C语言程序设计

课程设计报告

题目： 学生成绩管理系统 我

专业： 通信工程（计算机通信网络）我

学号： 我

姓名： 我

时间： 2023/9/7 我

**一、需求分析**

设计的学生成绩管理系统系统是用户管理学生的成绩，通过文件存储成绩数据，通过交互实现输入成绩、通过姓名或学号查询成绩、修改成绩、添加输入成绩、删除成绩、排序、计算总分、输出显示等功能，并能将成绩保存为文件。

**二、概要设计**

包括模块划分，系统模块结构图

菜单界面与各模块间的结构关系图。学生成绩管理系统由十四个模块组成，系统的模块结构如图1所示。

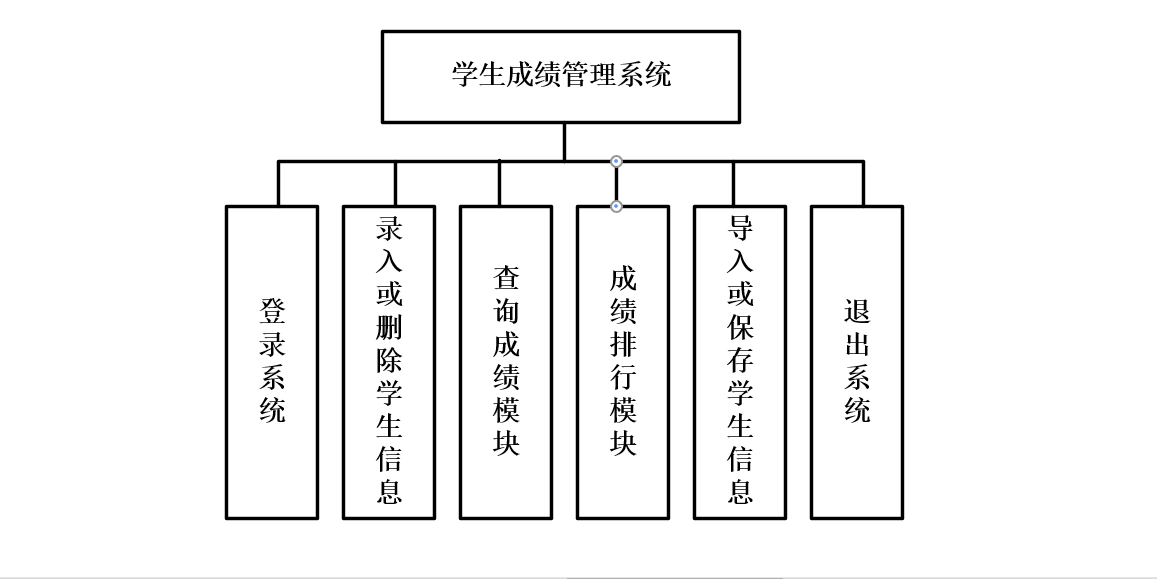


图1

**三、详细设计**

系统需要实现以下功能：

登录系统（通过输入用户名及密码判断是否可以登录）

录入学生信息

删除学生信息（通过姓名来删除学生信息）

修改学生成绩

查询学生成绩

查询所有学生的信息

查询所有学生成绩

学生成绩排序（冒泡排序）

查询单个学生

导入学生信息（通过文件导入学生信息）

保存学生信息（将学生信息保存至文件）

根据学号排序

查看班级平均分

退出系统（通过代码实现，不是Ctrl + C）

1. 初始化：开始程序，初始化学生信息数据结构和其他必要的变量。

2. 用户认证：要求用户进行登录或身份验证，通常需要输入用户名和密码。验证成功后，允许用户进入系统，否则显示错误消息并继续要求身份验证。

3. 主菜单：一旦用户登录成功，系统将显示主菜单，允许用户选择不同的功能选项。这些选项通常包括：

- 录入学生成绩

- 修改学生成绩

- 删除学生成绩

- 查询学生成绩

- 显示所有学生信息

- 按成绩排序

- 按学号排序

- 查看班级平均分

- 退出系统

4. 功能实现：根据用户的选择，执行相应的功能：

- 录入学生成绩：输入学生信息（学号、姓名、性别、年龄、成绩），将信息保存到数据结构中。

- 修改学生成绩：根据学号或姓名查找学生信息，然后修改成绩。

- 删除学生成绩：根据学号或姓名查找学生信息，然后删除该学生的记录。

- 查询学生成绩：根据学号或姓名查找学生信息，并显示学生成绩。

- 显示所有学生信息：遍历数据结构并显示所有学生的信息。

- 按成绩排序：根据成绩对学生信息进行排序并显示排名。

- 按学号排序：根据学号对学生信息进行排序并显示排名。

- 查看班级平均分：计算并显示所有学生的平均分。

- 退出系统：退出程序或返回登录界面。

5. 数据存储：为了保持学生信息，需要使用数组、链表或数据库来存储和管理数据。

6. 错误处理：在用户输入不正确数据或操作失败时，需要提供错误消息并要求用户重试或返回菜单。

7. 循环和退出：在每个功能执行后，用户可以选择返回主菜单或退出系统。

8. 保存和加载数据：为了持久化学生信息，将数据保存到文件中以便下次程序启动时加载。

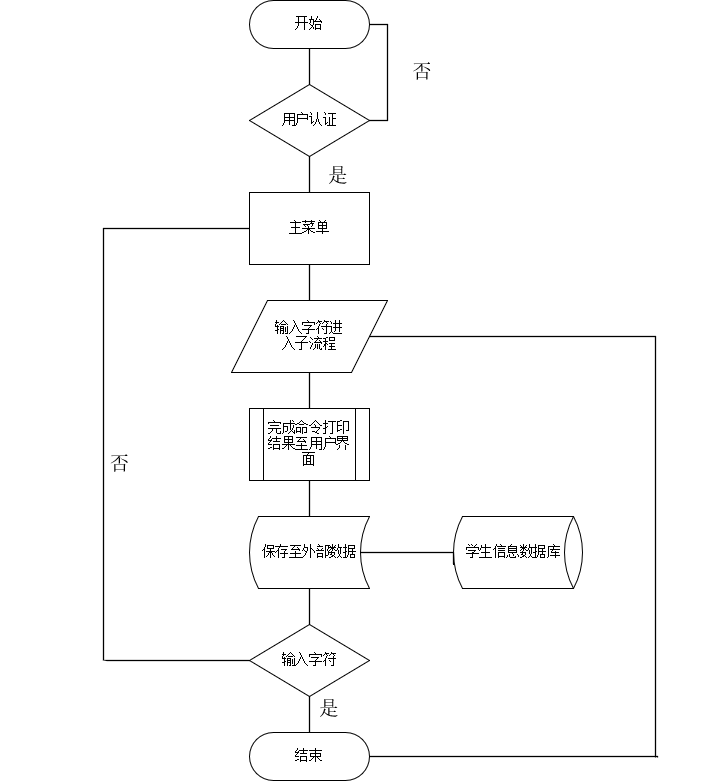


图2用户流程

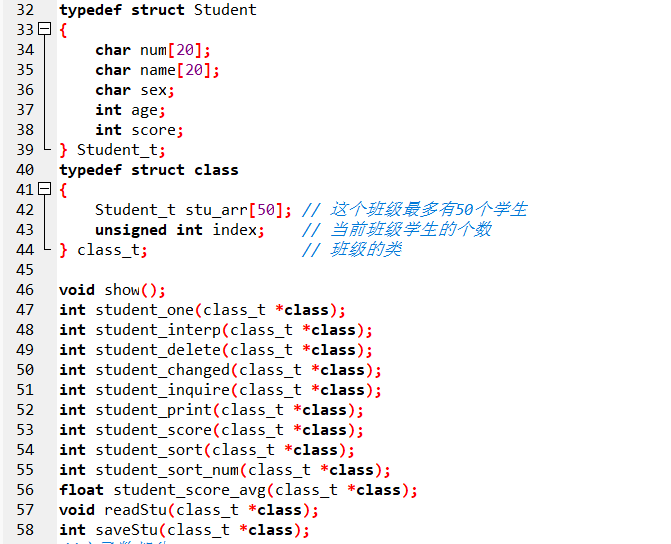
**四，功能实现**

**系统界面如下**

表格

中度可信度描述已自动生成

**定义结构体和声明函数**



**主函数**

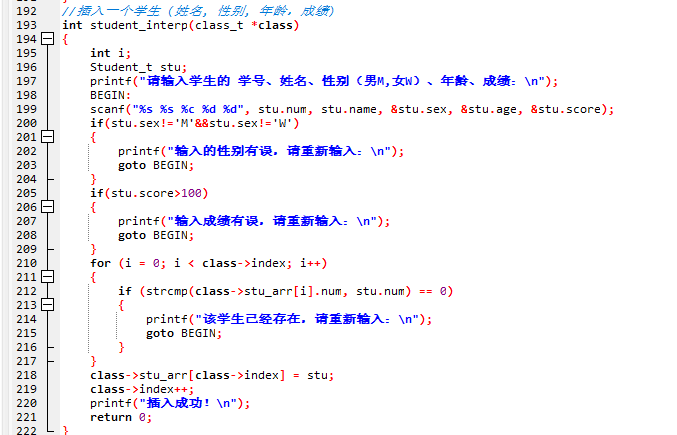
通过判断输入的账号密码来实现系统的登录，通过switch语句来控制实现不同的功能。通过while循环以及 goto语句来跳转退出系统。



**功能函数**

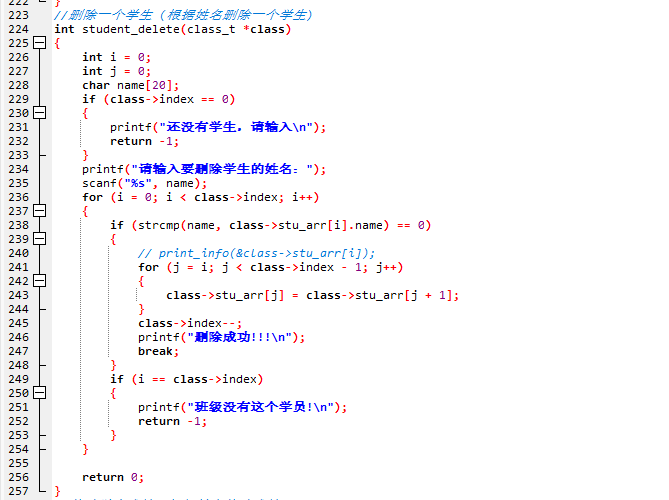
**学生信息录入**

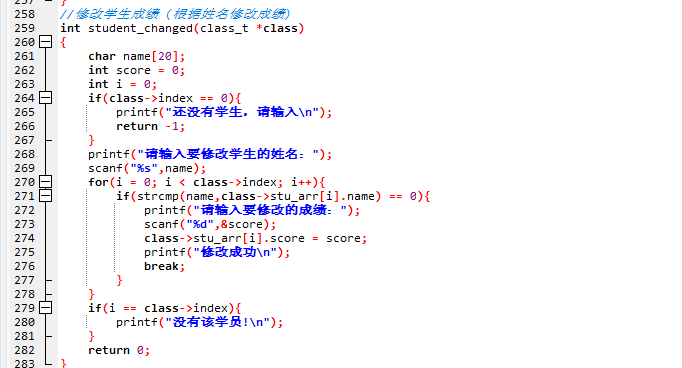
录入学生信息，如果出现相同学号的录入，则录入失败；相关代码如下

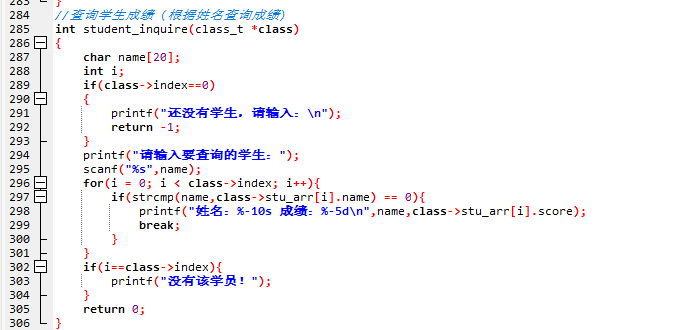


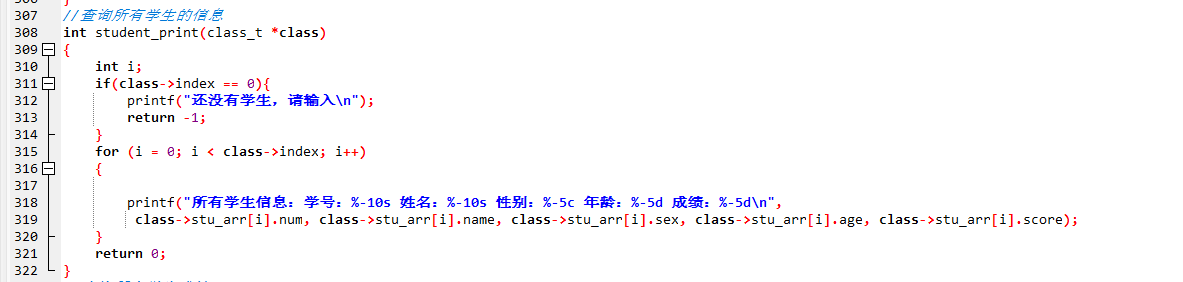
**删除学生信息**

通过输入要删除学生的姓名来删除相应的学生，先判断有无学生信息，再删除。代码如下：



**修改学生的成绩**通过输入学生的姓名，来修改此学生的成绩，代码如下：

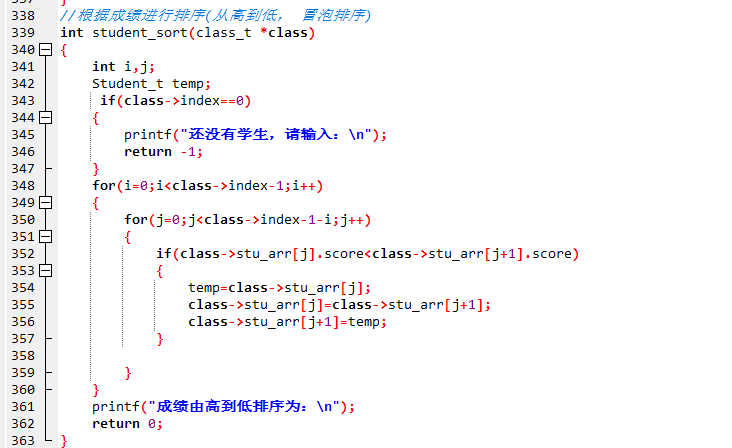
**查询学生成绩**通过姓名来查询学生的成绩；代码如下：

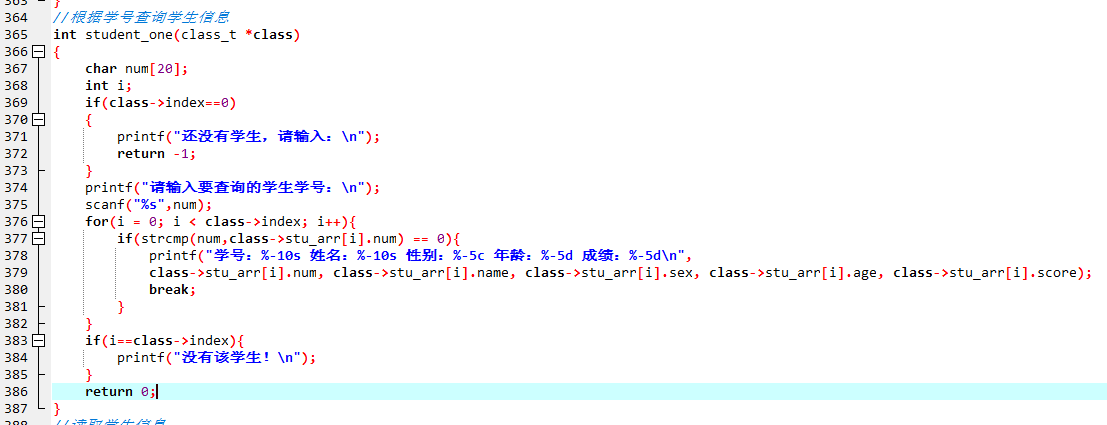
**查询所有学生的信息**遍历整个数组，把学生信息遍历出来；代码如下

**查询所有学生成绩**代码如下：

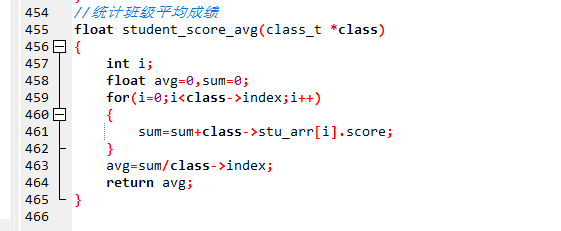
图形用户界面, 文本

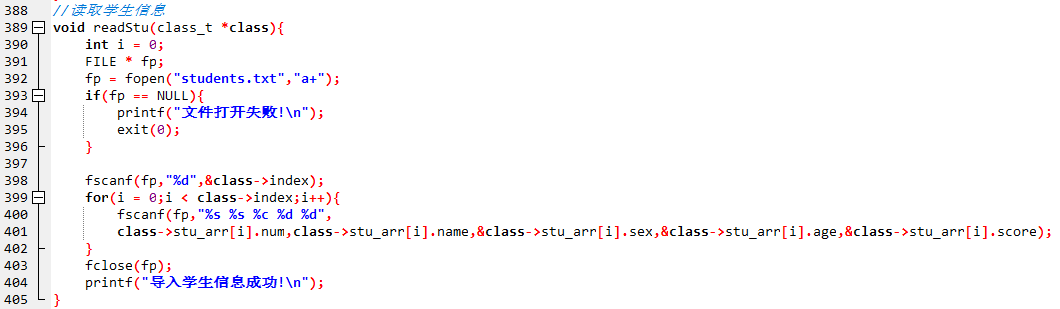
描述已自动生成

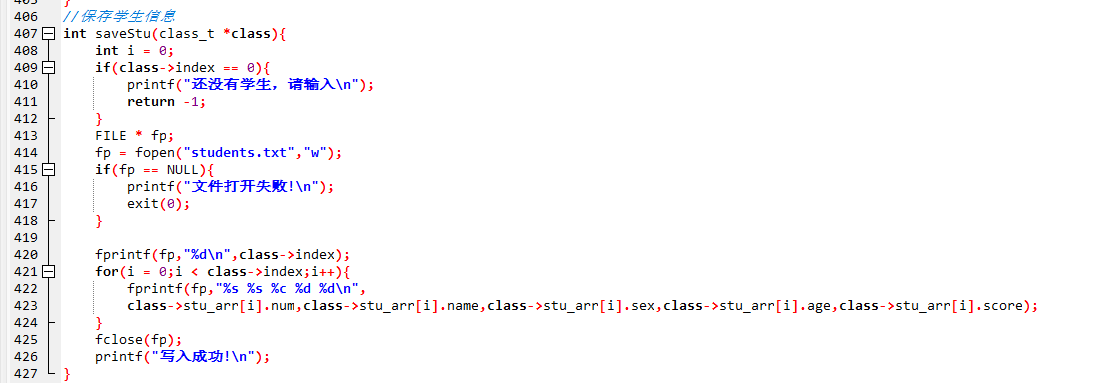
**根据成绩进行排序**使用冒泡排序，对学生的成绩进行排序，由高到低输出，定义一个学生变量temp进行交换；代码如下：

**按学号排序**同上，按学号由低到高排序，注意的是这里学号是字符串，用strcmp函数比较；代码如下：

**统计班级的平均成绩**计算班级所有学生的成绩，算出平均成绩；代码如下；

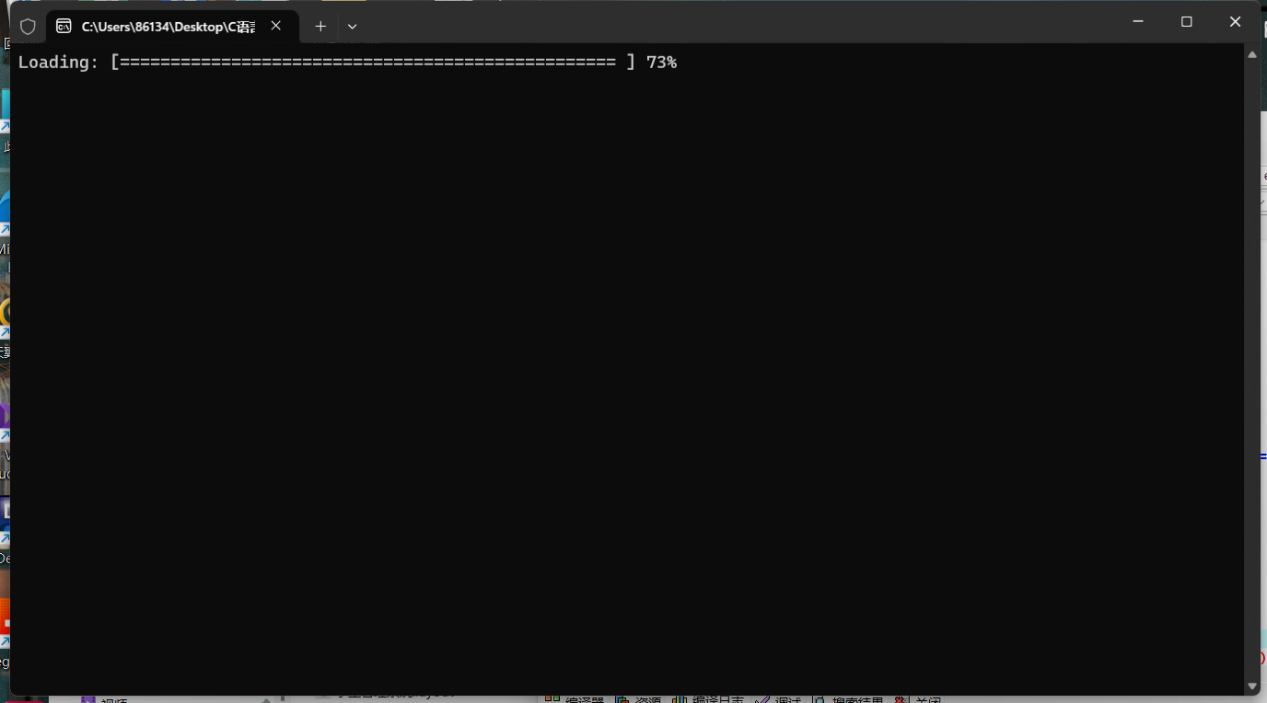


**读取学生的信息**实现从文件中导入学生信息，将文件的信息作为输入。代码如下：

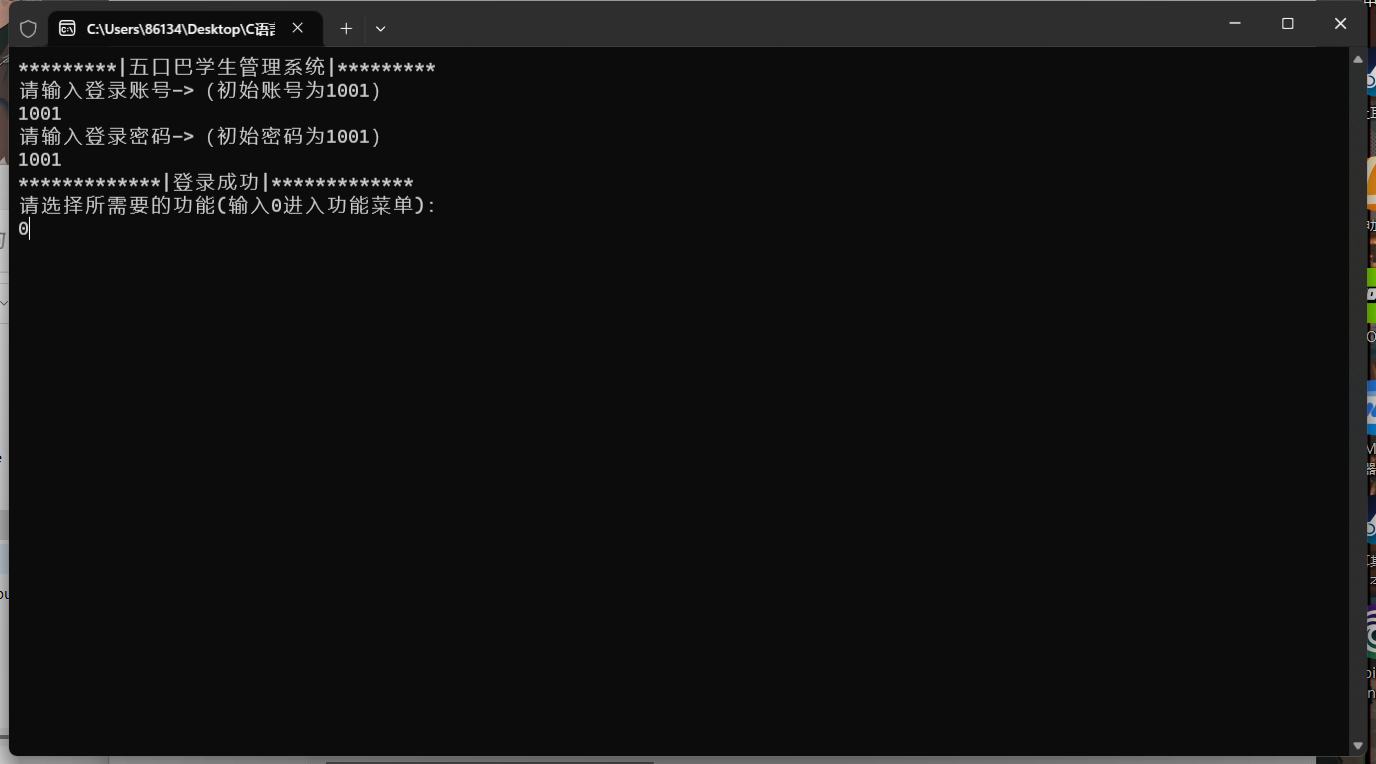
**保存文件信息**将系统输入的学生信息写入到文件中；代码如下：

**五，效果展示**

**初始化动画**



**登陆界面**



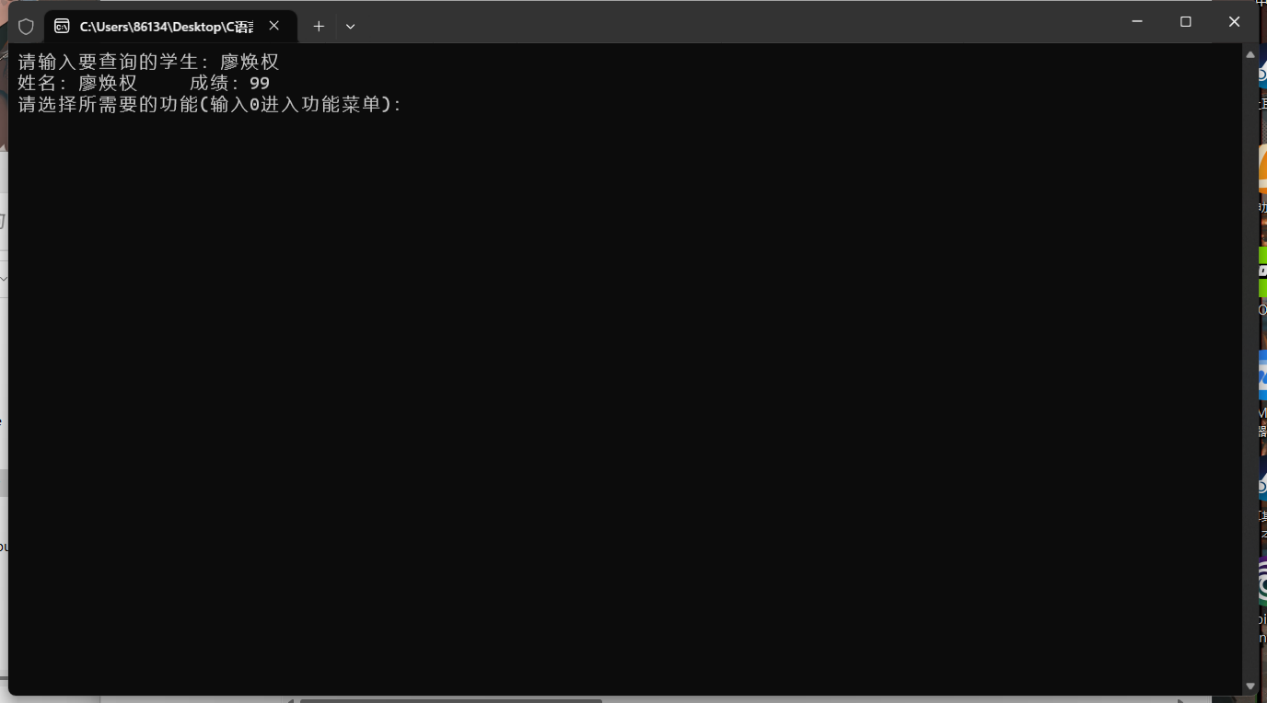
**功能菜单**



**录入学生信息**



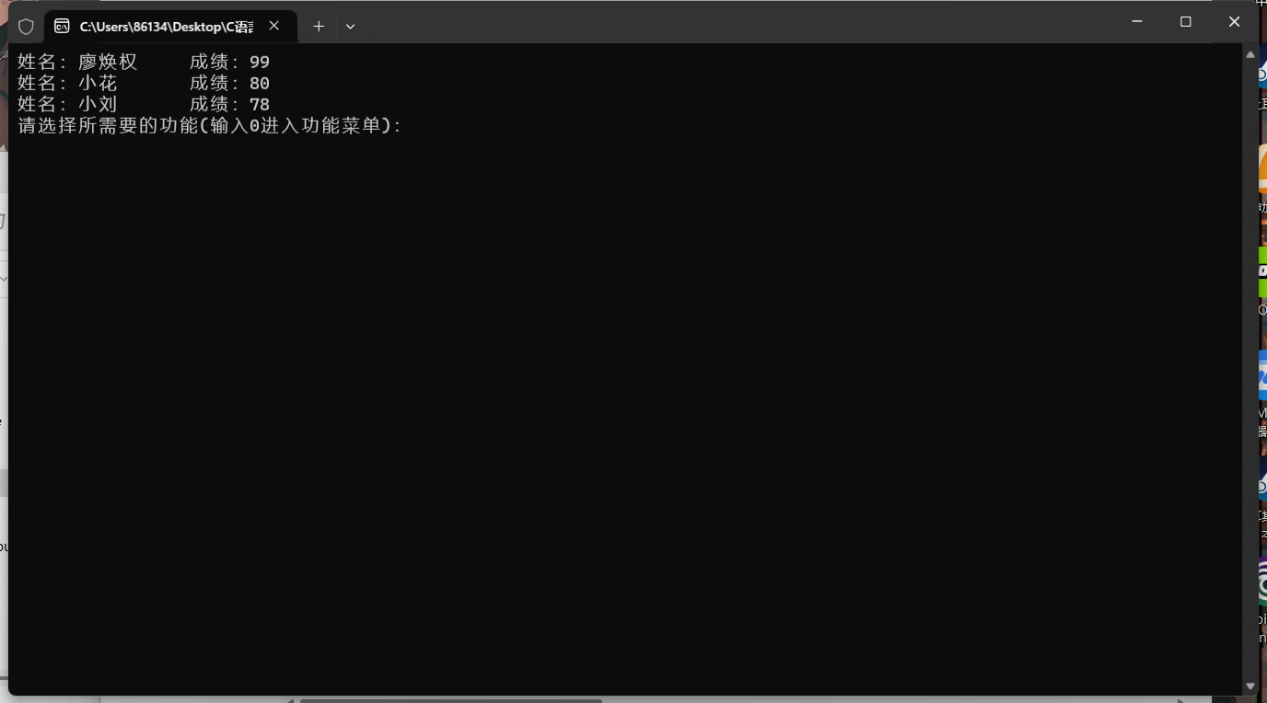
**查询学生信息**



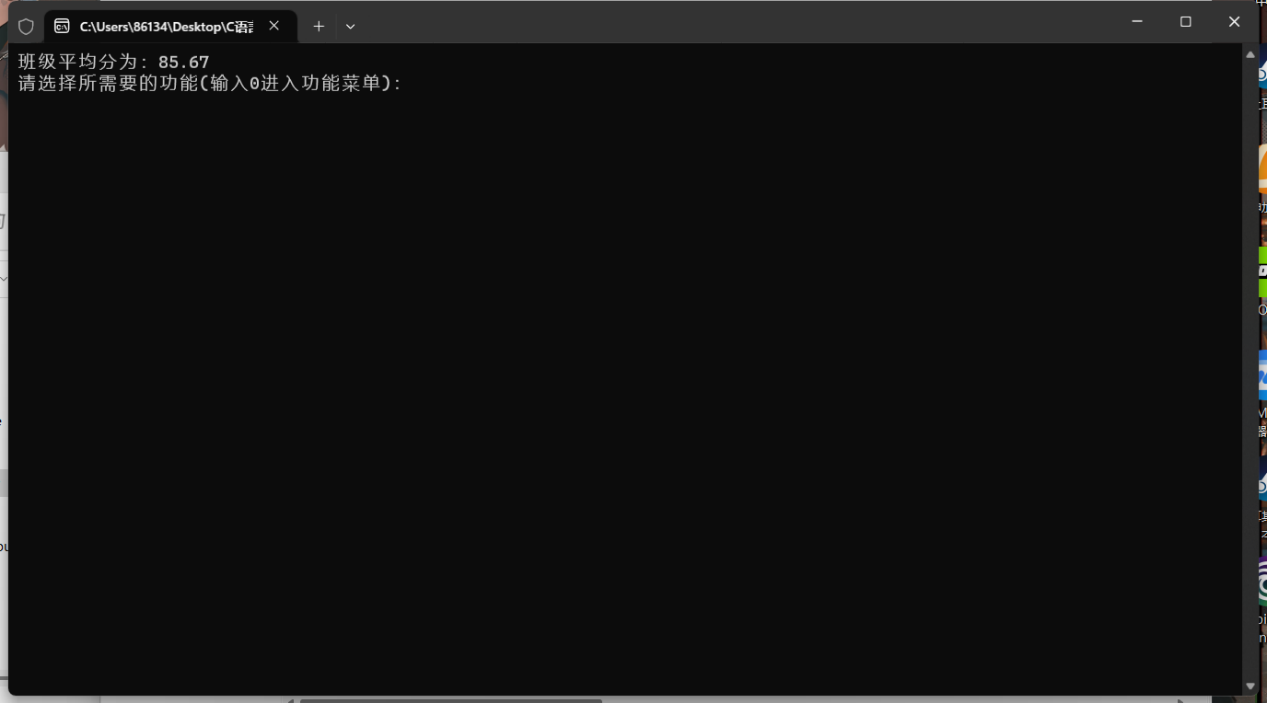
**按学号查询**



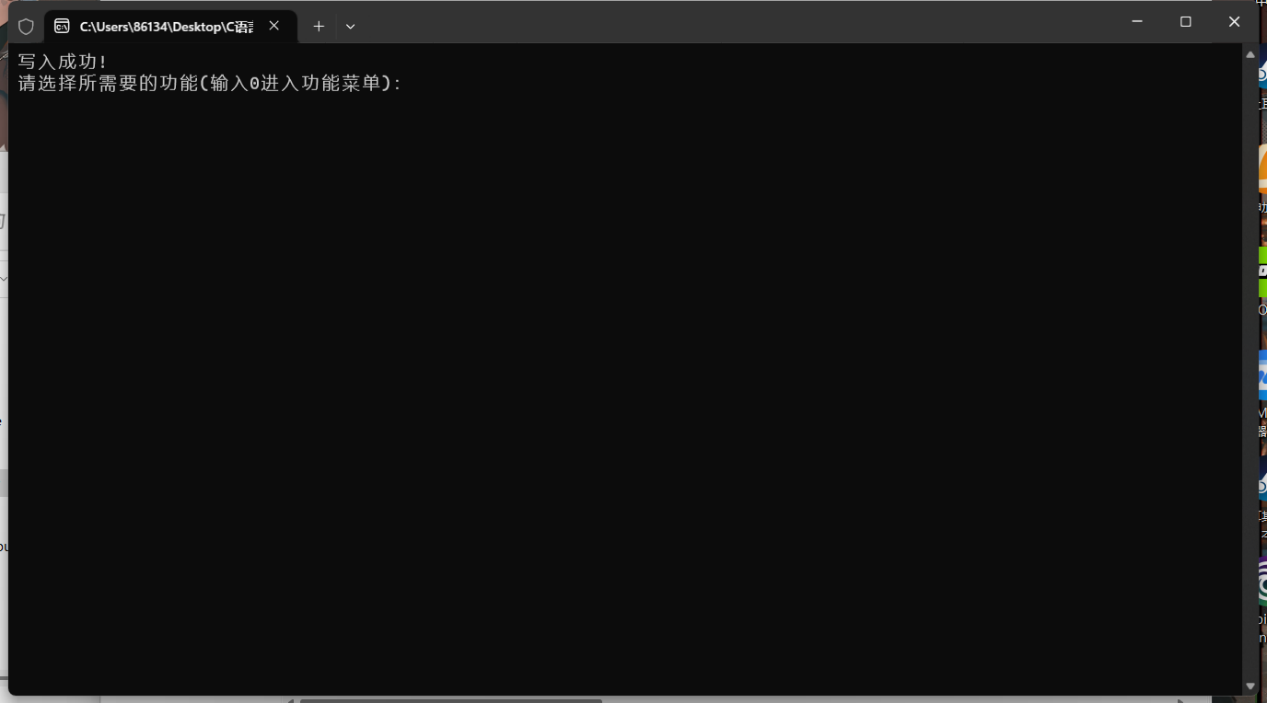
**所有学生的成绩**



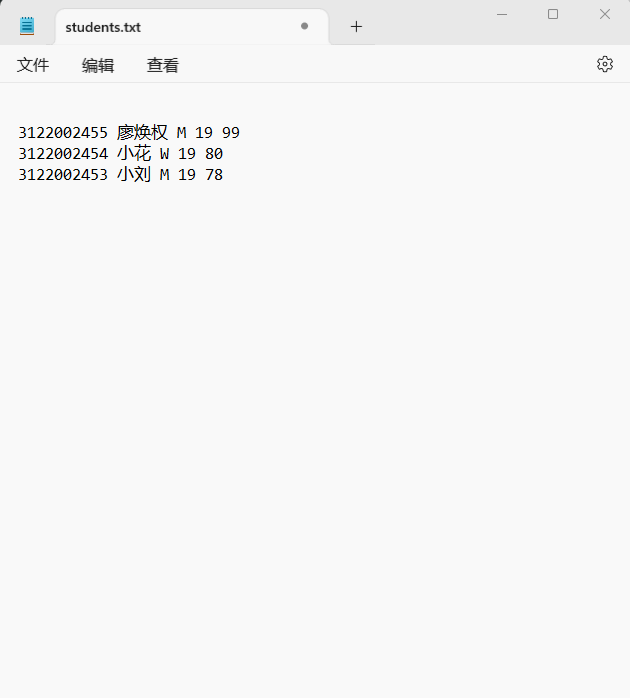
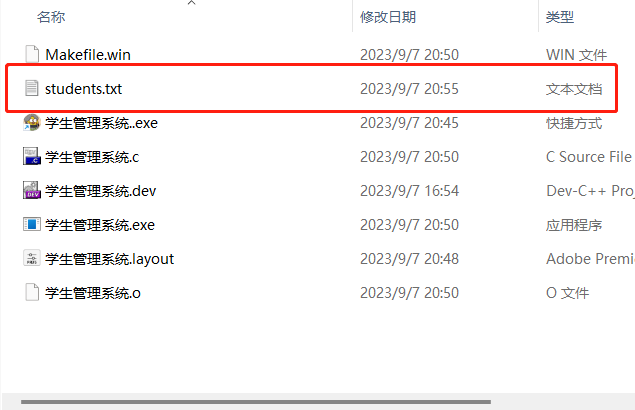
**班级平均分**



**保存数据**



在文件夹中生成一个记事本用于储存数据



**六，完整代码**

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #include <string.h>  #include <stdlib.h> //清屏头文件  #include <unistd.h> // 添加用于 usleep 的头文件  typedef struct Student  {  char num[20];  char name[20];  char sex;  int age;  int score;  } Student\_t;  typedef struct class  {  Student\_t stu\_arr[50]; // 这个班级最多有50个学生  unsigned int index; // 当前班级学生的个数  } class\_t; // 班级的类  void show();  int student\_one(class\_t \*class);  int student\_interp(class\_t \*class);  int student\_delete(class\_t \*class);  int student\_changed(class\_t \*class);  int student\_inquire(class\_t \*class);  int student\_print(class\_t \*class);  int student\_score(class\_t \*class);  int student\_sort(class\_t \*class);  int student\_sort\_num(class\_t \*class);  float student\_score\_avg(class\_t \*class);  void readStu(class\_t \*class);  int saveStu(class\_t \*class);  //主函数部分  int main(int argc, char const \*argv[])  {    int progress;  for (progress = 0; progress <= 100; progress++) {  // 清空屏幕  system("cls");  // 显示加载进度条  printf("Loading: [%-50s] %d%%\n", "================================================", progress);  // 延迟一段时间，模拟加载过程（3秒）  usleep(30000); // 30毫秒  // 清空当前行  printf("\r");  }  // 清空屏幕  system("cls");  int id = 0;  int code = 0;  int chose = 0;  class\_t \*class = (class\_t \*)malloc(sizeof(class\_t));  if (class == NULL)  {  printf("malloc memory failed!\n");  return -1;  }  class->index = 0;  while (1)  {  printf("\*\*\*\*\*\*\*\*\*|五口巴学生管理系统|\*\*\*\*\*\*\*\*\*\n");//登陆界面部分  printf("请输入登录账号->（初始账号为1001）\n");  scanf("%d", &id);  printf("请输入登录密码->（初始密码为1001）\n");  scanf("%d", &code);  if (id == 1001 && code == 1001)//这里修改账号密码  {  printf("\*\*\*\*\*\*\*\*\*\*\*\*\*|登录成功|\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  while (1)  {  printf("请选择所需要的功能(输入0进入功能菜单)：\n");  scanf("%d", &chose);  system ("cls");//清屏  switch (chose)  {  case 0:  show();  break;  case 1:  student\_interp(class);  break;  case 2:  student\_delete(class);  break;  case 3:  student\_changed(class);  break;  case 4:  student\_inquire(class);  break;  case 5:  student\_print(class);  break;  case 6:  student\_score(class);  break;  case 7:  student\_sort(class);  student\_score(class);  break;  case 8:  student\_one(class);  break;  case 9:  readStu(class);  student\_print(class);  break;  case 10:  saveStu(class);  break;  case 11:  student\_sort\_num(class);  student\_print(class);  break;  case 12:  printf("班级平均分为：%.2f\n",student\_score\_avg(class));  break;  case 13:  goto END;  default:  break;  }  }    }  else  {  printf("账号或密码错误\n");  }  }  END :  return 0;  }  //菜单  void show()  {  printf("=====================|五口巴学生信息管理系统|======================\n");  printf("|---------------------------|功能菜单|----------------------------|\n");  printf("| |\n");  printf("| 1.录入学生信息 |\n");  printf("| 2.删除学生信息 |\n");  printf("| 3.修改学生成绩 |\n");  printf("| 4.查询学生成绩 |\n");  printf("| 5.查询所有学生的信息 |\n");  printf("| 6.查询所有学生成绩 |\n");  printf("| 7.学生成绩排序 |\n");  printf("| 8.查询单个学生 |\n");  printf("| 9.导入学生信息 |\n");  printf("| 10.保存学生信息 |\n");  printf("| 11.根据学号排序 |\n");  printf("| 12.查看班级平均分 |\n");  printf("| 13.退出系统 |\n");  printf("| |\n");  printf("-------------------------------------------------------------------\n");    }  //插入一个学生 (姓名, 性别, 年龄，成绩)  int student\_interp(class\_t \*class)  {  int i;  Student\_t stu;  printf("请输入学生的 学号、姓名、性别（男M,女W）、年龄、成绩：\n");  BEGIN:  scanf("%s %s %c %d %d", stu.num, stu.name, &stu.sex, &stu.age, &stu.score);  if(stu.sex!='M'&&stu.sex!='W')  {  printf("输入的性别有误，请重新输入：\n");  goto BEGIN;  }  if(stu.score>100)  {  printf("输入成绩有误，请重新输入：\n");  goto BEGIN;  }  for (i = 0; i < class->index; i++)  {  if (strcmp(class->stu\_arr[i].num, stu.num) == 0)  {  printf("该学生已经存在，请重新输入：\n");  goto BEGIN;  }  }  class->stu\_arr[class->index] = stu;  class->index++;  printf("插入成功！\n");  return 0;  }  //删除一个学生 (根据姓名删除一个学生)  int student\_delete(class\_t \*class)  {  int i = 0;  int j = 0;  char name[20];  if (class->index == 0)  {  printf("还没有学生，请输入\n");  return -1;  }  printf("请输入要删除学生的姓名：");  scanf("%s", name);  for (i = 0; i < class->index; i++)  {  if (strcmp(name, class->stu\_arr[i].name) == 0)  {  // print\_info(&class->stu\_arr[i]);  for (j = i; j < class->index - 1; j++)  {  class->stu\_arr[j] = class->stu\_arr[j + 1];  }  class->index--;  printf("删除成功!!!\n");  break;  }  if (i == class->index)  {  printf("班级没有这个学员!\n");  return -1;  }  }  return 0;  }  //修改学生成绩 (根据姓名修改成绩)  int student\_changed(class\_t \*class)  {  char name[20];  int score = 0;  int i = 0;  if(class->index == 0){  printf("还没有学生，请输入\n");  return -1;  }  printf("请输入要修改学生的姓名：");  scanf("%s",name);  for(i = 0; i < class->index; i++){  if(strcmp(name,class->stu\_arr[i].name) == 0){  printf("请输入要修改的成绩：");  scanf("%d",&score);  class->stu\_arr[i].score = score;  printf("修改成功\n");  break;  }  }  if(i == class->index){  printf("没有该学员!\n");  }  return 0;  }  //查询学生成绩 (根据姓名查询成绩)  int student\_inquire(class\_t \*class)  {  char name[20];  int i;  if(class->index==0)  {  printf("还没有学生，请输入：\n");  return -1;  }  printf("请输入要查询的学生：");  scanf("%s",name);  for(i = 0; i < class->index; i++){  if(strcmp(name,class->stu\_arr[i].name) == 0){  printf("姓名：%-10s 成绩：%-5d\n",name,class->stu\_arr[i].score);  break;  }  }  if(i==class->index){  printf("没有该学员！");  }  return 0;  }  //查询所有学生的信息  int student\_print(class\_t \*class)  {  int i;  if(class->index == 0){  printf("还没有学生，请输入\n");  return -1;  }  for (i = 0; i < class->index; i++)  {  printf("所有学生信息：学号：%-10s 姓名：%-10s 性别：%-5c 年龄：%-5d 成绩：%-5d\n", class->stu\_arr[i].num, class->stu\_arr[i].name, class->stu\_arr[i].sex, class->stu\_arr[i].age, class->stu\_arr[i].score);  }  return 0;  }  //查询所有学生成绩  int student\_score(class\_t \*class)  {  if(class->index == 0){  printf("还没有学生，请输入\n");  return -1;  }  int i;  for (i = 0; i < class->index; i++)  {  printf("姓名：%-10s 成绩：%-5d\n", class->stu\_arr[i].name, class->stu\_arr[i].score);  }  return 0;  }  //根据成绩进行排序(从高到低， 冒泡排序)  int student\_sort(class\_t \*class)  {  int i,j;  Student\_t temp;  if(class->index==0)  {  printf("还没有学生，请输入：\n");  return -1;  }  for(i=0;i<class->index-1;i++)  {  for(j=0;j<class->index-1-i;j++)  {  if(class->stu\_arr[j].score<class->stu\_arr[j+1].score)  {  temp=class->stu\_arr[j];  class->stu\_arr[j]=class->stu\_arr[j+1];  class->stu\_arr[j+1]=temp;  }    }  }  printf("成绩由高到低排序为：\n");  return 0;  }  //根据学号查询学生信息  int student\_one(class\_t \*class)  {  char num[20];  int i;  if(class->index==0)  {  printf("还没有学生，请输入：\n");  return -1;  }  printf("请输入要查询的学生学号：\n");  scanf("%s",num);  for(i = 0; i < class->index; i++){  if(strcmp(num,class->stu\_arr[i].num) == 0){  printf("学号：%-10s 姓名：%-10s 性别：%-5c 年龄：%-5d 成绩：%-5d\n", class->stu\_arr[i].num, class->stu\_arr[i].name, class->stu\_arr[i].sex, class->stu\_arr[i].age, class->stu\_arr[i].score);  break;  }  }  if(i==class->index){  printf("没有该学生！\n");  }  return 0;  }  //读取学生信息  void readStu(class\_t \*class){  int i = 0;  FILE \* fp;  fp = fopen("students.txt","a+");  if(fp == NULL){  printf("文件打开失败!\n");  exit(0);  }    fscanf(fp,"%d",&class->index);  for(i = 0;i < class->index;i++){  fscanf(fp,"%s %s %c %d %d",class->stu\_arr[i].num,class->stu\_arr[i].name,&class->stu\_arr[i].sex,&class->stu\_arr[i].age,&class->stu\_arr[i].score);  }  fclose(fp);  printf("导入学生信息成功!\n");  }  //保存学生信息  int saveStu(class\_t \*class){  int i = 0;  if(class->index == 0){  printf("还没有学生，请输入\n");  return -1;  }  FILE \* fp;  fp = fopen("students.txt","w");  if(fp == NULL){  printf("文件打开失败!\n");  exit(0);  }    fprintf(fp,"%d\n",class->index);  for(i = 0;i < class->index;i++){  fprintf(fp,"%s %s %c %d %d\n",class->stu\_arr[i].num,class->stu\_arr[i].name,class->stu\_arr[i].sex,class->stu\_arr[i].age,class->stu\_arr[i].score);  }  fclose(fp);  printf("写入成功!\n");  }  //按学号排序  int student\_sort\_num(class\_t \*class)  {  int i,j;  Student\_t temp;  if(class->index==0)  {  printf("还没有学生，请输入：\n");  return -1;  }  for(i=0;i<class->index-1;i++)  {  for(j=0;j<class->index-1-i;j++)  {  if(strcmp(class->stu\_arr[j].num,class->stu\_arr[j+1].num)>0)  {  temp=class->stu\_arr[j];  class->stu\_arr[j]=class->stu\_arr[j+1];  class->stu\_arr[j+1]=temp;  }    }  }  printf("学号由低到高排序：\n");  return 0;  }  //统计班级平均成绩  float student\_score\_avg(class\_t \*class)  {  int i;  float avg=0,sum=0;  for(i=0;i<class->index;i++)  {  sum=sum+class->stu\_arr[i].score;  }  avg=sum/class->index;  return avg;  } |