// StringSort.cpp : 定义控制台应用程序的入口点。

// 第一周第二题

/\*

题目：输入5个英文字符串，按照字符串长度排序，由小到大输出。

示例：

输入：

apple banana watermelon dish one

输出：

one dish apple banana watermelon

编译环境 Visual Studio 2015

作者：杨政权

\*/

//实验代码

#include "stdafx.h"

#include <iostream>

#include < string>

using namespace std;

#define SIZE 5 //字符串数量

#define MAX 30 //字符串长度

char input[SIZE][MAX] = { 0 };

void sort(char str[][MAX], int size)

{

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\* 对size个字符串按长度用冒泡法进行升序排序

\*\*\* MAX-字符串最大长度

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int counter1, counter2;

char temp[MAX];

for (counter1 = 0; counter1<size - 1; ++counter1)

for (counter2 = 0; counter2<size - 1 - counter1; ++counter2)

{

//如果字符串str[counter2]长度大于它后面的字符串

if (strlen((char \*)(str[counter2]))>strlen((char \*)

(str[counter2 + 1])))

{

//将字符串str[counter2]与它后面的字符串换个位置

strcpy(temp, (char \*)(str[counter2 + 1]));

strcpy((char \*)(str[counter2 + 1]),

(char \*)(str[counter2]));

strcpy((char \*)(str[counter2]), temp);

}

/\* 如果字符串str[counter2]长度等于它后面的字符串 且

\* 字符串str[counter2]按字母排序大于它后面的字符

\*/

if (strlen((char \*)(str[counter2])) == strlen((char \*)

(str[counter2 + 1])) &&

strcmp((char \*)(str[counter2]),

(char \*)(str[counter2 + 1]))>0)

{

//将字符串str[counter2]与它后面的字符串换个位置

strcpy(temp, (char \*)(str[counter2 + 1]));

strcpy((char \*)(str[counter2 + 1]),

(char \*)(str[counter2]));

strcpy((char \*)(str[counter2]), temp);

}

}

}

int main()

{

int counter = 0;

//输入字符串

while (counter<SIZE)

{

cout << "请输入第" << counter+1 << "个字符串," << "共" << SIZE << "个字符串" << endl;

cin >> input[counter];

++counter;

}

//调用排序函数

sort(input, SIZE);

//输出排序后的字符串

cout << "排序后的字符串" << endl;

counter = 0;

while (counter<SIZE)

{

cout << "第" << counter + 1 << "个字符串：" << input[counter] << endl; //输出数据

++counter;

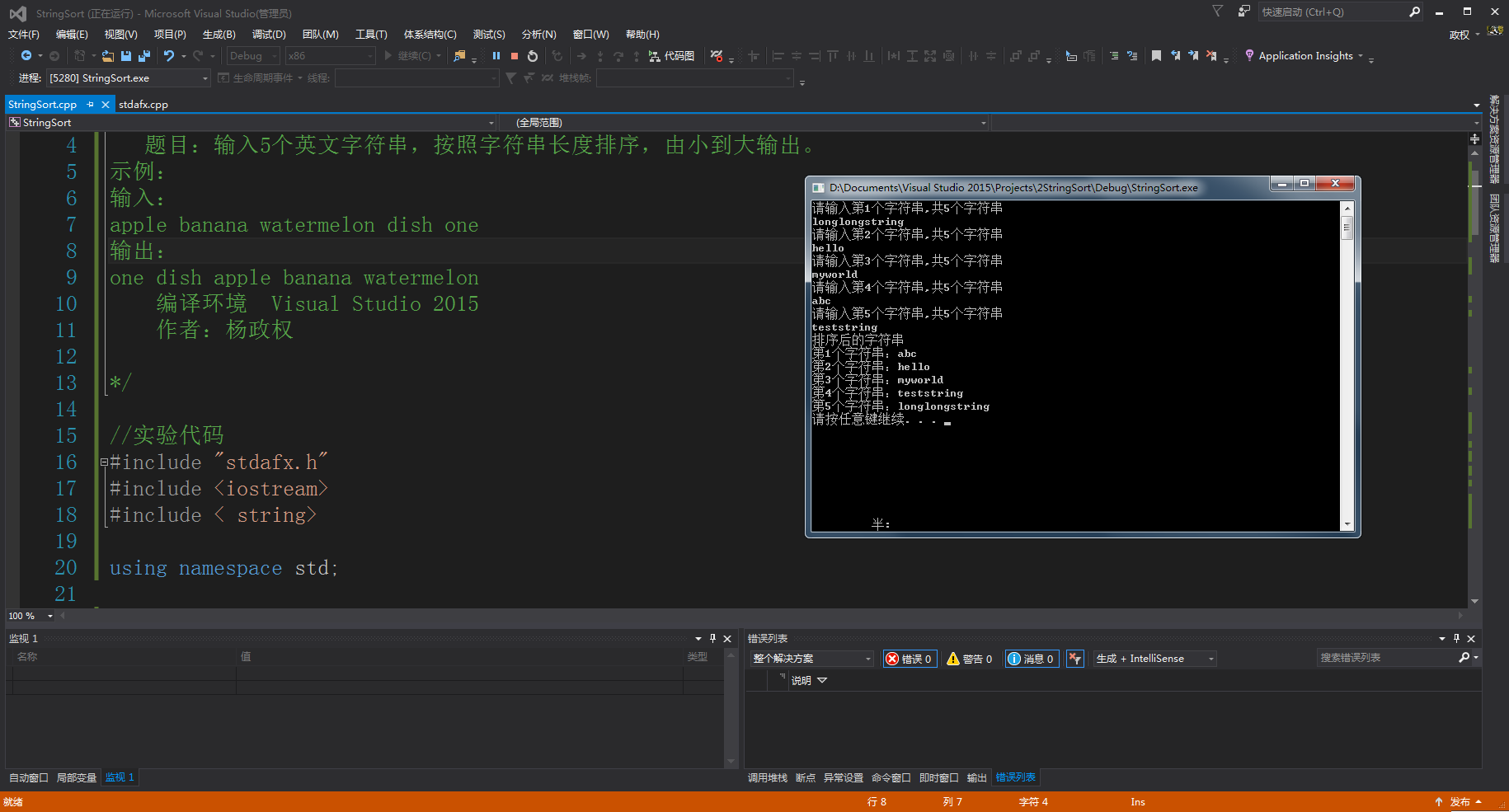
}

system("pause");

return 0;

}

//实验运行截图



/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\* 实验总结：

\*\*\* 主要用了冒泡排序来给字符串升序排序，

\*\*\* strcpy()函数不安全了，

\*\*\* 尽量使用strcpy\_s()函数，

\*\*\* 用strcpy()函数，编译器会报错，

\*\*\* 要对编译器进行设置，忽视此条警告

\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/