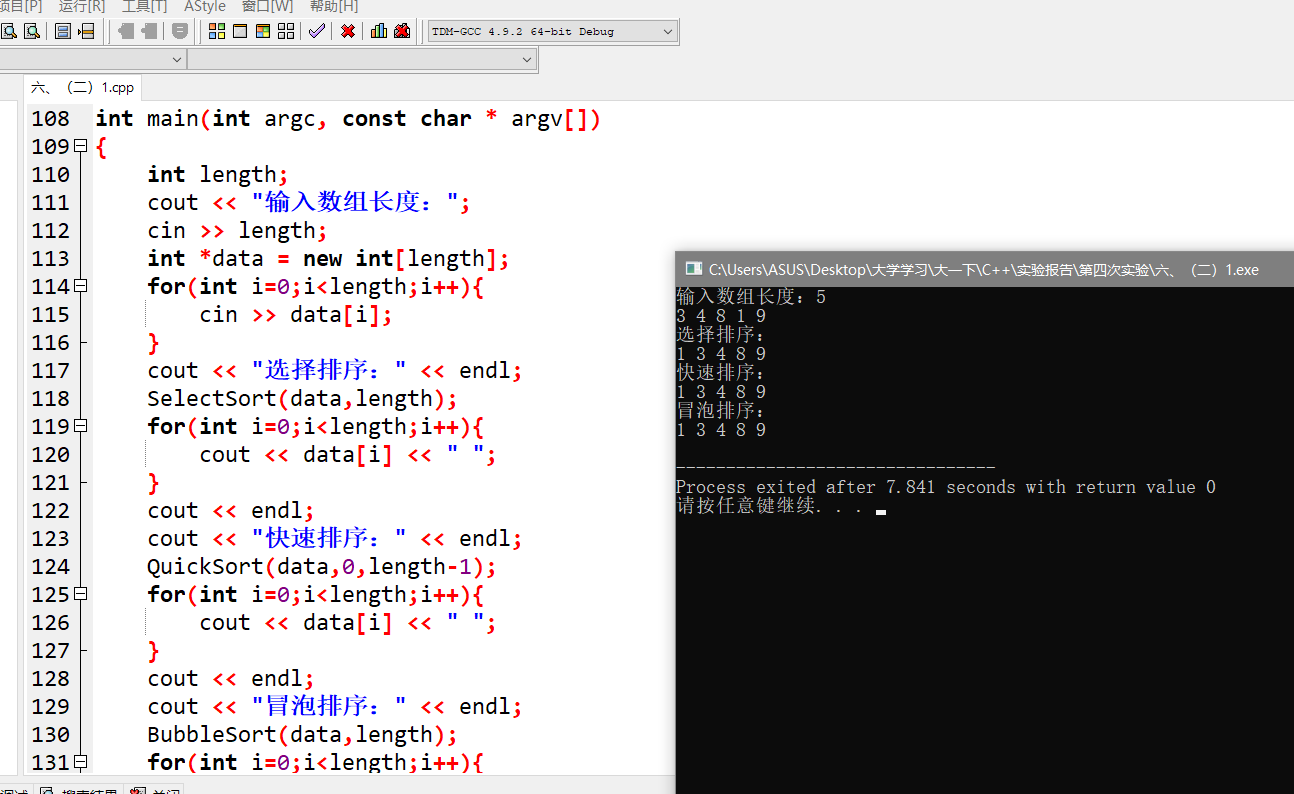
冒泡排序：



动态数组读入：

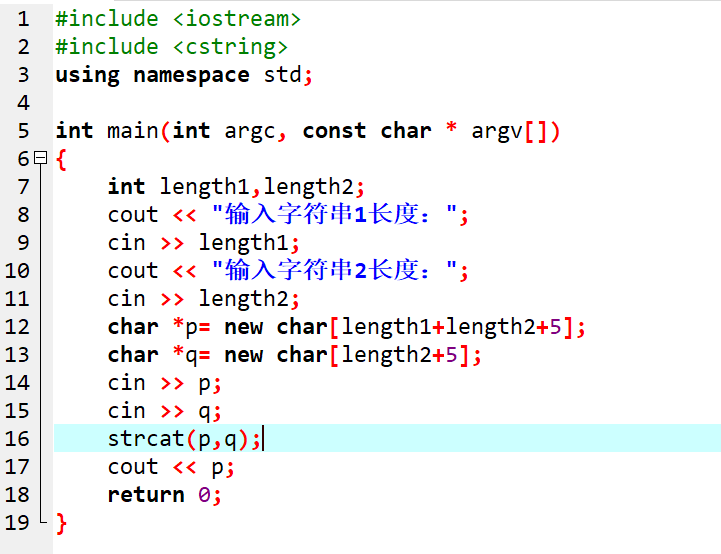


程序黑盒测试：

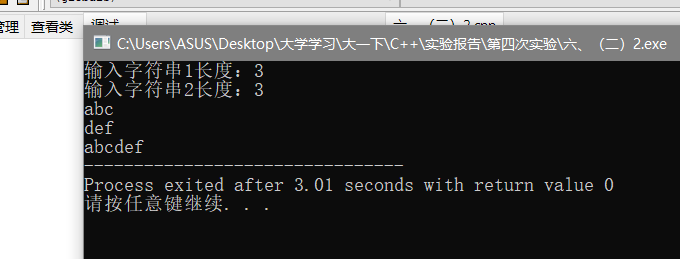


2、用动态数组分别存储两个字符串，并将两个字符串连接的结果存储至第一个动态数组中。【重点描述，若动态数组空间申请不够时，如何动态扩展新空间】

代码实现：



黑盒测试：



动态拓展：不够时可以用new新开辟一个两倍空间大小的数组，利用memcpy()函数拷贝数据。例如下图代码：

