# 实验8：结构体

## 【实验目的及要求】

1．掌握C语言的结构体

2. 完成实验报告

## 【实验内容】

1. 习题九第2题da29225e873ed9fb2638a5d039b6229
2. 输入30个学生的信息（姓名、年龄、5门功课成绩），然后按照平均成绩从高到低输出，要求：

（1）、自定义输入函数 void read(student \*x)

（2）、自定义输出函数 void write(student x)

请将代码附在下面：

# 1.习题9.2

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

struct Contact {

char name[50]; char phone[20];

};

int main() {

int n;

printf("输入联系人数： ");

scanf("%d", &n);

struct Contact \*contacts = (struct Contact \*)malloc(n \* sizeof(struct Contact));

if (contacts == NULL) {

fprintf(stderr, "Memory allocation failed\n");

return 1;

}

while (getchar() != '\n');

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

printf("输入第 %d 位联系人的姓名和电话：\n", i + 1);

printf("姓名: ");

fgets(contacts[i].name, sizeof(contacts[i].name), stdin);

contacts[i].name[strcspn(contacts[i].name, "\n")] = '\0';

printf("电话号码: ");

fgets(contacts[i].phone, sizeof(contacts[i].phone), stdin);

contacts[i].phone[strcspn(contacts[i].phone, "\n")] = '\0';

}

for (int i = 0; i < n - 1; i++) {

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

if (strcmp(contacts[j].phone, contacts[j + 1].phone) > 0) {

struct Contact temp = contacts[j];

contacts[j] = contacts[j + 1];

contacts[j + 1] = temp;

}

}

}

printf("\n联系人的姓名和手机号为：:\n");

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

printf("%s, \t %s\n", contacts[i].name, contacts[i].phone);

}

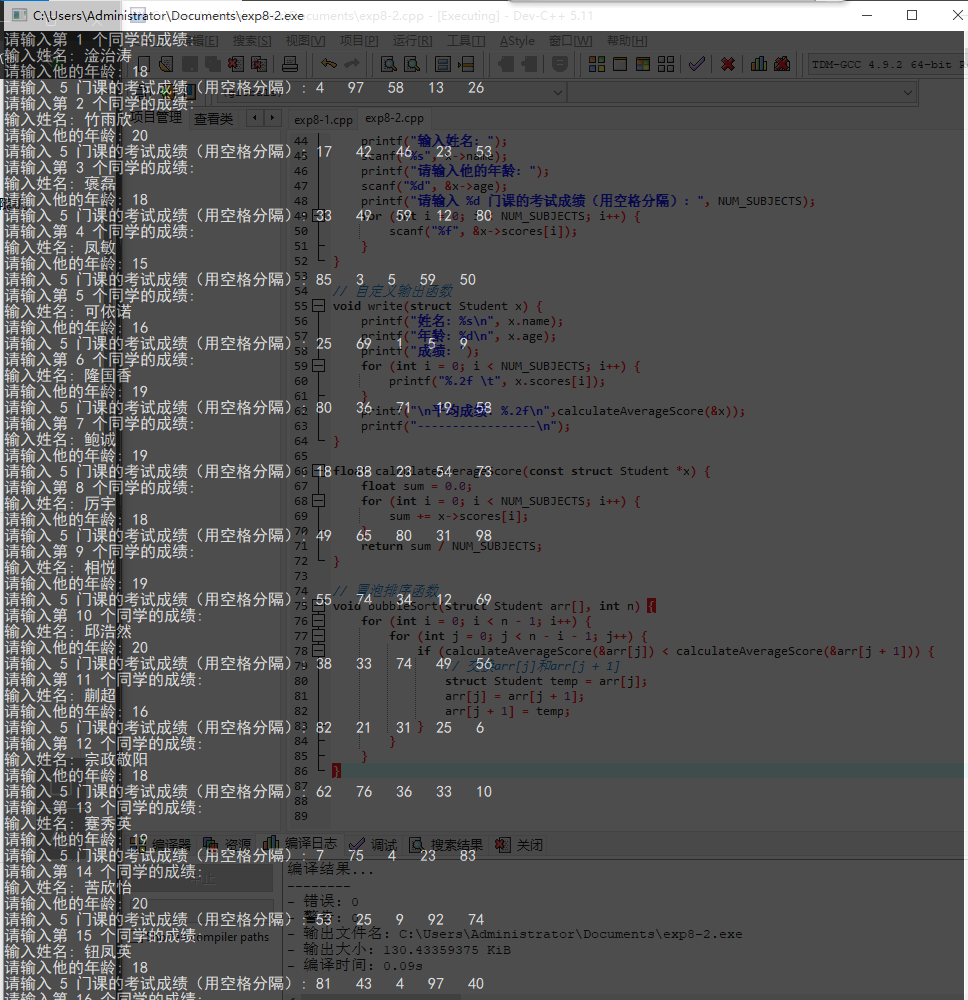
free(contacts);

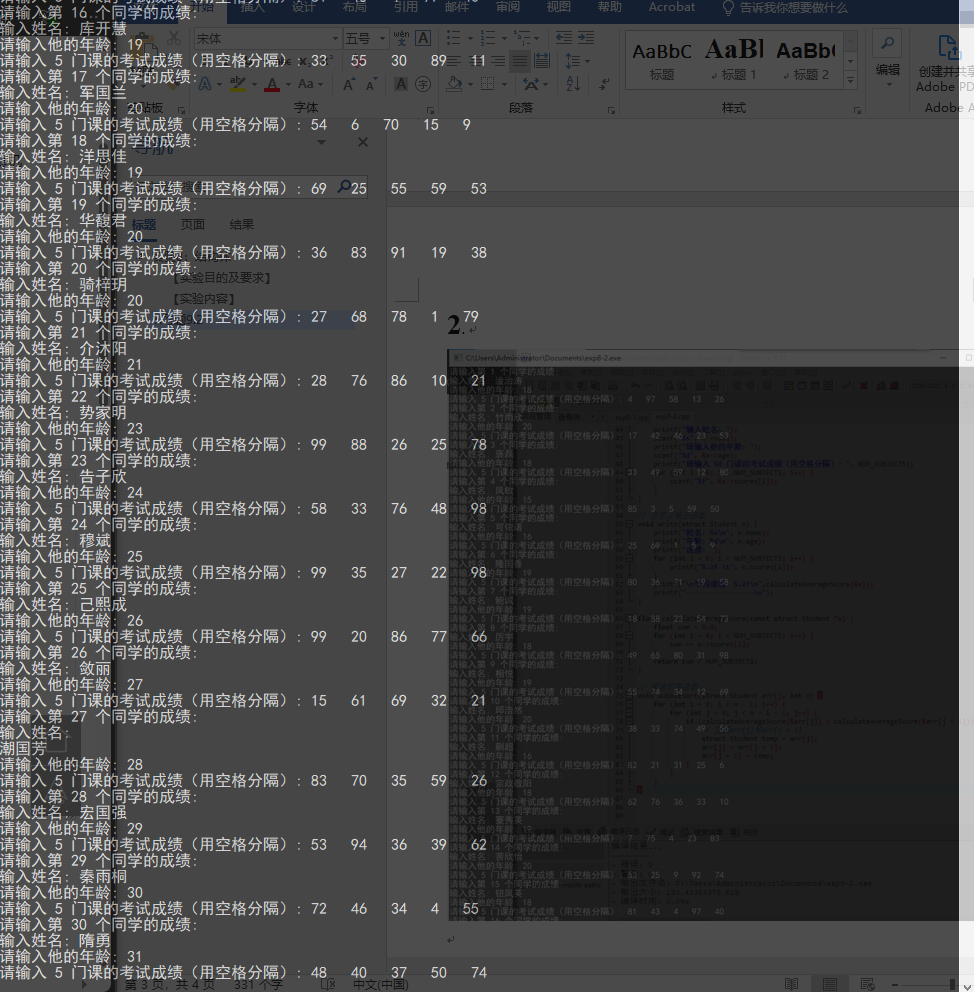
return 0;

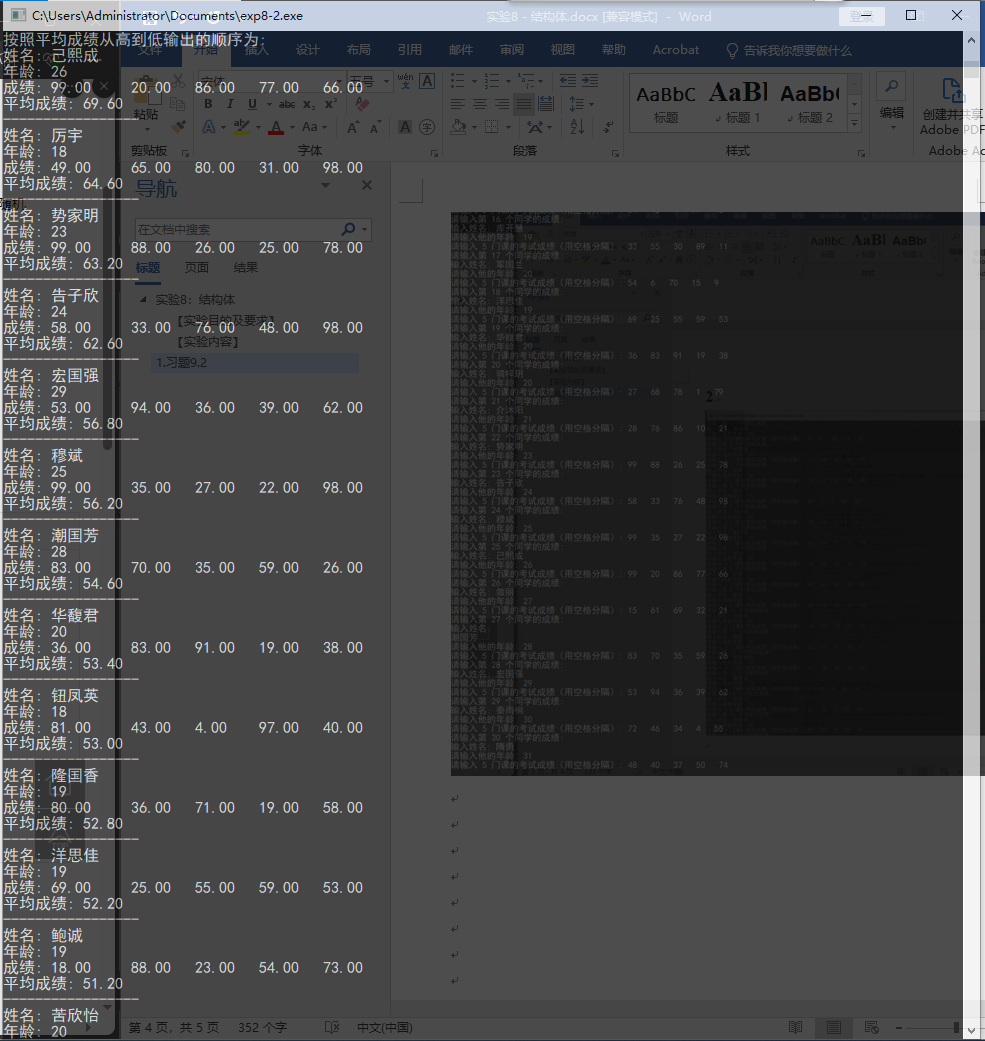
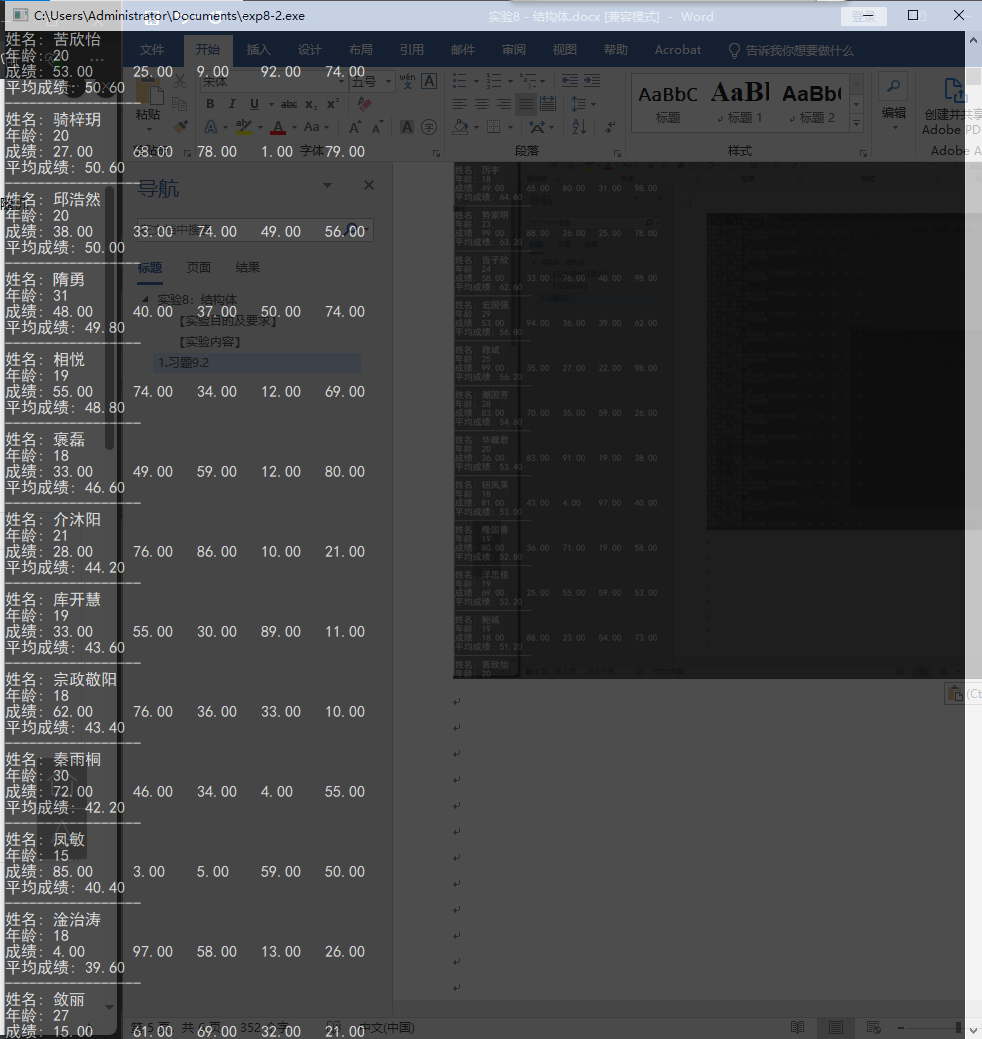
}

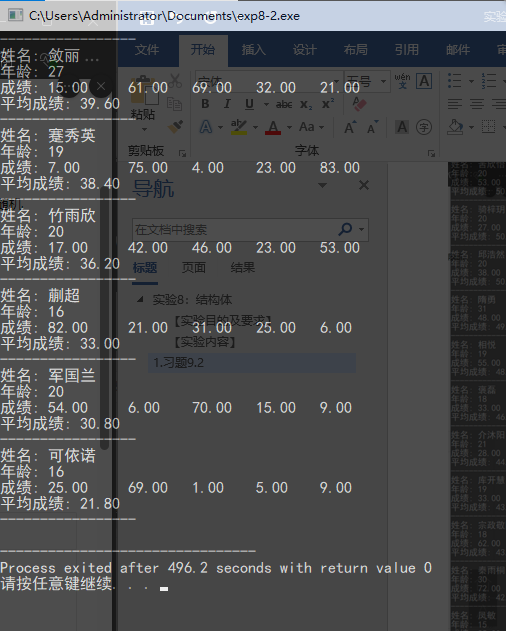
# 2.好长的输入和两个要求

不得不说 这道题的输入给我带来了非常大的麻烦









#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define NUM\_STUDENTS 30

#define NUM\_SUBJECTS 5

// 定义结构体

struct Student {

char name[50];

int age;

float scores[NUM\_SUBJECTS];

};

// 声明函数原型

void read(struct Student \*x);

void write(struct Student x);

float calculateAverageScore(const struct Student \*x);

void bubbleSort(struct Student arr[], int n);

int main() {

struct Student students[NUM\_STUDENTS];

// 输入每个学生的信息

for (int i = 0; i < NUM\_STUDENTS; i++) {

printf("请输入第 %d 个同学的成绩:\n", i + 1);

read(&students[i]);

}

// 按照平均成绩从高到低排序

bubbleSort(students, NUM\_STUDENTS);

// 输出排序后的信息

printf("\n按照平均成绩从高到低输出的顺序为:\n");

for (int i = 0; i < NUM\_STUDENTS; i++) {

write(students[i]);

}

return 0;

}

// 自定义输入函数

void read(struct Student \*x) {

printf("输入姓名: ");

scanf("%s", x->name);

printf("请输入他的年龄: ");

scanf("%d", &x->age);

printf("请输入 %d 门课的考试成绩（用空格分隔）: ", NUM\_SUBJECTS);

for (int i = 0; i < NUM\_SUBJECTS; i++) {

scanf("%f", &x->scores[i]);

}

}

// 自定义输出函数

void write(struct Student x) {

printf("姓名: %s\n", x.name);

printf("年龄: %d\n", x.age);

printf("成绩: ");

for (int i = 0; i < NUM\_SUBJECTS; i++) {

printf("%.2f \t", x.scores[i]);

}

printf("\n平均成绩: %.2f\n",calculateAverageScore(&x));

printf("-----------------\n");

}

float calculateAverageScore(const struct Student \*x) {

float sum = 0.0;

for (int i = 0; i < NUM\_SUBJECTS; i++) {

sum += x->scores[i];

}

return sum / NUM\_SUBJECTS;

}

// 冒泡排序函数

void bubbleSort(struct Student arr[], int n) {

for (int i = 0; i < n - 1; i++) {

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

if (calculateAverageScore(&arr[j]) < calculateAverageScore(&arr[j + 1])) {

// 交换arr[j]和arr[j + 1]

struct Student temp = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = temp;

}

}

}

}