

**实验十五**

**综合练习2**

学期：2024-2025 第一学期

编制日期：2024 年 11月 28 日

编制人：江家玮

学号：22281188

班级：计科2204

**实验十五**

**知识覆盖：**分支、循环、函数、字符串、指针、结构体

1. 编写程序，用循环打印如下图案：

@

@#@

@###@

@#####@

@#######@

@#####@

@###@

@#@

@

要求：打印程序在函数void picture(int line)中实现，在主程序中调用函数picture实现测试。其中，line表示字符最长行所在的行数，如上图line=5。

#include <stdio.h>

// 打印图案的函数

void picture(int line) {

// 打印上半部分

for (int i = 1; i <= line; i++) {

// 打印前面的空格

for (int j = 1; j <= line - i; j++) {

printf(" ");

}

// 打印左边的 '@'

printf("@");

// 打印中间的 '#'

for (int j = 1; j <= 2 \* i - 3; j++) {

printf("#");

}

// 打印右边的 '@'

if (i > 1) {

printf("@");

}

// 换行

printf("\n");

}

// 打印下半部分

for (int i = line - 1; i >= 1; i--) {

// 打印前面的空格

for (int j = 1; j <= line - i; j++) {

printf(" ");

}

// 打印左边的 '@'

printf("@");

// 打印中间的 '#'

for (int j = 1; j <= 2 \* i - 3; j++) {

printf("#");

}

// 打印右边的 '@'

if (i > 1) {

printf("@");

}

// 换行

printf("\n");

}

}

int main() {

int line;

// 输入图案的最长行所在的行数

printf("请输入图案的最长行所在的行数: ");

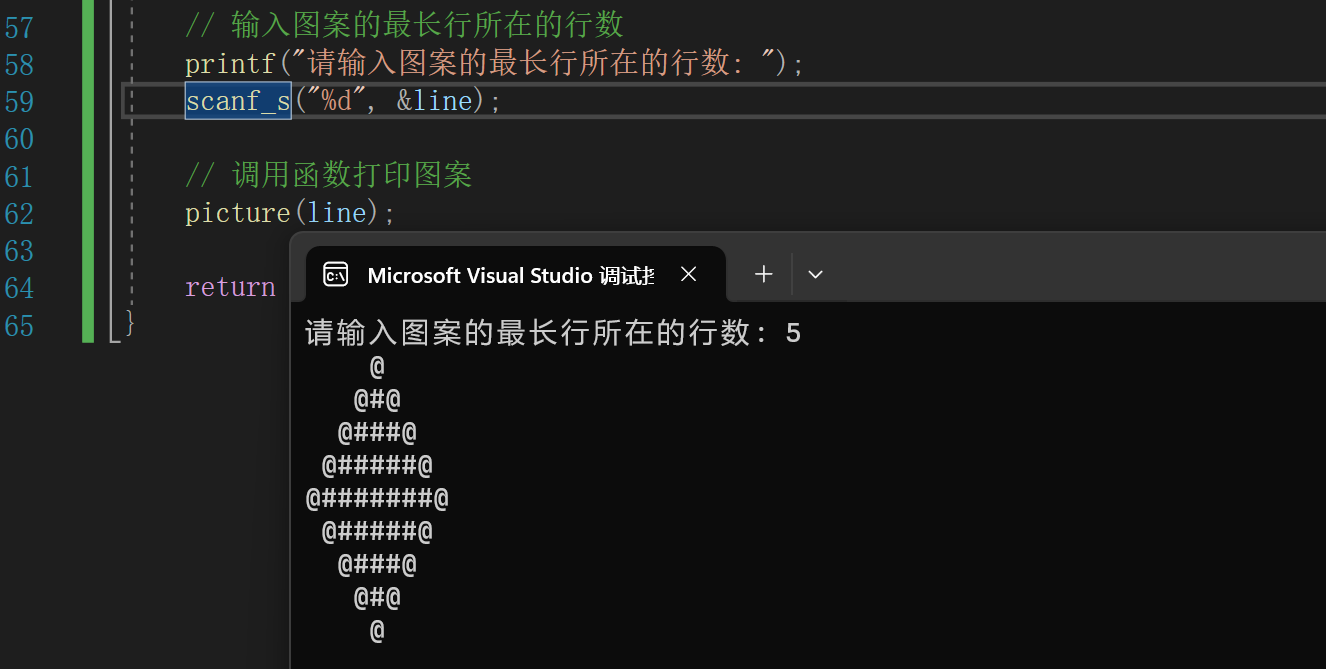
scanf\_s("%d", &line);

// 调用函数打印图案

picture(line);

return 0;

}

2. 从键盘输入2个字符串，统计子字符串在另一个字符串中出现的次数。统计过程由函数实现，在主程序中调用，其中函数原型为 int proc( char \*str, char \*sub)或int proc( char str[], char sub[])，该函数可以统计一个子字符串在另一个字符串中出现的次数。例如，假定输入的字符串为：asd ascasdfg asd asm mlosd, 子字符串为asd, 则应输出3。为方便起见，这里字符串的长度不超过100，子字符串长度不超过原字符串长度。

#include <stdio.h>

#include <string.h>

// 统计子字符串在字符串中出现的次数

int proc(char\* str, char\* sub) {

int count = 0;

int subLen = strlen(sub);

// 遍历字符串，查找子字符串

for (int i = 0; str[i] != '\0'; i++) {

// 检查从当前位置开始的子字符串是否匹配

if (strncmp(&str[i], sub, subLen) == 0) {

count++;

i += subLen - 1; // 跳过已匹配的子字符串

}

}

return count;

}

int main() {

char str[100], sub[100];

// 输入字符串

printf("请输入字符串: ");

fgets(str, sizeof(str), stdin);

str[strcspn(str, "\n")] = '\0'; // 去除fgets读取的换行符

// 输入子字符串

printf("请输入子字符串: ");

fgets(sub, sizeof(sub), stdin);

sub[strcspn(sub, "\n")] = '\0'; // 去除fgets读取的换行符

// 调用函数统计子字符串出现的次数

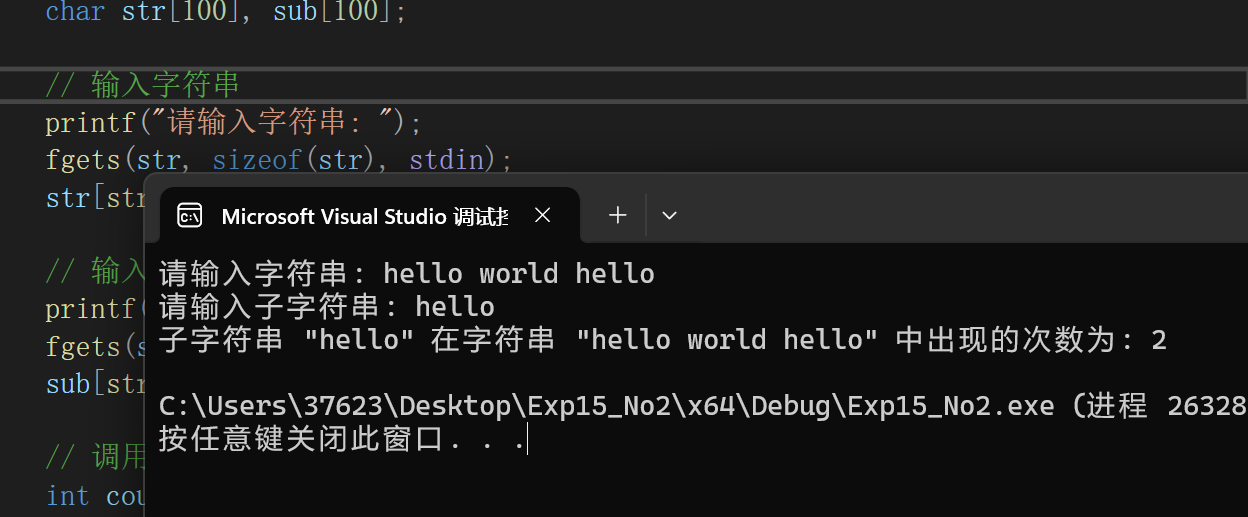
int count = proc(str, sub);

// 输出结果

printf("子字符串 \"%s\" 在字符串 \"%s\" 中出现的次数为: %d\n", sub, str, count);

return 0;

}

3. 请编写一个函数void convert(char \*p)，将一个字符串中每一个单词的第一个字母转换为大写（字符串全部由小写字母组成，任意两个单词之间由一个空格分开）。在主程序中调用该函数并输出转换后的结果。

例如：输入you are a student，输出为You Are A Student。

#include <stdio.h>

#include <ctype.h>

#include <string.h>

// 将字符串中每一个单词的第一个字母转换为大写

void convert(char\* p) {

int len = strlen(p);

int i = 0;

// 处理第一个单词的第一个字母

if (len > 0) {

p[0] = toupper(p[0]);

}

// 遍历字符串，将每个单词的第一个字母转换为大写

for (i = 1; i < len; i++) {

if (p[i - 1] == ' ' && p[i] != ' ') {

p[i] = toupper(p[i]);

}

}

}

int main() {

char str[100];

// 输入字符串

printf("请输入一个由小写字母组成的字符串: ");

fgets(str, sizeof(str), stdin);

str[strcspn(str, "\n")] = '\0'; // 去除fgets读取的换行符

// 调用函数进行转换

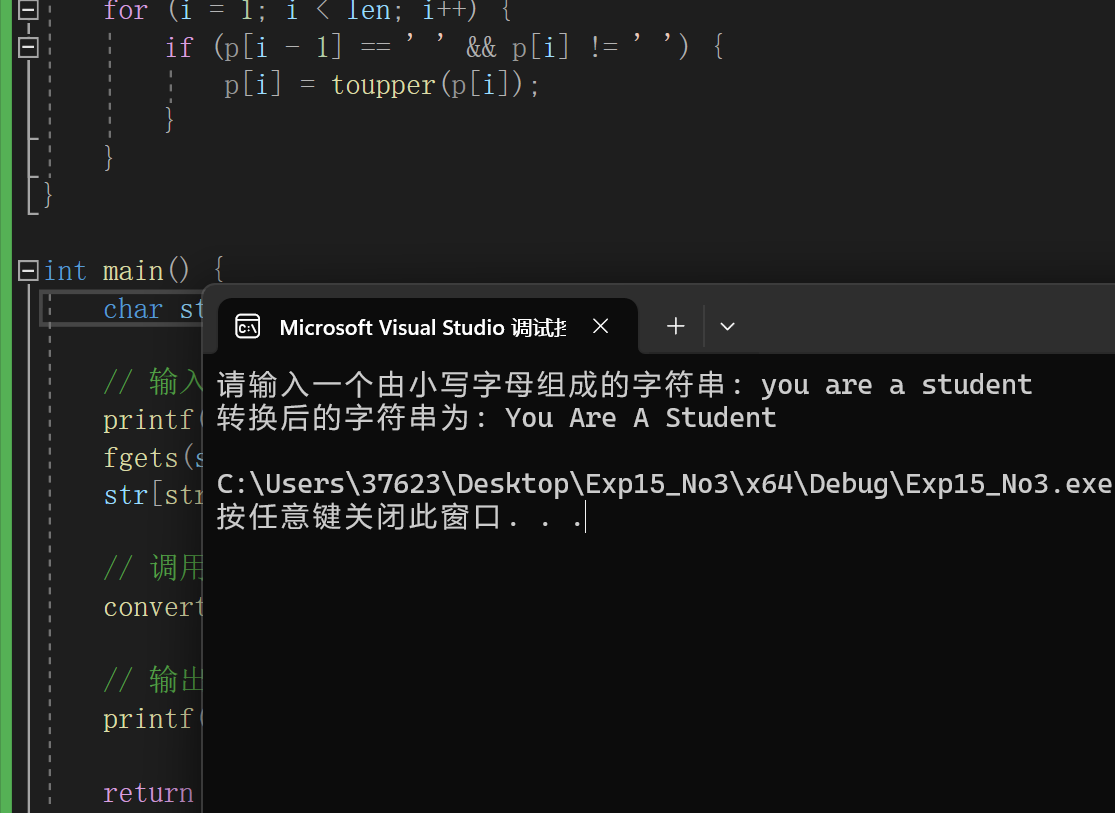
convert(str);

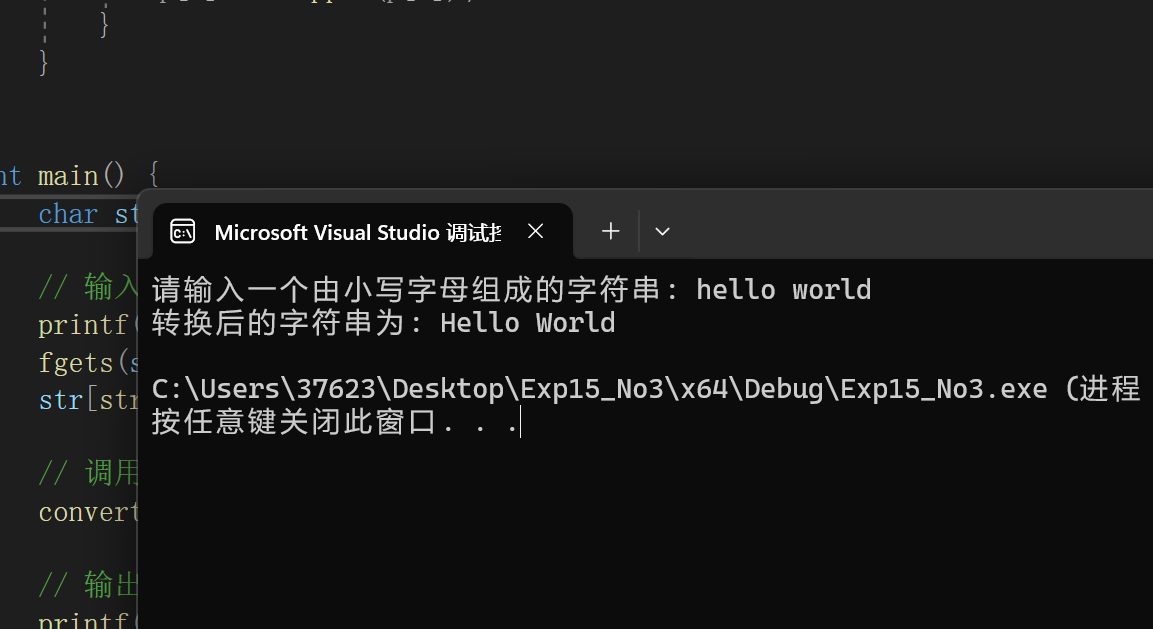
// 输出转换后的字符串

printf("转换后的字符串为: %s\n", str);

return 0;

}





4. 编写一个函数，把字符串中的所有数字字符转换成整数，并去掉其他字符。例如， 字符串“3a56bc”，转换为整数后是 356。要求编写子函数实现转换功能，该函数返回值为int型整数。函数原型为

int ConvertInt(char \*str)； 或者 int ConvertInt(char str[])；（提示：字符0对应的ASCII码为48，字符1对应的ASCII码为49，以此类推）

#include <stdio.h>

#include <ctype.h>

#include<string>

#include<string.h>

// 将字符串中的所有数字字符转换成整数

int ConvertInt(char\* str) {

int result = 0;

// 遍历字符串，将数字字符转换为整数

for (int i = 0; str[i] != '\0'; i++) {

if (isdigit(str[i])) { // 检查字符是否为数字

result = result \* 10 + (str[i] - '0'); // 将字符转换为整数并累加

}

}

return result;

}

int main() {

char str[100];

// 输入字符串

printf("请输入一个包含数字字符的字符串: ");

fgets(str, sizeof(str), stdin);

str[strcspn(str, "\n")] = '\0'; // 去除fgets读取的换行符

// 调用函数进行转换

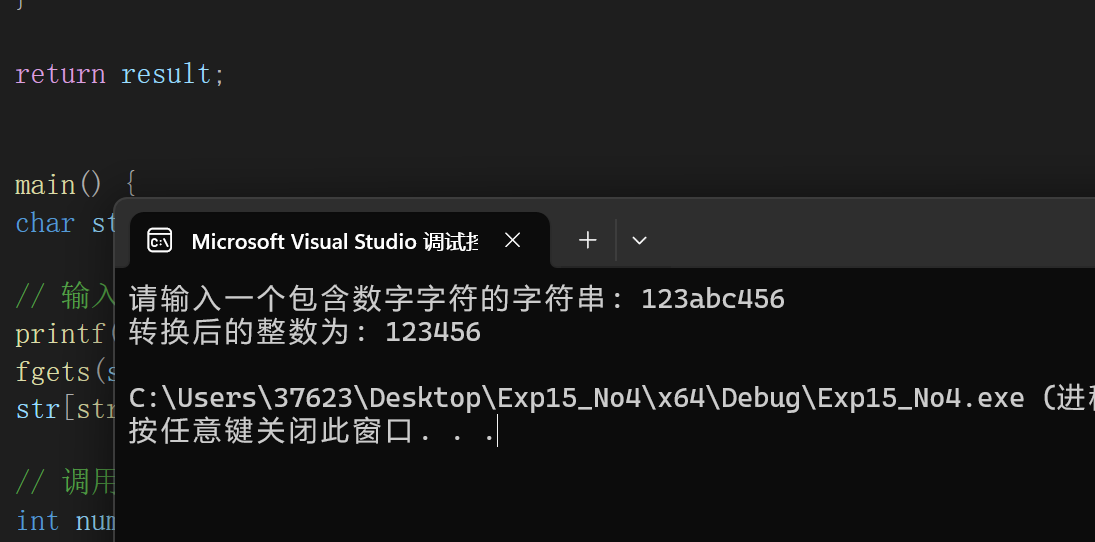
int number = ConvertInt(str);

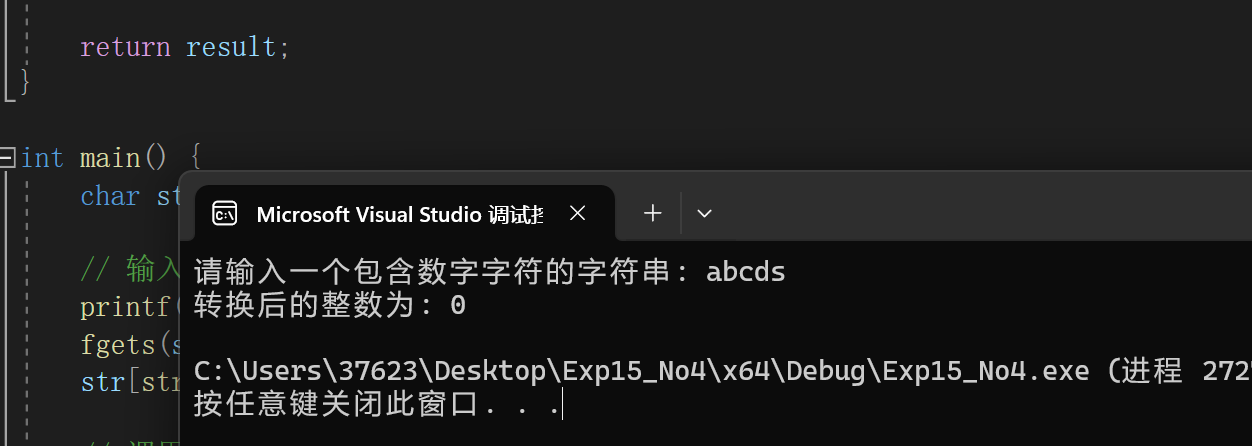
// 输出转换后的整数

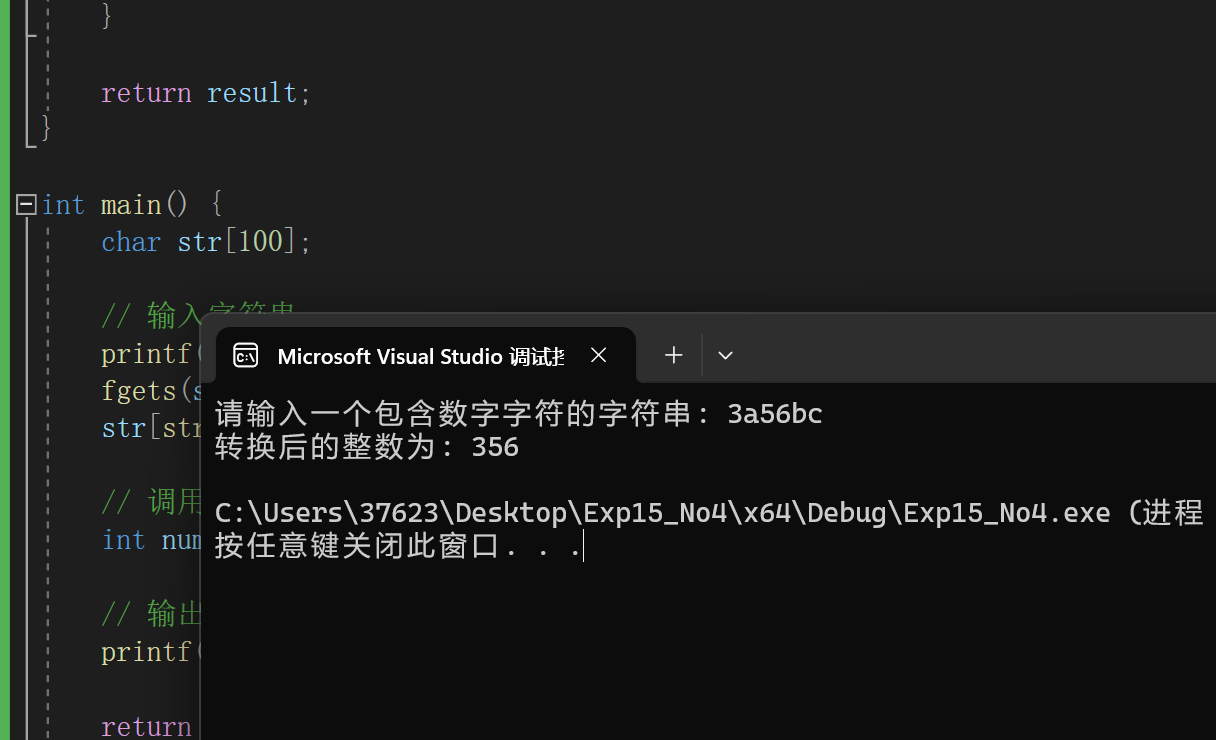
printf("转换后的整数为: %d\n", number);

return 0;

}







5. 设有如下学生结构：

typedef struct{

char szName[20];//姓名

double dScore[2]; //两门课的成绩

double avgScore; //两门课的平均成绩

} STUD;

设班里有一批学生成绩如下(以下是测试样例，可以有多名学生)：

3

LiMing 97 51

WangTao 71 85

ZhaoLi 45 71

输入包含nLen +1行：第一行是一个整数nLen（0 < nLen <= 10）, nLen个学生。第二行至nLen +1行是nLen个学生的记录: 姓名 分数，每个学生占一行，姓名与分数之间用一个空格隔开。

请编写如下函数：double studLowestAvgScore(STUD \*pStu, int nLen)，其中nLen为班级人数。该函数需要求该班所有学生两门课的平均成绩，并在其对应结构体成员变量avgScore中进行存储，打印具有最低平均分的学生的所有相关信息，同时返回所有学生中的最低平均分。在主程序中测试，并输出最低平均分。

#define \_CRT\_SECURE\_NO\_WARNINGS 1

#include <stdio.h>

#include <string.h>

// 定义学生结构

typedef struct {

char szName[20]; // 姓名

double dScore[2]; // 两门课的成绩

double avgScore; // 两门课的平均成绩

} STUD;

// 计算并返回班级中最低平均分，同时打印具有最低平均分的学生的信息

double studLowestAvgScore(STUD\* pStu, int nLen) {

double lowestAvg = pStu[0].dScore[0] + pStu[0].dScore[1]; // 初始化最低平均分为第一个学生的总分

int lowestIndex = 0; // 记录最低平均分的学生的索引

// 计算每个学生的平均成绩，并找出最低平均分

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

pStu[i].avgScore = (pStu[i].dScore[0] + pStu[i].dScore[1]) / 2.0;

if (pStu[i].avgScore < lowestAvg) {

lowestAvg = pStu[i].avgScore;

lowestIndex = i;

}

}

// 打印具有最低平均分的学生的信息

printf("具有最低平均分的学生的信息:\n");

printf("姓名: %s\n", pStu[lowestIndex].szName);

printf("第一门课成绩: %.2lf\n", pStu[lowestIndex].dScore[0]);

printf("第二门课成绩: %.2lf\n", pStu[lowestIndex].dScore[1]);

printf("平均成绩: %.2lf\n", pStu[lowestIndex].avgScore);

return lowestAvg;

}

int main() {

int nLen; // 学生人数

STUD students[10]; // 定义学生数组，最多10个学生

char line[100]; // 用于存储每行输入

// 输入学生人数

printf("请输入学生人数 (0 < nLen <= 10): ");

scanf\_s("%d", &nLen);

getchar(); // 读取换行符，避免影响后续输入

// 输入每个学生的信息

printf("请输入每个学生的姓名和两门课的成绩:\n");

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

printf("第 %d 个学生: ", i + 1);

fgets(line, sizeof(line), stdin); // 读取一行输入

sscanf(line, "%s %lf %lf", students[i].szName, &students[i].dScore[0], &students[i].dScore[1]);

}

// 调用函数计算并输出最低平均分

double lowestAvg = studLowestAvgScore(students, nLen);

printf("所有学生中的最低平均分为: %.2lf\n", lowestAvg);

return 0;

}

