

# 《C 语言程序设计》课程设计报告

班 级 软件 172

二级学院 信息与通信

姓 名 朱洪龙

实训地点：信息楼 505

指导教师：李新燕

      年      月      日

# C 语言程序设计课程设计报告

---

## 一、设计目的

为了能更加方便系统的管理学生三门课程的成绩，且可以进行一些必要的读取、录入、修改、查询、删除、统计等相关操作并将相关信息保存起来方便日后读取和查看，因此，我精心设计了一个学生成绩管理系统。

## 二、设计要求

- 用结构体数组设计一个学生成绩管理系统
- 每个学生的基本数据为：学号、姓名、三科成绩（英语、高等数学、计算机），三科平均分；
- 系统具有以下功能：
  - 1、**打开文件**：从指定文件读入原始数据
  - 2、**添加记录**：输入学号，如果该学号的学生不存在则把他的数据添加到结构体数组中。平均分不输入，由程序计算得出。要求输入时确保数据类型的正确。
  - 3、**删除功能**：输入学号，若该学号存在则显示该记录，用户确认要删除后才从数组中删除该记录并作相应处理。
  - 3、**修改功能**：输入学号，若该学号存在则显示并允许修改除学号以外的其余数据，显示修改后的数据，若有错可以再进行修改。
  - 4、**简单查询功能**：能按学号或姓名查询，显示查询到的记录，如果不存在则输出提示信息。
  - 5、**统计查询功能**：
    - 1) 三科成绩均大于 85 分的记录
    - 2) 有不及格课程的记录
    - 3) 平均分在指定范围的记录
    - 4) 平均分在各分数段的人数和比例（60 分以上 10 分为一个分数段，60 分以下为一个分数段）。
  - 6、**保存数据到文件**：将结构体数组中的数据保存到指定文件中。
  - 7、**主控功能**：反复显示包含各个功能选择项的菜单，等待用户选择，直至用户选择退出。
- 课程设计报告要求包含以下内容：
  - 1) 模块结构图；
  - 2) 在程序中使用到的所有的文件名、函数名、数据类型、变量类型和变量名及其说明；
  - 3) 各模块程序清单；
  - 4) 各模块功能描述；
- 5) 总体调试时的记录数据和输入数据，程序运行结果。

## 三、设计准备

1. 确定在 Windows 操作平台上进行设计，使用 C 与 C++ 程序设计学习与实验系统 2013.2 的编译器。

# C 语言程序设计课程设计报告

2. 确定信息操作的核心主体为结构体（一个结构体代表一个学生），结构体中记录学生的相关信息（学号，姓名，英语、高数和计算机三门课程的成绩，还有三科的平均分，此平均分由电脑计算自动得出），为了方便设计，在此采用了指针数组的方式存储每个结构体，所以应该在录入前先判断是否达到了数据录入的最大条数（用宏定义指针数组的最大容量）。确定菜单，也将其设为函数（操作指令错误时给出提示并可以重新输入指令）。

## 四、设计内容(步骤及程序)

### 1.数据内容及数据类型:

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <malloc.h>
#define MAXSIZE 20
#define ID 12//学号位数
#define MAXLEN 4//4表示名字的最大长度
#define KEY 'c' //密匙，初始密码为123456（六位）
const char* filename="E:\\CYuYan\\grades_table.txt";//数据文件
const char* filename2="E:\\CYuYan\\password.txt";//密码文件
//定义一个指向文件路径的常量指针来声明文件路径！
typedef struct {
    char sID[ID+1]; //学号
    char name[MAXLEN*2+1];
    //中文占两个字节，并且要预留一个空间给结束符'\0'
    float English; //英语成绩
    float Math; //高数成绩
    float Computer; //计算机成绩
    float Average; //三科平均分
}Student;
```

### 2.成绩管理

为了实现学生成绩管理的几种操作功能，首先设计一个含有多个菜单项的主控菜单程序，然后再为这些菜单项配上相应的功能。

#### 2. 1 主控菜单设计

##### 2. 1. 1 菜单内容

```
printf(" ++++-----欢迎进入学生成绩管理系统 -----++++\n");
printf("+-----          1. 读入文件          -----+\n");
printf("+-----          2. 录入学生信息        -----+\n");
printf("+-----          3. 修改学生成绩        -----+\n");
```

## C 语言程序设计课程设计报告

```
printf("+----- 4. 查询学生成绩 -----+\n");
printf("+----- 5. 删除学生成绩 -----+\n");
printf("+----- 6. 查看所有学生信息 -----+\n");
printf("+----- 7. 统计查询模块 -----+\n");
printf("+----- 8. 保存文件 -----+\n");
printf("+----- 9. 修改密码 -----+\n");
printf("+----- 0. 退出程序 -----+\n");
printf("+++++-----+\\n");
```

### 2. 1. 2 设计要求

使用数字 0—9 来选择菜单项，其他输入则不起作用。

### 2. 1. 3 设计实例

首先编写一个主控菜单驱动程序，输入 0—9 以进入相应选择项。实现循环和功能选择假设输入选择用变量 i 存储，它作为 menu 函数的返回值提供给 switch 语句。使用 while 循环实现重复选择，并在主函数 main() 中实现。

```
void main()
{
    int i=0,count;char password[8];
    Student sall[MAXSIZE];
    Student *p;
    FILE *fop=NULL;
    p=sall;

    printf("password(6 bit):");
    scanf("%6s",password); //只取前六位数字字符

    if(decrypt(password)) {
        printf("正在读入文件信息.....\n\n");
        count=readDocument(p);
        printf("文件中的原始数据如下: \n\n");
        seeAllMessage(count,p);
        i=menu();
        while(i!=0) {
            switch(i) {
            case 1: system("CLS");
                printf("\n\n");
                printf("\t\t-----<<读入文件>>-----\n\n");
                count=readDocument(p);
                if(count==0)
                    printf("\n\t\t\t\t此文件为空文件! \n\n");
                else
                    printf("\n\t\t\t\t文件读取成功! \n\n");
                system("PAUSE");
```

4

## 2. 1. 4 得到 i 的合理值

```
int menu()
{
    int i;
    printf("\n\n");
    printf("+++++-----  欢迎进入学生成绩管理系统  -----+++++\n");
    printf("  +-----  1. 读入文件  -----+\n");
    printf("  +-----  2. 录入学生信息  -----+\n");
    printf("  +-----  3. 修改学生成绩  -----+\n");
    printf("  +-----  4. 查询学生成绩  -----+\n");
    printf("  +-----  5. 删除学生成绩  -----+\n");
    printf("  +-----  6. 查看所有学生信息  -----+\n");
    printf("  +-----  7. 统计查询模块  -----+\n");
    printf("  +-----  8. 保存文件  -----+\n");
    printf("  +-----  9. 修改密码  -----+\n");
    printf("  +-----  0. 退出程序  -----+\n");
    printf("+++++-----+++++\n");
    printf("\n\t\t\t");
    do{
        printf("请输入【0-9】来执行操作：");
        scanf("%d",&i);
        fflush(stdin);
```

## C 语言程序设计课程设计报告

```
        if(i<0||i>9)
            printf("Error! The character is not able to be identified! Please input your
number again!\n\n");
    }while(i<0||i>9);
    return i;
}
```

### 2. 2 功能函数

本节的设计是以上一节设计的菜单为基础，增加菜单项的处理功能。

#### 2. 2. 1 读取文件功能

确定数据保存的路径和文件，为了简洁和方便随时调用，应该将读取文件和写入文件的操作分别定义为函数，并将 main 函数中的计数变量 count 作为返回值和参数来记录数据的条数（可以说是整个程序中最重要变量）。最后先对 FILE 类型的指针初始化为 NULL，读取文件用模式 ‘r’，用 if 条件根据此指针返回的返回值来判断，等于 NULL 则给出错误提示并自动终止程序，不等则进一步用 if 判断此文件是否为空，若如此，则给出提示和关闭数据输出流并对此函数，否则才通过文件指针来进行取出数据的操作并进行计数（此计数器应当首先初始化为 0）。

```
int readDocument(Student *p)//读取文件信息
{
    int i=0,count=0;//一定要先初始化数据条数为0以防文件为空时遇到错误！
    FILE *fp=NULL;char c;
    if((fp=fopen(filename,"r"))==NULL) {
        fprintf(stderr,"can't open the file!\n");
        exit(1);
    }
    else {
        c=fgetc(fp);
        //一定得先将文件判空写到读取文件代码的外面（得在此（while）之前进行判断）
        if(c==EOF) {
            printf("\t\t\t此文件内容为空！\n\n");
        }
        //feof()使用时应注意：如果用while在“前面”判断，
        else {
            //当第一行遇到空值时则会出错（它是通过fread或fscanf的返回值来进行判断是否遇到错误）

            rewind(fp);
            while(!feof(fp)) {

                fscanf(fp,"%s%s%f%f", &p[i].sID, &p[i].name, &p[i].English, &p[i].Math, &p[i].
Computer);

                //应特别注意要传地址值！（非指针时只传字符数组的地址）
            }
        }
    }
}
```

## C 语言程序设计课程设计报告

```
        p[i].Average=(p[i].English+p[i].Math+p[i].Computer)/3;
        i++;
    }
    count=i;
}
}
fclose(fp);
return count;
}
```

### 2. 2. 2 写入文件功能

先对 FILE 类型的指针初始化为 NULL，写入模式为 ‘w’，特性为一旦打开就会自动清空原数据，文件根据 fopen 函数返回给此指针的返回值来判断，等于 NULL 则给出错误提示并自动终止程序，不等则进一步用 if 判断所写入的数据条数是否为 0，若如此，则直接关闭数据输入流并返回，否则才进行数据的写入，不然会写入垃圾数据。

```
void writeDocument(int count, Student *p)//将结构体中的数据存入文件
{
    int i=0;FILE *fp=NULL;
    if((fp=fopen(filename, "w"))==NULL) {
        fprintf(stderr, "can't open the file!\n");
        exit(1);
    }
    else {
        if(count==0) { //写入时也要判断数据是否为空，不然会写入垃圾数据
            fclose(fp);
            return;
        }
        else
            for(i=0; i<count; i++) {

                fprintf(fp, "%s %s %.2f %.2f %.2f", p[i].sID, p[i].name, p[i].English, p[i].Math
, p[i].Computer);
                if(i!=count-1)
                    fprintf(fp, "\n");
            }
        fclose(fp);
    }
}
```

### 2. 2. 3 录入功能

为了规范学号和检查学号是否存在，应该独立设计一个检测函数，与录入相匹配(学号合法并在原数据中不存在(学号的唯一性))，后面的修改和查询依情况而定也可以调



## C 语言程序设计课程设计报告

用上述函数来进行检查。对名字也要规范限制，在顶部用宏定义 MAXLEN 来设置名字长度（最长为 4 个字），不用 gets（），因其能读入空格！。调用此函数时一次只能录入一条数据，进入前会进行确认是否要录入，录入成绩时会进行判断是否符合要求（0~100）并给出相应的提示，所有无误后则录入成功给出提示并 return 数据条数。

```
char *iInspectID(int count, Student *p) //输入并检查学号是否存在(与录入功能匹配)
{
    char num[ID+1], buffer[100]; char *c;
    int i;
    printf("请输入学生的十二位学号:");
    while(1) {
        fflush(stdin);
        gets(buffer);
        if(strlen(buffer) != 12) {
            printf("输入错误！您输入的学号为空或不为十二位\n\n");
            printf("请重新输入: ");
        }
        else {
            strcpy(num, buffer);
            for(i=0; i<count; i++) {
                if(strcmp(num, p[i].sID) == 0) {
                    printf("此学号已存在！请重新输入:");
                    break;
                }
            }
            if(i == count) {
                c = (char *) malloc((strlen(num) + 1)); //
                strcpy(c, num);
                printf("学号合法！请继续向下执行！\n\n");
                break;
            }
        }
    }
    return c;
}
```

```
int insertRecord(int count, Student *p) //录入学生信息
{
    int i=0; char *arrays, nam[50]; float e, m, c;

    int k=0;
    printf("您确定要录入学生成绩?");
    printf("\t“确定” 请输入“1”，任意字符键“取消”:");
```

## C 语言程序设计课程设计报告

---

```
scanf("%d",&k);
fflush(stdin);
if(k==1) {
arrays=iInspectID(count,p);
printf("请输入学生姓名:");
scanf("%s",&nam);
fflush(stdin);
printf("\n");
while(strlen(nam)>MAXLEN*2) {
    printf("名字长度超过最大长度! 请重新输入:");
    scanf("%s",&nam); //不用gets(), 因其能读入空格!
    fflush(stdin);
    printf("\n");
}
printf("名字合法! 请继续向下执行! \n\n");
printf("请输入英语成绩:");
scanf("%f",&e);
fflush(stdin);
while(e<0||e>100) {
    printf("错误! 科目成绩应该在0~100的范围");
    scanf("%f",&e);
    fflush(stdin);
}
printf("\n\n");
printf("请输入高数成绩:");
scanf("%f",&m);
fflush(stdin);
while(m<0||m>100) {
    printf("错误! 科目成绩应该在0~100的范围");
    scanf("%f",&m);
    fflush(stdin);
}
printf("\n\n");
printf("请输入计算机成绩:");
scanf("%f",&c);
fflush(stdin);
while(c<0||c>100) {
    printf("错误! 科目成绩应该在0~100的范围");
    scanf("%f",&c);
    fflush(stdin);
}
printf("\n\n");
strcpy(p[count].sID, arrays);
strcpy(p[count].name, nam);
```

## C 语言程序设计课程设计报告

```
p[count].English=e;
p[count].Math=m;
p[count].Computer=c;
p[count].Average=(e+m+c)/3;
printf("录入信息成功! \n\n");
system("PAUSE");
system("CLS");
}
else if(k==0) {
    system("PAUSE");
    system("CLS");
}
else {
    system("PAUSE");
    system("CLS");
}
return ++count;
}
```

### 2. 2. 4 查看所有学生信息功能

用宏定义对输出的数据进行格式的规范,使内容呈现得更加清晰明了,使代码更简洁。

```
#define HD printf("\t|      学号      |   姓名   |   英语   |   高数   |   计  
算机   |   平均分   |\n")
#define BT printf("\t|   %s   |   %s   |   %.2f   |   %.2f   |   %.2f   |   %.2f  
|\n", p[x].sID, p[x].name, p[x].English, p[x].Math, p[x].Computer, p[x].Average)
#define UD printf("\t|-----|\n")

void seeAllMessage(int count, Student *p)//查看所有信息
{
    int x;
    UD;HD;UD;
    for(x=0;x<count;x++) {
        BT;UD;
    }
    system("PAUSE");
    system("CLS");
}
```

### 2. 2. 5 删除功能

为了规范学号和检查学号是否存在,应该独立设计一个检测函数,与删除相匹配(学号合法并在原数据中存在),后面的修改和查询依情况而定也可以调用上述函数来进行

## C 语言程序设计课程设计报告

检查。为了使操作更加得人性化，设置了学号合法时进行查找，若两次找不到则自动退出至主界面。找到时则显示所找到的信息并进行确认是否要删除，执行成功后会给出相应的提示并 return 数据条数。

```
char *dInspectID(int count, Student *p) //输入并检查学号是否存在(与删除功能匹配)
{
    char num[ID+1], buffer[100]; char *c;
    printf("请输入学生的十二位学号:");
    while(1) {
        fflush(stdin);
        gets(buffer);
        if(strlen(buffer)!=12) {
            printf("输入错误! 您输入的学号为空或不为十二位\n\n");
            printf("请重新输入: ");
        }
        else {
            strcpy(num, buffer);
            c=(char*)malloc((strlen(num)+1));
            strcpy(c, num);
            printf("学号合法! 请继续向下执行! \n\n");
            break;
        }
    }
    return c;
}
```

```
int deleteRecord(int count, Student *p) //删除学生信息记录
{
    char *arrays; int i, j, k=0; int c, d=0, move;
    char head[6][7]={"学号", "姓名", "英语", "高数", "计算机", "平均分"};

    c=count;
    while(1) {
        arrays=dInspectID(count, p);
        for(i=0; i<count; i++) {
            if(strcmp(arrays, p[i].sID)==0) {
                for(j=0; j<6; j++)
                    printf("%10s", head[j]);

                printf("\n");

                printf("%-15s", p[i].sID);
                printf("%-10s", p[i].name);
                printf("%.2f", p[i].English);
                printf("%.2f", p[i].Math);
            }
        }
    }
}
```

## C 语言程序设计课程设计报告

---

```
printf("%.2f", p[i].Computer);
printf("%.2f", p[i].Average);
printf("\n\n");
printf("您确定要删除此学生的所有信息? \n");
printf("“确定” 请输入“1”， 按任意键“取消” :");
scanf("%d", &k);
fflush(stdin);
if(k==1) {
    if(i==count-1) {
        --count;
        break;
    }
    else {
        move=count-(i+1);
        for (j=0; j<move; j++, i++) {
            p[i]=p[i+1];
        }
        --count;
        break;
    }
}
else if(k==0)
    break;
else
    break;
}
}

if(c!=count) {
    printf("\n");
    printf("..... 已删除此学生的所有信息! \n\n");
    system("PAUSE");
    system("CLS");
    break;
}

else if(i==count) {
    printf("此学号不存在! 请重新输入\n\n");
    d++;
    if(d==2) { //为了使操作更加人性化, 规定了查找学号次数, 自动退出
        printf("或许您可以先查看一下所有学生信息! \n\n");
        system("PAUSE");
        system("CLS");
        break;
    }
    system("PAUSE");
}
```

```
        system("CLS");
    }
    else {
        printf("请慎重考虑后再来删除信息! \n\n");
        system("PAUSE");
        system("CLS");
        break;
    }
}
return count;
}
```

### 2. 2. 6查找功能

进入查找界面时会给出选择所需要的查找方式，按学号或按姓名或退出，按学号则找到一条就显示出来并退出且可以再次选择所要查找的方式，直到选择了退出。按姓名则会进行全局扫描，有符合的就显示出来，退出后则进入主菜单界面。

```
void searchRecord(int count, Student *p)//查询学生信息记录
{
    int i, j, k=0, b=0; char *arrays, name[MAXLEN*2+1], nam[100];
    int sum[MAXSIZE]={0}; //记录查询到的数据的序号
    char head[6][7]={ "学号", "姓名", "英语", "高数", "计算机", "平均分" };
    while(1) {
        printf("按学号查询请输入“1”，按姓名查询请输入“2”，按任意键退出\n\n");

        printf("请选择: ");
        scanf("%d", &k);
        fflush(stdin);
        if(k==1) {
            arrays=dInspectID(count, p);
            for(i=0; i<count; i++) {
                if(strcmp(arrays, p[i].sID)==0) {

                    for(j=0; j<6; j++)
                        printf("%10s", head[j]);

                    printf("\n");

                    printf("%-15s", p[i].sID);
                    printf("%-10s", p[i].name);
                    printf("%.2f", p[i].English);
                    printf("%.2f", p[i].Math);
                    printf("%.2f", p[i].Computer);
                    printf("%.2f", p[i].Average);
                }
            }
        }
    }
}
```

## C 语言程序设计课程设计报告

---

```
        printf("\n\n");
        break;//学号唯一，找到一条就退出循环！
    }
}
if(i==count) {
    printf("\n");
    printf("未能查询到与此学号相关的信息，请确认无误后再重新查询！\n\n");
}
}
else if(k==2) {
    printf("请输入学生姓名：");
    scanf("%s", nam);
    fflush(stdin);
    printf("\n");
    while(strlen(nam)>MAXLEN*2) {
        printf("名字长度超过最大长度！请重新输入:");
        scanf("%s", nam);//不用gets()，因其能读入空格！
        fflush(stdin);
        printf("\n");
    }

    strcpy(name, nam);
    for(i=0;i<count;i++) {
        if(strcmp(name, p[i].name)==0) {
            sum[b]=i;
            b++;
            //姓名不唯一，所以不能终止（每次都对所有数据进行遍历）
        }
    }

    if(b==0) {
        printf("\n");
        printf("未能查询到与此姓名相关的信息，请确认无误后再重新查询！\n\n");
    }
    else {
        for(i=b-1;i>=0;i--) {
            printf("\n\n在第 %d 条数据中存在此姓名！\n", sum[i]+1);
            for(j=0;j<6;j++)
                printf("%10s", head[j]);
            printf("\n");
            printf("%-15s", p[sum[i]].sID);
            printf("%-10s", p[sum[i]].name);
            printf("%.2f", p[sum[i]].English);
        }
    }
}
```

## C 语言程序设计课程设计报告

```
        printf("%.2f", p[sum[i]].Math);
        printf("%.2f", p[sum[i]].Computer);
        printf("%.2f", p[sum[i]].Average);
        printf("\n\n");
    }
}
}
else if(k==0) {
    system("PAUSE");
    system("CLS");
    break;
}
else {
    system("PAUSE");
    system("CLS");
    break;
}
}
```

### 2. 2. 7 修改学生信息功能

为了使操作更加得人性化，设置了学号合法时进行查找，若两次找不到则自动退出至主界面。找到时则显示所找到的信息并进行确认是否要修改，执行成功后会给出相应的提示，确认后会有个小菜单进行选择所要修改的类型（姓名，三门成绩或选择退出），小菜单可无限循环直到选择了退出。

```
void modifyRecord(int count, Student *p)//修改学生信息记录
{
    int k, d=0, y; char *arrays, nam[100]; float e, m, c;
    char head[6][7]={"学号", "姓名", "英语", "高数", "计算机", "平均分"};

    while(1) {
        int i, j;
        arrays=dInspectID(count, p);
        for(i=0; i<count; i++) {
            if(strcmp(arrays, p[i].sID)==0) {
                for(j=0; j<6; j++)
                    printf("%10s", head[j]);
                printf("\n");

                printf("%-15s", p[i].sID);
                printf("%-10s", p[i].name);
                printf("%.2f", p[i].English);
```



## C 语言程序设计课程设计报告

```
printf("%.2f", p[i].Math);
printf("%.2f", p[i].Computer);
printf("%.2f", p[i].Average);
printf("\n\n");
printf("您确定要修改此学生的信息? \n");
printf("“确定” 请输入 “1”， 按任意字符键 “取消” :");
scanf("%d", &k);
fflush(stdin);
while(k==1) {
    int b=0; //给个初始值，使得按任意键时，即使是不能存入的字符，
    也可以退出修改！
```

```
printf("+++++-----++++\n");
printf("+-+ 1. 修改姓名 +-+\n");
printf("+-+ 2. 修改英语成绩 +-+\n");
printf("+-+ 3. 修改高数成绩 +-+\n");
printf("+-+ 4. 修改计算机成绩 +-+\n");
printf("+-+ 按任意字符键退出 +-+\n");
printf("+++++-----++++\n");
printf("请选择您的操作: ");
scanf("%d", &b);
fflush(stdin);
if(b==1) {
    printf("请输入新学生姓名: ");
    scanf("%s", &nam);
    fflush(stdin);
    printf("\n");
    while(strlen(nam)>MAXLEN*2) {
        printf("名字长度超过最大长度！请重新输入:");
        scanf("%s", &nam); //不用gets()，因其能读入空格！
        fflush(stdin);
        printf("\n");
    }
    strcpy(p[i].name, nam);
    j=2;
    printf("名字修改成功! \n\n");
}
else if(b==2) {
    printf("请输入新英语成绩: ");
    scanf("%f", &e);
    fflush(stdin);
    while(e<0 || e>100) {
        printf("错误！科目成绩应该在0~100的范围");
        scanf("%f", &e);
        fflush(stdin);
    }
}
```

## C 语言程序设计课程设计报告

---

分

```
    }
    printf("\n");
    p[i].English=e;
    j=2;
    //在测试函数时发现每次修改一次成绩就得当即计算平均

    p[i].Average=(e+p[i].Math+p[i].Computer)/3;
    printf("英语成绩修改成功! \n\n");
}
else if(b==3) {
    printf("请输入高数成绩: ");
    scanf("%f",&m);
    fflush(stdin);
    while(m<0||m>100) {
        printf("错误! 科目成绩应该在0~100的范围");
        scanf("%f",&m);
        fflush(stdin);
    }
    printf("\n");
    p[i].Math=m;
    j=2;
    p[i].Average=(p[i].English+m+p[i].Computer)/3;
    printf("高数成绩修改成功! \n\n");
}
else if(b==4) {
    printf("请输入计算机成绩: ");
    scanf("%f",&c);
    fflush(stdin);
    while(c<0||c>100) {
        printf("错误! 科目成绩应该在0~100的范围");
        scanf("%f",&c);
        fflush(stdin);
    }
    printf("\n");
    p[i].Computer=c;
    j=2;
    p[i].Average=(p[i].English+p[i].Math+c)/3;
    printf("计算机成绩修改成功! \n\n");
}
else if(b==0)
    break;
else
    break;
}
```

## C 语言程序设计课程设计报告

---

```
        if(k!=1)y=1;
        break;
    }
}
if(j==2) {
    printf("\n");
    printf("..... 已修改此学生的相关信息! \n\n");
    system("PAUSE");
    system("CLS");
    break;
}
else if(y==1) {
    printf("\n");
    printf("请慎重考虑后再来修改! \n\n");
    system("PAUSE");
    system("CLS");
    break;
}
else {
    printf("此学号不存在! 请重新输入\n\n");
    d++;
    if(d==2) { //为了使操作更加人性化, 规定了查找学号次数
        printf("或许您可以先查看一下所有学生信息! \n\n");
        system("PAUSE");
        system("CLS");
        break;
    }
    system("PAUSE");
    system("CLS");
}
}
```

### 2. 2. 8 统计功能

进入界面后有个小菜单进行选择所要统计的东西, 可无限循环直到选择退出。

```
void statistics(int count, Student *p) //统计查询功能
{
    int i, j; int undersix=0, six=0, seven=0, eight=0, nine=0, ten=0; float sum;
    char head[6][7]={"学号", "姓名", "英语", "高数", "计算机", "平均分"};
    while(1) {
        int b=0;
        //给个初始值, 使得按任意字符键时, 即使是不能存入的字符, 也可以退出修改!
```

## C 语言程序设计课程设计报告

```
printf("\n\n\n");
printf("++++-----++++\n");
printf("+-      1. 三科成绩均大于85分的记录      --+\n");
printf("+-      2. 有不及格课程的记录              --+\n");
printf("+-      3. 平均分在指定范围的记录          --+\n");
printf("+-      4. 平均分在各分数段的人数和比例    --+\n");
printf("+-      按任意字符键退出                    --+\n");
printf("++++-----++++\n");
printf("请选择您的操作：");
scanf("%d",&b);
fflush(stdin);
if(b==1) {
    int k=0;
    for(i=0;i<count;i++) {
        if(p[i].English>85&& p[i].Math>85&& p[i].Computer>85) {
            for(j=0;j<6;j++)
                printf("%10s",head[j]);
            printf("\n");

            printf("%-15s",p[i].sID);
            printf("%-10s",p[i].name);
            printf("%.2f",p[i].English);
            printf("%.2f",p[i].Math);
            printf("%.2f",p[i].Computer);
            printf("%.2f",p[i].Average);
            k=1;
            printf("\n\n");
        }
    }
    if(k==0)
        printf("数据中未找到三科成绩均大于85分的记录！\n\n");
    system("PAUSE");
    system("CLS");
}
else if(b==2) {
    int k=0;
    for(i=0;i<count;i++) {
        if(p[i].English<60 || p[i].Math<60 || p[i].Computer<60) {
            for(j=0;j<6;j++)
                printf("%10s",head[j]);
            printf("\n");

            printf("%-15s",p[i].sID);
            printf("%-10s",p[i].name);
```

## C 语言程序设计课程设计报告

---

```
        printf("%.2f", p[i].English);
        printf("%.2f", p[i].Math);
        printf("%.2f", p[i].Computer);
        printf("%.2f", p[i].Average);
        k=1;
        printf("\n\n");
    }
}

if(k==0)
    printf("数据中未找到三科成绩中存在小于60分的记录! \n\n");
system("PAUSE");
system("CLS");
}

else if(b==3) {
    float a,b;int k=0;
    printf("平均分的范围: [下限值, 上限值], ps: 包含端点值\n\n");
    printf("请输入平均分的下限值: ");
    scanf("%f",&a);
    fflush(stdin);
    printf("\n");
    printf("请输入平均分的上限值: ");
    scanf("%f",&b);
    fflush(stdin);
    for(i=0;i<count;i++) {
        if(a<=p[i].Average&& p[i].Average<=b) {
            for(j=0;j<6;j++)
                printf("%10s", head[j]);
            printf("\n");

            printf("%-15s", p[i].sID);
            printf("%-10s", p[i].name);
            printf("%.2f", p[i].English);
            printf("%.2f", p[i].Math);
            printf("%.2f", p[i].Computer);
            printf("%.2f", p[i].Average);
            k=1;
            printf("\n\n");
        }
    }
    if(k==0)
        printf("数据中未找到平均分在所给范围的记录! \n\n");
    system("PAUSE");
    system("CLS");
}
```

## C 语言程序设计课程设计报告

```
else if(b==4) {
    printf("\n");
    printf("提示: 60分以上10分为一个分数段, 60分以下为一个分数段\n\n");

    for(i=0;i<count;i++) {
        if(p[i].Average<60) {
            undersix++;
        }
        else if(60<=p[i].Average&& p[i].Average<=69) {
            six++;
        }
        else if(70<=p[i].Average&& p[i].Average<=79) {
            seven++;
        }
        else if(80<=p[i].Average&& p[i].Average<=89) {
            eight++;
        }
        else if(90<=p[i].Average&& p[i].Average<=99) {
            nine++;
        }
        else if(p[i].Average==100) {
            ten++;
        }
    }

    printf("小于60分的人数为:%d\n", undersix);
    printf("[60-69]的人数为: %d\n", six);
    printf("[70-79]的人数为: %d\n", seven);
    printf("[80-89]的人数为: %d\n", eight);
    printf("[90-99]的人数为: %d\n", nine);
    printf("等于100分的人数为: %d\n\n", ten);
    sum=(float)(undersix+six+seven+eight+nine+ten);
    printf("平均分在各个分数段的比例为: \n\n");
    printf("<60 : [60-69] : [70-79] : [80-89] : [90-99] : =100\n\n");

    printf("%.2f%% : %.2f%% : %.2f%% : %.2f%% : %.2f%% : %.2f%%\n\n",
        (undersix/sum)*100, (six/sum)*100, (seven/sum)*100, (eight/sum)*100, (nine/sum)*100, (ten/sum)*100);

    //分数或分母必须至少有一个是小数才能输出带小数的值!
    system("PAUSE");
    system("CLS");
}

else if(b==0) {
    system("PAUSE");
    system("CLS");
}
```

```
        break;
    }
    else {
        system("PAUSE");
        system("CLS");
        break;
    }
}
}
```

## 2. 2. 9 解密功能

Ps: (此系统未对此功能有要求, 可以省略) 为了不让他人随意篡改数据, 确保数据的准确性和完整性, 设计加密算法 (应用在进入管理系统之后的修改密码功能, 即: 重新对其加密) 和解密算法 (需要一个文件来保存密码, 应该用 if 在进入系统前根据此函数的返回值进行判断, 此函数分两种情况, 一种是不存在保存密码的文件 (第一次使用), 所以可以自动创建一个保存密码的文件, 并记录下初始密码并返回真值, 使管理员可以进入管理系统; 另一种是有文件但也得分两种情况, 一种是有文件里面为空, 一种是有文件里面不为空, 为空的就跟设置初始密码一样, 不为空则开始它真正的解密工作 (解密应该有次数限制, 对则返回真, 否则返回假))。

```
int decrypt(char word[8])//解密算法 (若第一次使用则进行加密), KEY为密钥
{
    FILE *fp=NULL; char w1[8], w2[8]; int i=0, j=0;
    if((fp=fopen(filename2, "r"))==NULL) {
        FIRST:printf("请设置初始密码(六位数字):");
        scanf("%6s", w1); //字符数组可不用地址符
        fflush(stdin);
        printf("\n\n");
        printf("请再次确认密码:");
        scanf("%6s", w2); //只取输入流中的前6位数
        fflush(stdin);
        if(strcmp(w1, w2)==0 && strlen(w1)==6) {
            for(i=strlen(w2)-1; i>=0; i--) { //简单加密
                w2[i]+=KEY;
            }
            if(fp!=NULL)
                fclose(fp);
            if((fp=fopen(filename2, "w+"))==NULL) {
                fprintf(stderr, "设置失败!\n\n");
                fclose(fp);
            }
        }
    }
}
```

## C 语言程序设计课程设计报告

---

```
        return 0;
    }
    else {
        fputs(w2, fp);
        fclose(fp);
        printf("设置成功! \n\n");
        return 2;
    }
}
else {
    printf("设置失败! 请再次输入\n\n");
    goto FIRST;
}
}
else {
    while(!feof(fp)) {
        char c;
        c=fgetc(fp);
        if(c==EOF)break;
        rewind(fp);
        fscanf(fp, "%s", &w1);
        j++;
    }
    if(j!=0) { //文件不为空
        int k=5; //还有5次输入密码的机会
        for(i=strlen(w1)-1; i>=0; i--) { //解密
            w1[i]-=KEY;
        }
        while(k) {
            if(strcmp(w1, word)==0&&strlen(word)==6) {
                printf("\n\n\t\t密码正确, 欢迎进入学生成绩管理系统! \n\n\n");
                fclose(fp);
                return 1;
            }
            else {
                printf("密码错误! 请重新输入(剩余%d次)\n\n", k--);
                scanf("%6s", word);
                fflush(stdin);
            }
        }
    }
}
else {
    SECOND:printf("请设置初始密码(六位数字):");
```



## C 语言程序设计课程设计报告

```
scanf("%6s", w1); // 字符数组可不用地址符
fflush(stdin);
printf("\n\n");
printf("请再次确认密码:");
scanf("%6s", w2); // 只取输入流中的前6位字符
fflush(stdin);
if(strcmp(w1, w2) == 0 && strlen(w1) == 6) {
    for(i = strlen(w2) - 1; i >= 0; i--) { // 简单加密
        w2[i] += KEY;
    }
    fclose(fp);
    if((fp = fopen(filename2, "w+")) == NULL) {
        fprintf(stdin, "设置失败! ");
        fclose(fp);
        return 0;
    }
    else {
        fputs(w2, fp);
        printf("设置成功! \n\n");
        fclose(fp);
        return 2;
    }
}
else {
    printf("设置失败! 请再次输入\n\n");
    goto SECOND;
}
}
}
printf("输入次数已用尽! \n\n");
fclose(fp);
return 0;
}
```

### 2. 2. 10 加密（修改密码）功能

```
int encrypt() // 加密算法, KEY为密钥
{
    FILE *fp = NULL; char w1[8], w2[8]; int i, k = 5; // 可设置5次密码
    NEW: printf("请设置新密码(六位数字):");
        scanf("%6s", w1); // 字符数组可不用地址符
        fflush(stdin);
        printf("\n\n");
        printf("请再次确认密码:");
```

## C 语言程序设计课程设计报告

---

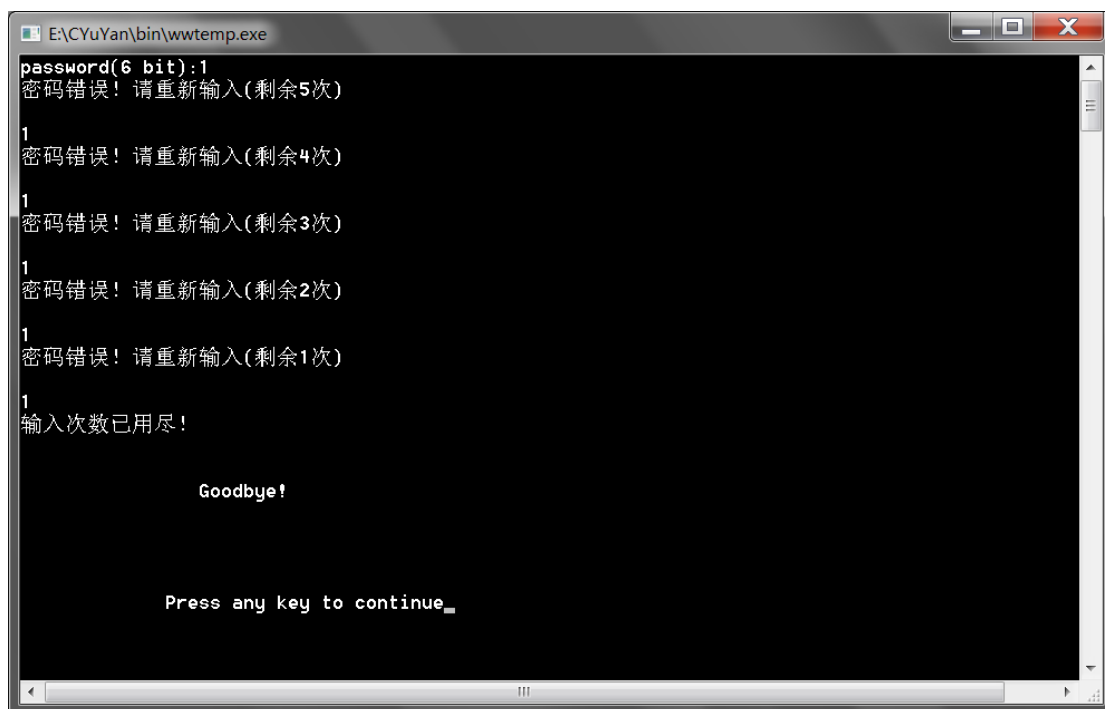
```
scanf("%6s", w2); //只取输入流中的前6位数
fflush(stdin);
k--;
if(strcmp(w1, w2) == 0 && strlen(w1) == 6) {
    for(i = strlen(w2) - 1; i >= 0; i--) { //简单加密
        w2[i] += KEY;
    }
    if((fp = fopen(filename2, "w")) == NULL) {
        fprintf(stdin, "设置失败！");
        fclose(fp);
        return 0;
    }
    else {
        fputs(w2, fp);
        printf("设置成功！\n\n");
        fclose(fp);
        return 1;
    }
}
else {
    printf("设置失败！请再次输入\n\n");
    if(k != 0) {
        goto NEW;
    }
}
return 0;
}
```

### 五、设计结果(结果分析)

#### 1. 程序初始界面测试

启动程序后会首先要求输入密码，六次输入错误后会自动终止程序。

# C 语言程序设计课程设计报告



若正确, 则首先会读取原文件数据并显示, 按任意键后则会进入主菜单界面。



## 2. 菜单功能测试

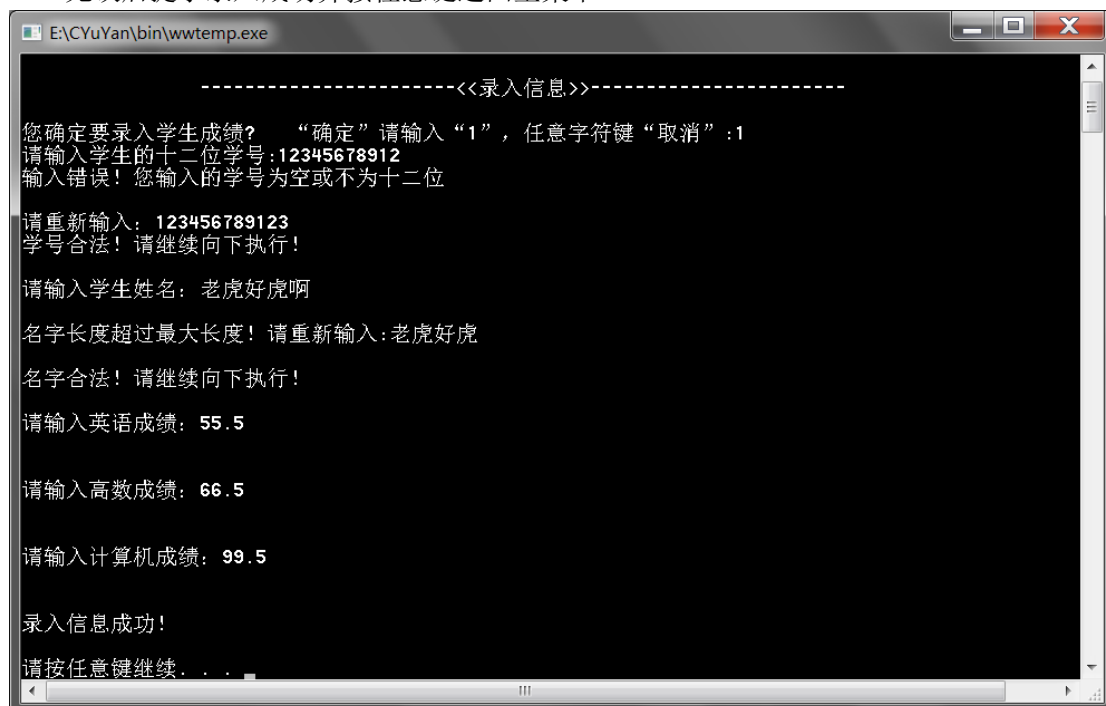
使用数字 0—9 来选择菜单项, 其他输入则不起作用。若指令输入错误则给出提示并实现重复选择, 直到指令正确或选择退出。

# C 语言程序设计课程设计报告



## 3. 录入功能测试

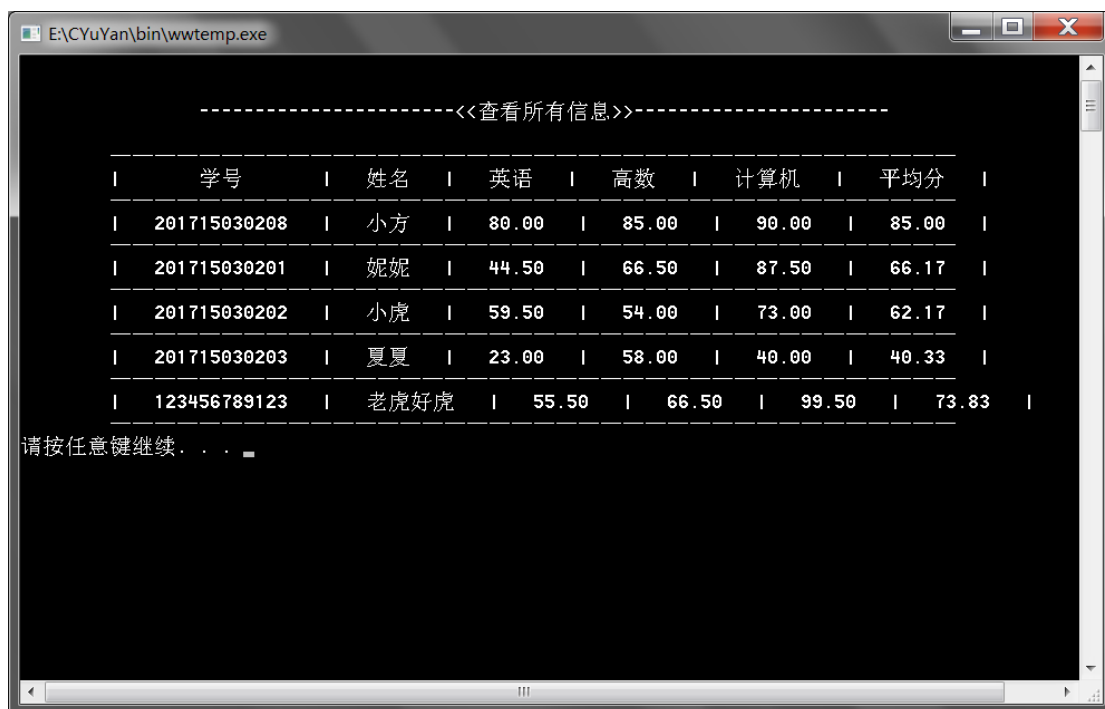
首先输入学号，不为 12 位则给出提示，并可重复输入，正确后则进行下一步输入名字，超过 4 个字给出提示并可重复输入，正确后分别输入三科成绩（限制在 0~100）无误后提示录入成功并按任意键返回主菜单。



## 4. 查看所有信息功能测试

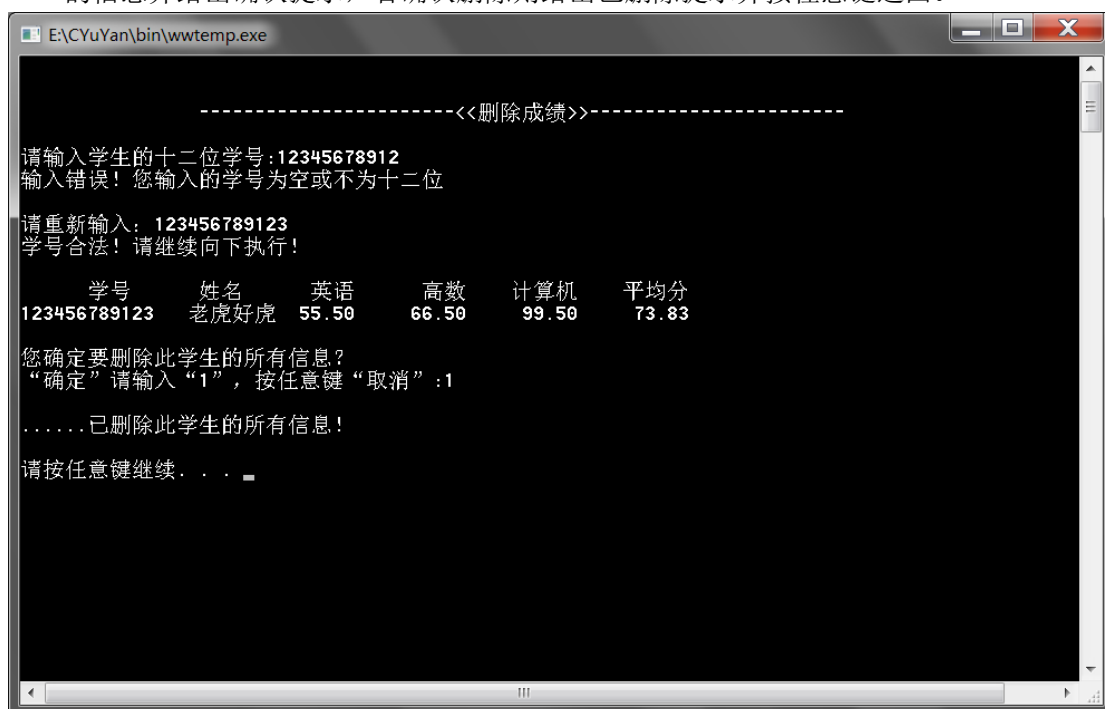
显示当前结构体中存在的所有数据

## C 语言程序设计课程设计报告



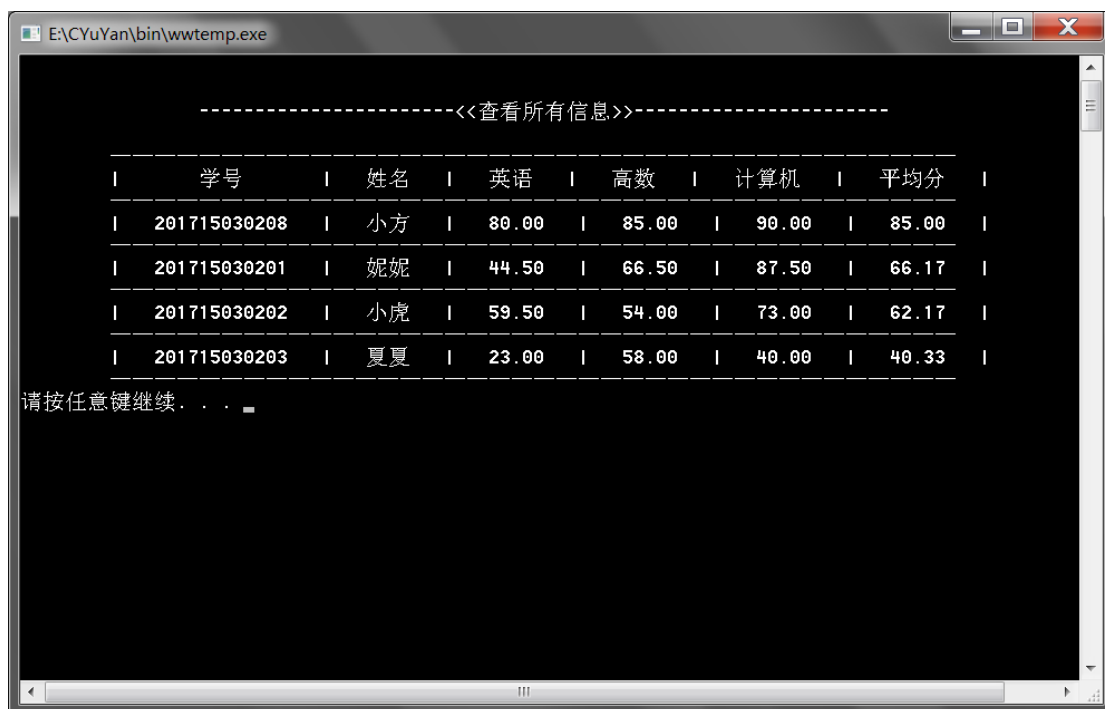
### 5. 删除功能测试

首先输入学号，不为 12 位则给出提示，并可重复输入，正确后则会显示所要删除的信息并给出确认提示，若确认删除则给出已删除提示并按任意键返回。



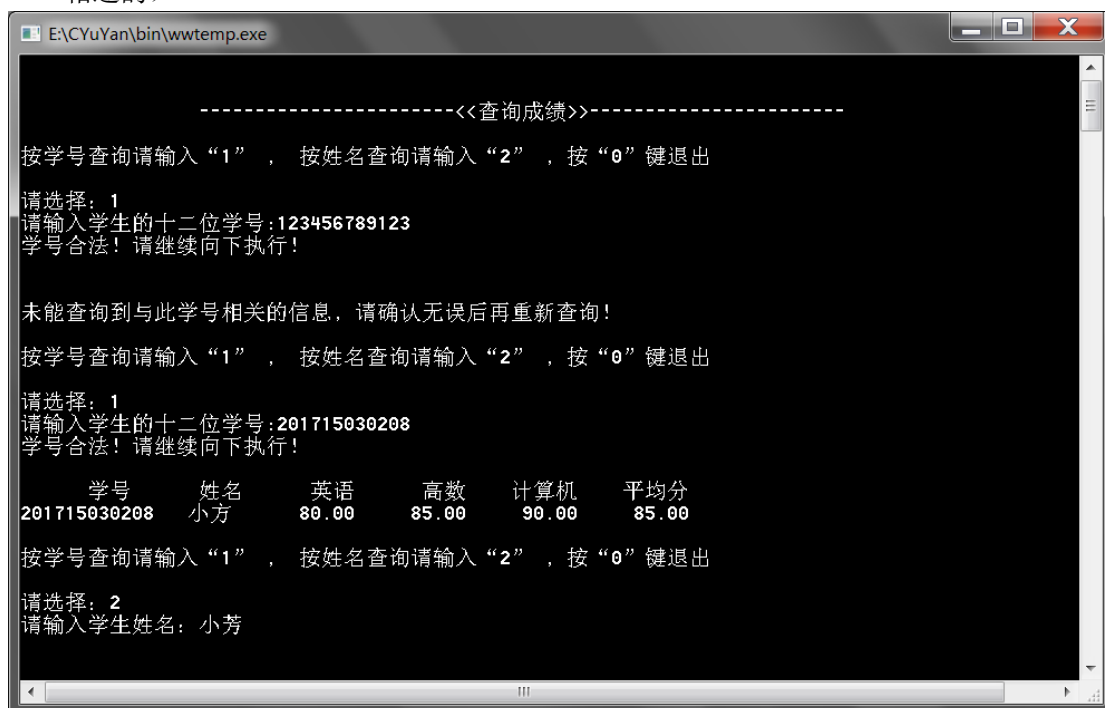
然后再去查看一下是否已删除所选数据（已删除）

## C 语言程序设计课程设计报告

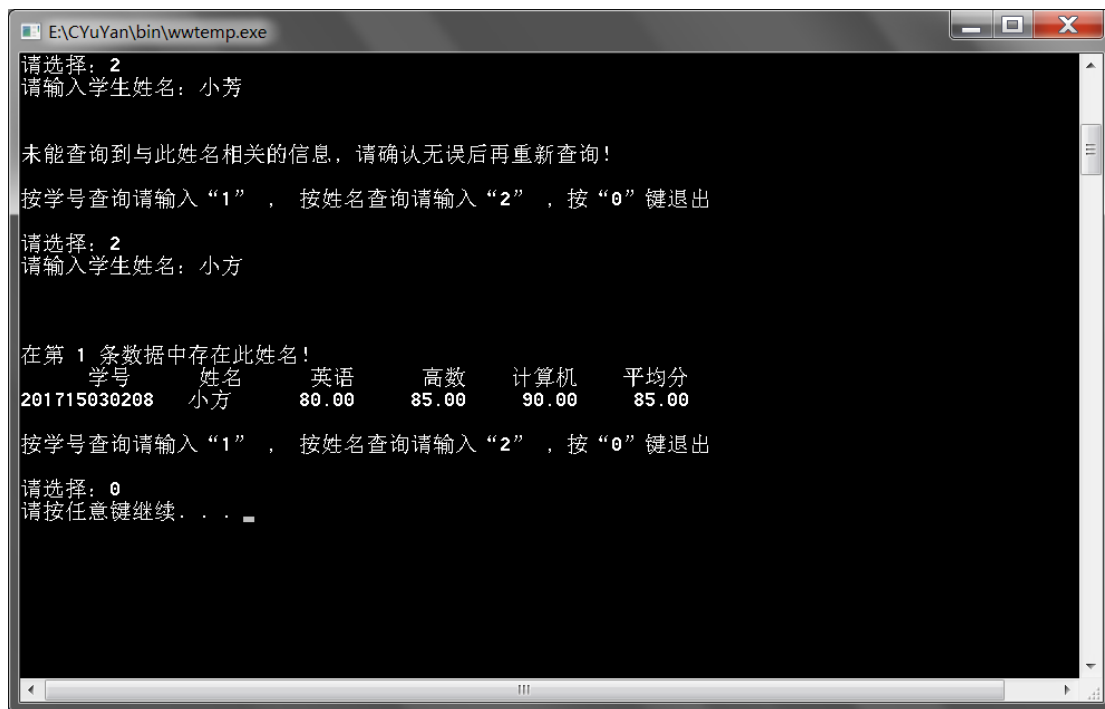


### 6. 查询功能测试

进入查找界面时会给出选择所需要的查找方式，按学号或按姓名或退出，按学号则找到一条就显示出来并退出且可以再次选择所要查找的方式，直到选择了退出。按姓名则会进行全局扫描，有符合的就显示出来，退出后则进入主菜单界面。（以下两个图为相连的）

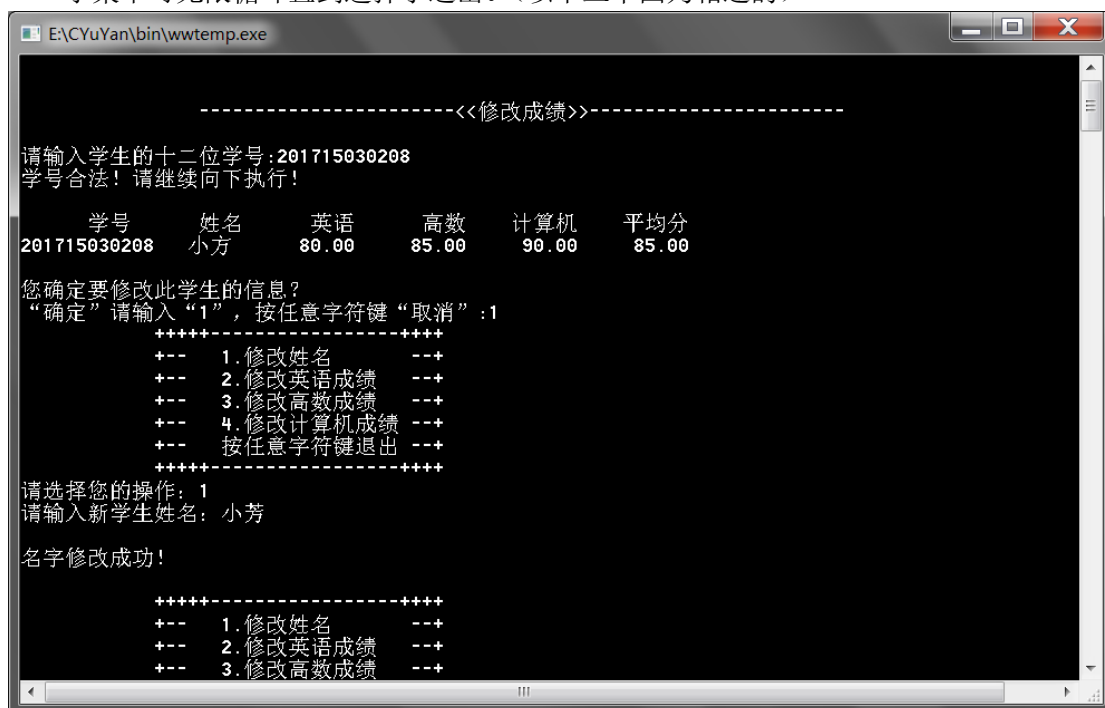


## C 语言程序设计课程设计报告

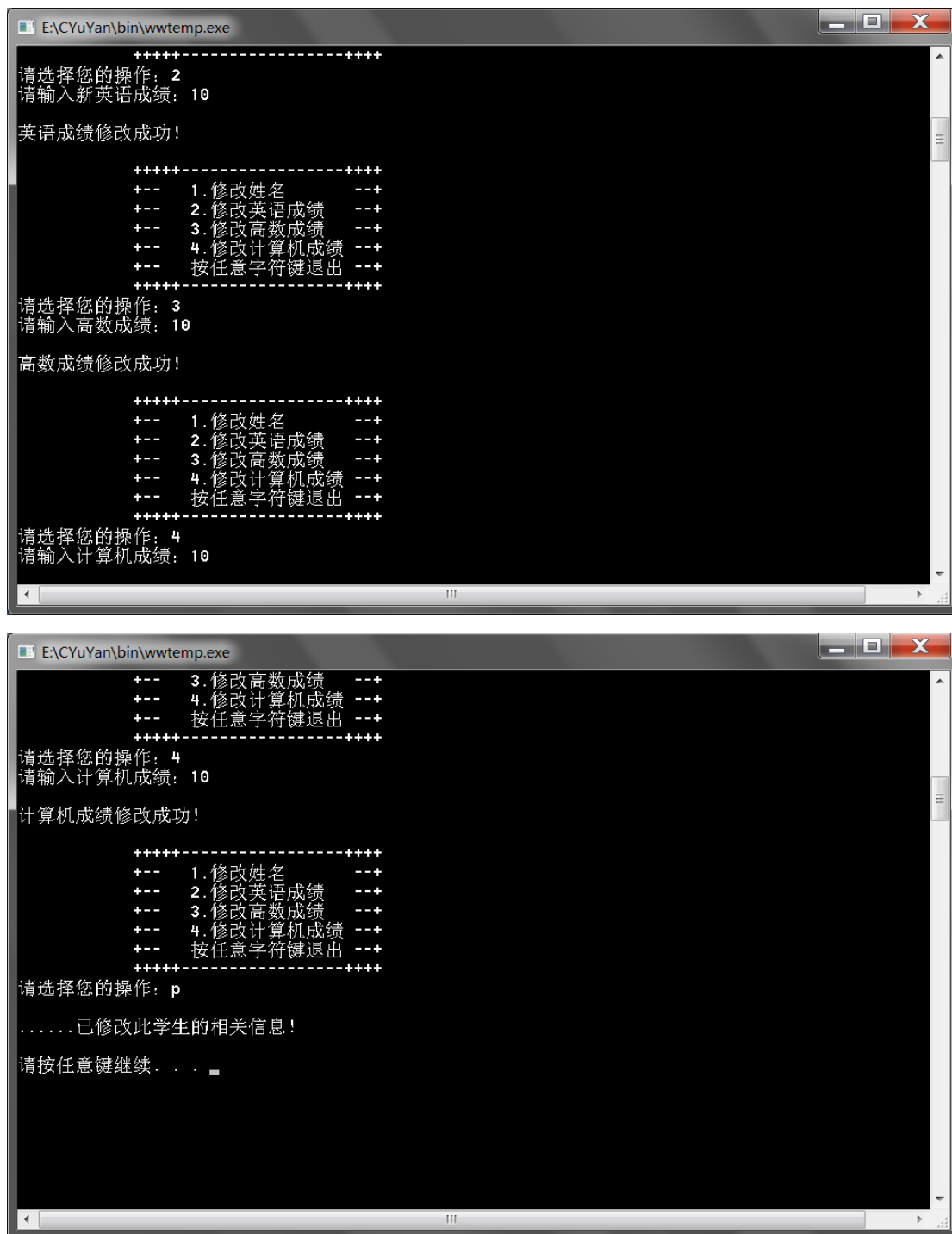


### 7. 修改功能测试

为了使操作更加得人性化, 设置了学号合法时进行查找, 若两次找不到则自动退出至主界面。找到时则显示所找到的信息并进行确认是否要修改, 执行成功后会给出相应的提示, 确认后会有个小菜单进行选择所要修改的类型(姓名, 三门成绩或选择退出), 小菜单可无限循环直到选择了退出。(以下三个图为相连的)



## C 语言程序设计课程设计报告



```
E:\CYuYan\bin\wwtemp.exe
+++++-----+++++
请选择您的操作: 2
请输入新英语成绩: 10
英语成绩修改成功!

+++++-----+++++
+-- 1.修改姓名      --+
+-- 2.修改英语成绩  --+
+-- 3.修改高数成绩  --+
+-- 4.修改计算机成绩 --+
+-- 按任意字符键退出 --+
+++++-----+++++
请选择您的操作: 3
请输入高数成绩: 10
高数成绩修改成功!

+++++-----+++++
+-- 1.修改姓名      --+
+-- 2.修改英语成绩  --+
+-- 3.修改高数成绩  --+
+-- 4.修改计算机成绩 --+
+-- 按任意字符键退出 --+
+++++-----+++++
请选择您的操作: 4
请输入计算机成绩: 10

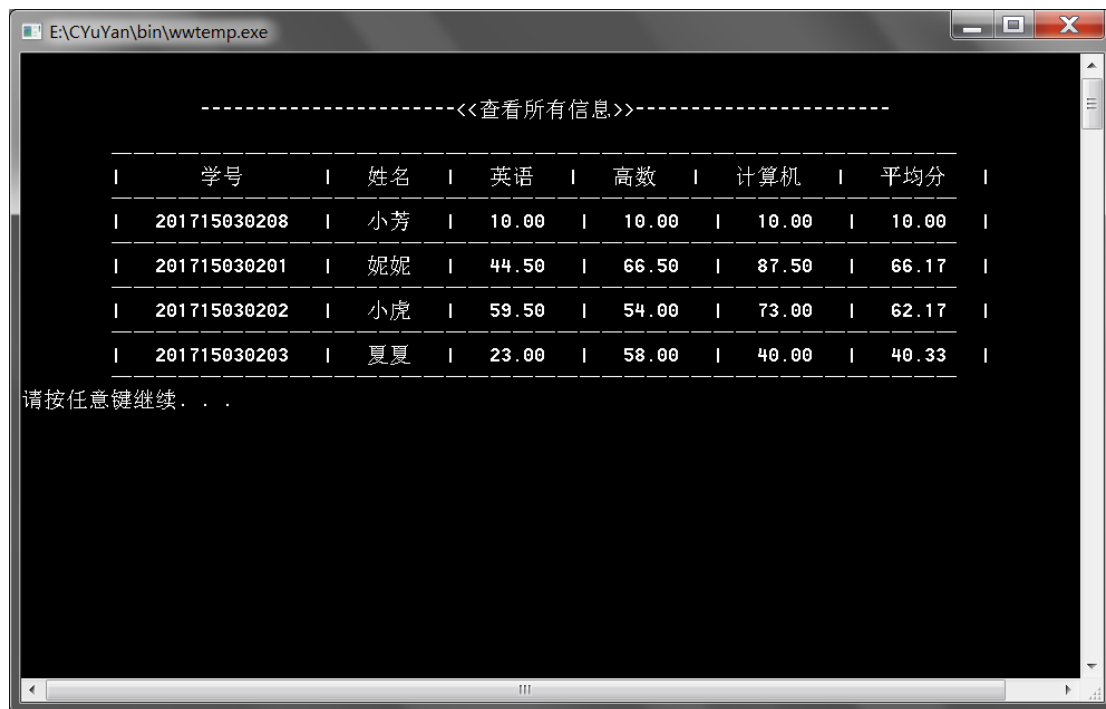
+++++-----+++++
+-- 3.修改高数成绩  --+
+-- 4.修改计算机成绩 --+
+-- 按任意字符键退出 --+
+++++-----+++++
请选择您的操作: 4
请输入计算机成绩: 10
计算机成绩修改成功!

+++++-----+++++
+-- 1.修改姓名      --+
+-- 2.修改英语成绩  --+
+-- 3.修改高数成绩  --+
+-- 4.修改计算机成绩 --+
+-- 按任意字符键退出 --+
+++++-----+++++
请选择您的操作: p
.....已修改此学生的相关信息!
请按任意键继续. . .
```

然后再去查看一下是否已修改所选数据（已修改）



## C 语言程序设计课程设计报告



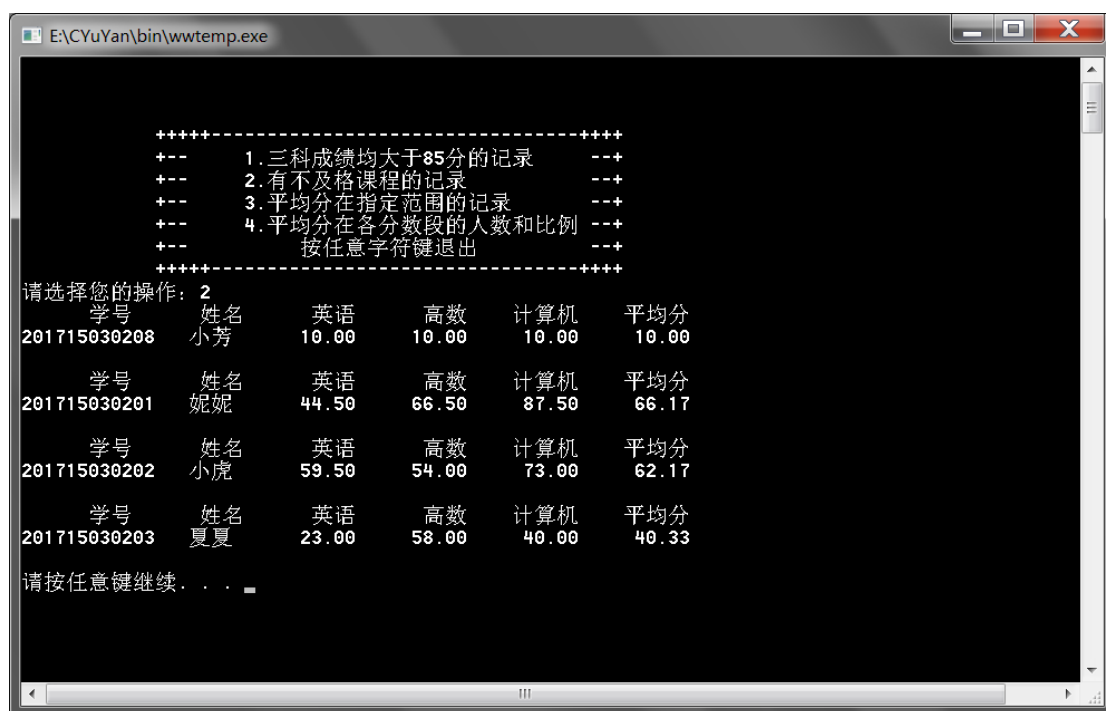
### 8. 统计功能测试

按 1 则显示三科成绩均大于 85 分的记录，没有则给出提示：

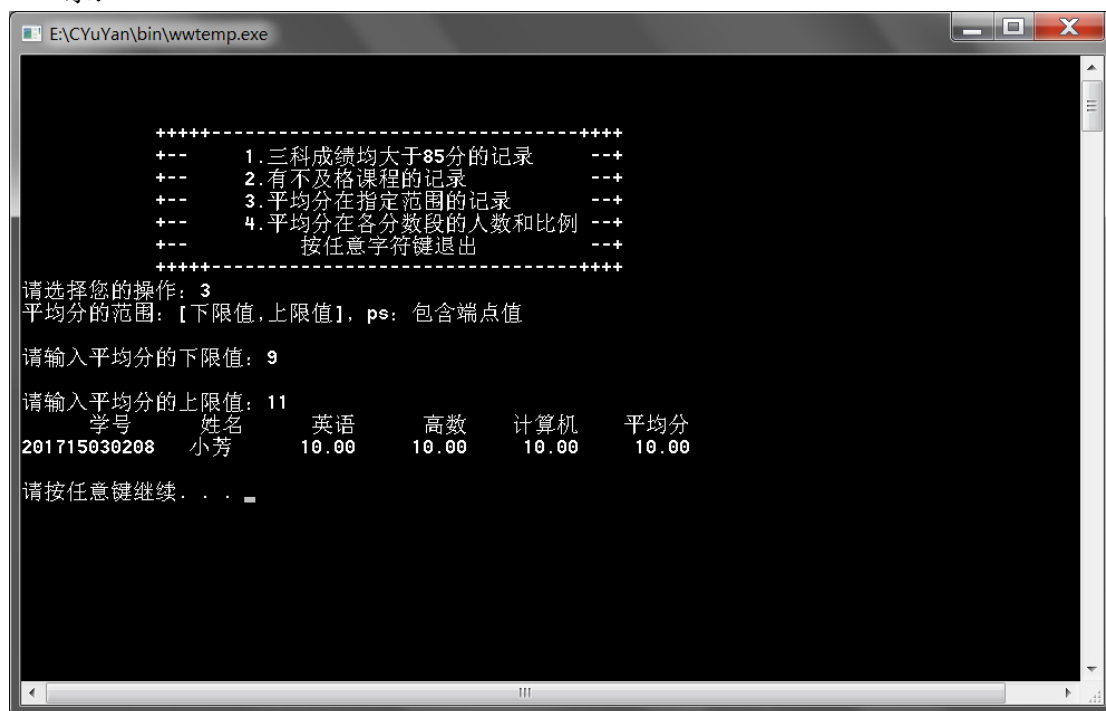


输入 2 则显示有不及格课程的记录：

## C 语言程序设计课程设计报告



按 3 则显示平均分在指定范围（分别输入上线值和下线值）的记录，没有则给出提示：



按 4 则显示平均分在各分数段的人数和比例(最后可按任意字符键退出至主菜单):

## C 语言程序设计课程设计报告



```
E:\CYuYan\bin\wwtemp.exe

+++++-----+
+--- 1. 三科成绩均大于85分的记录  ---+
+--- 2. 有不及格课程的记录        ---+
+--- 3. 平均分在指定范围的记录    ---+
+--- 4. 平均分在各分数段的人数和比例 ---+
+---      按任意字符键退出        ---+
+++++-----+

请选择您的操作: 4

提示: 60分以上10分为一个分数段, 60分以下为一个分数段

小于60分的人数为: 2
[60-69]的人数为: 2
[70-79]的人数为: 0
[80-89]的人数为: 0
[90-99]的人数为: 0
等于100分的人数为: 0

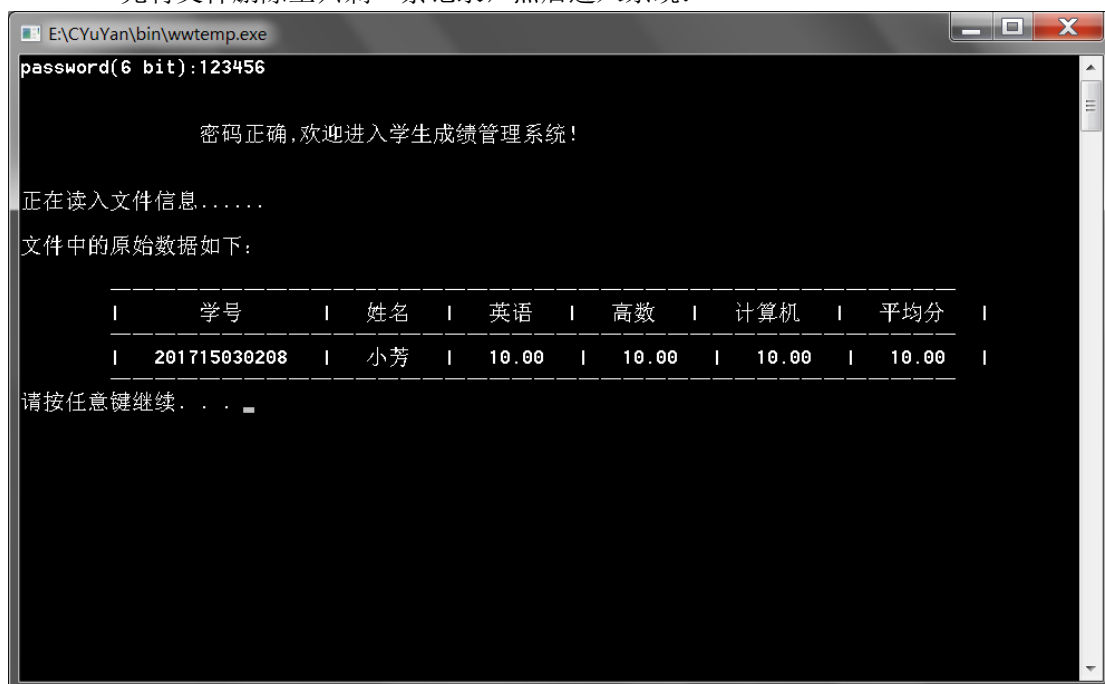
平均分在各个分数段的比例为:

<60 : [60-69] : [70-79] : [80-89] : [90-99] : =100
50.00% : 50.00% : 0.00% : 0.00% : 0.00% : 0.00%

请按任意键继续. . .
```

### 9. 文件为空的情况测试

先将文件删除至只剩一条记录，然后进入系统：



```
E:\CYuYan\bin\wwtemp.exe

password(6 bit):123456

    密码正确,欢迎进入学生成绩管理系统!

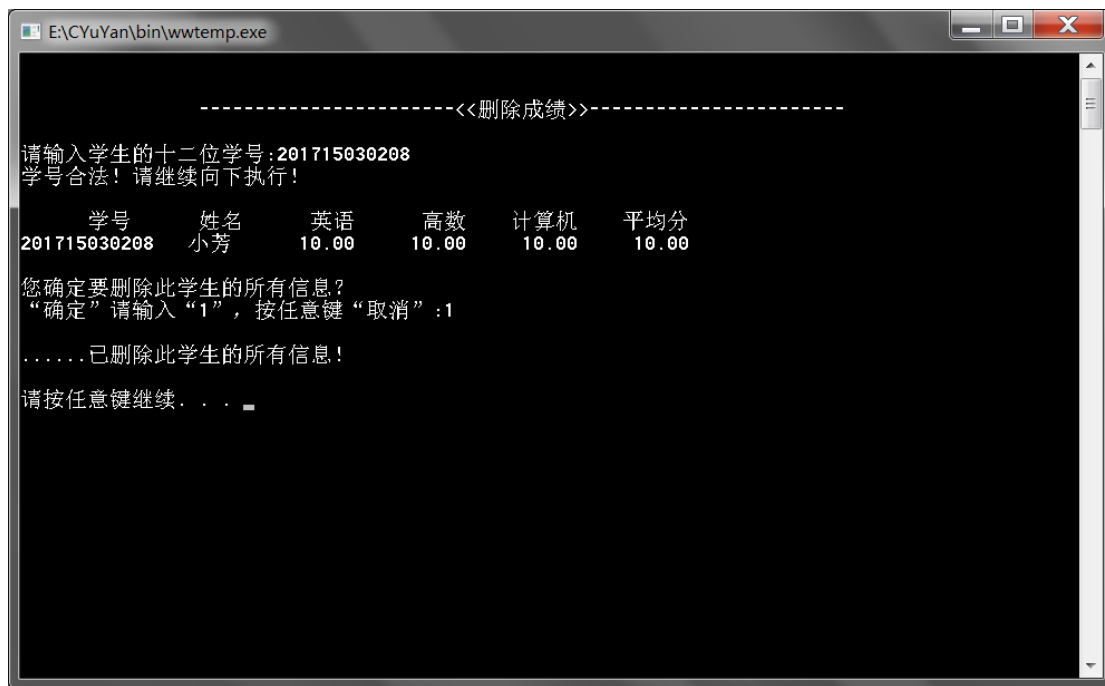
正在读入文件信息.....
文件中的原始数据如下:

+-----+-----+-----+-----+-----+-----+
| 学号   | 姓名   | 英语   | 高数   | 计算机 | 平均分 |
+-----+-----+-----+-----+-----+-----+
| 201715030208 | 小芳   | 10.00  | 10.00  | 10.00  | 10.00  |
+-----+-----+-----+-----+-----+-----+

请按任意键继续. . .
```

将最后这一条记录删除

## C 语言程序设计课程设计报告

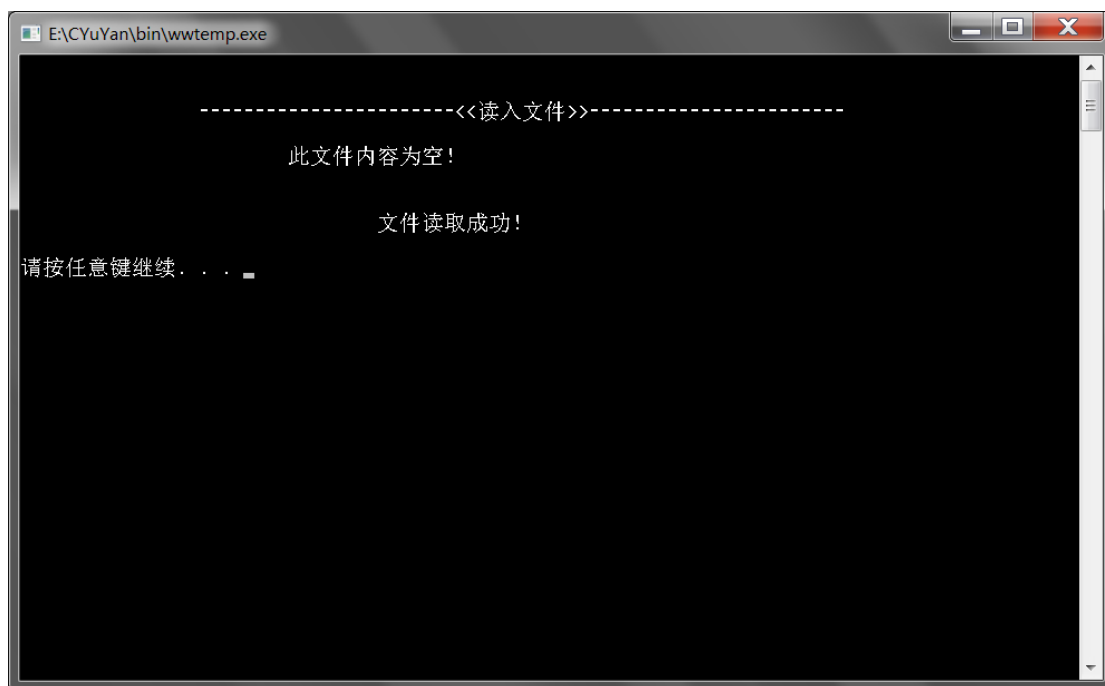


选择保存文件:

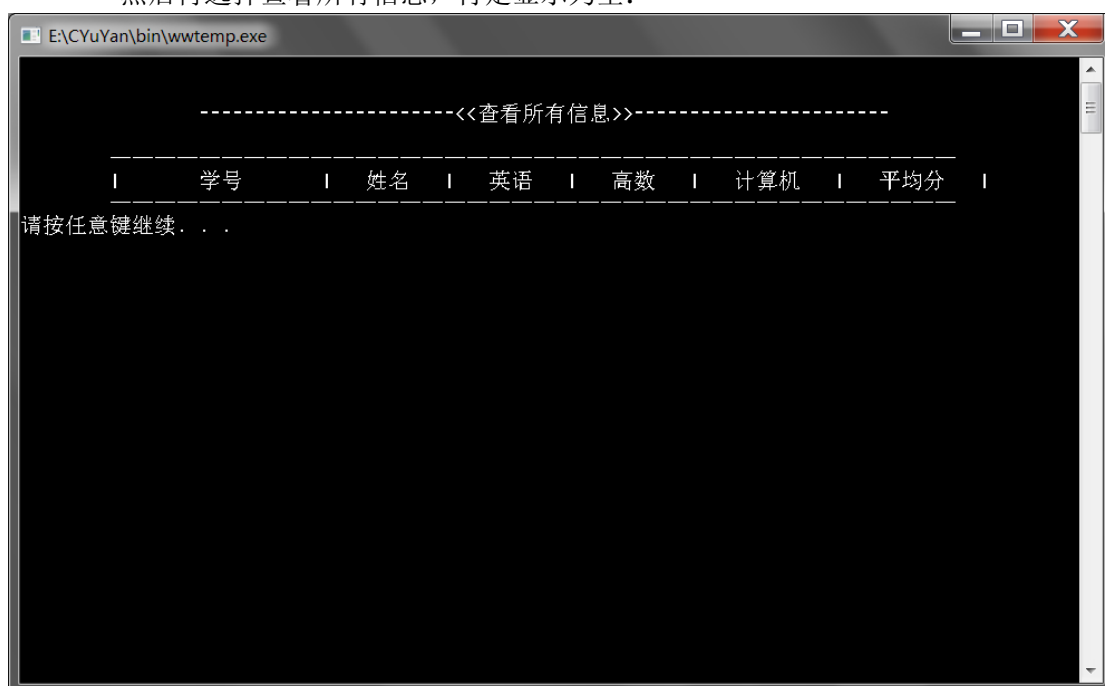


然后再在主菜单中选择读入文件（给出了文件为空的提示）:

## C 语言程序设计课程设计报告



然后再选择查看所有信息，肯定显示为空：



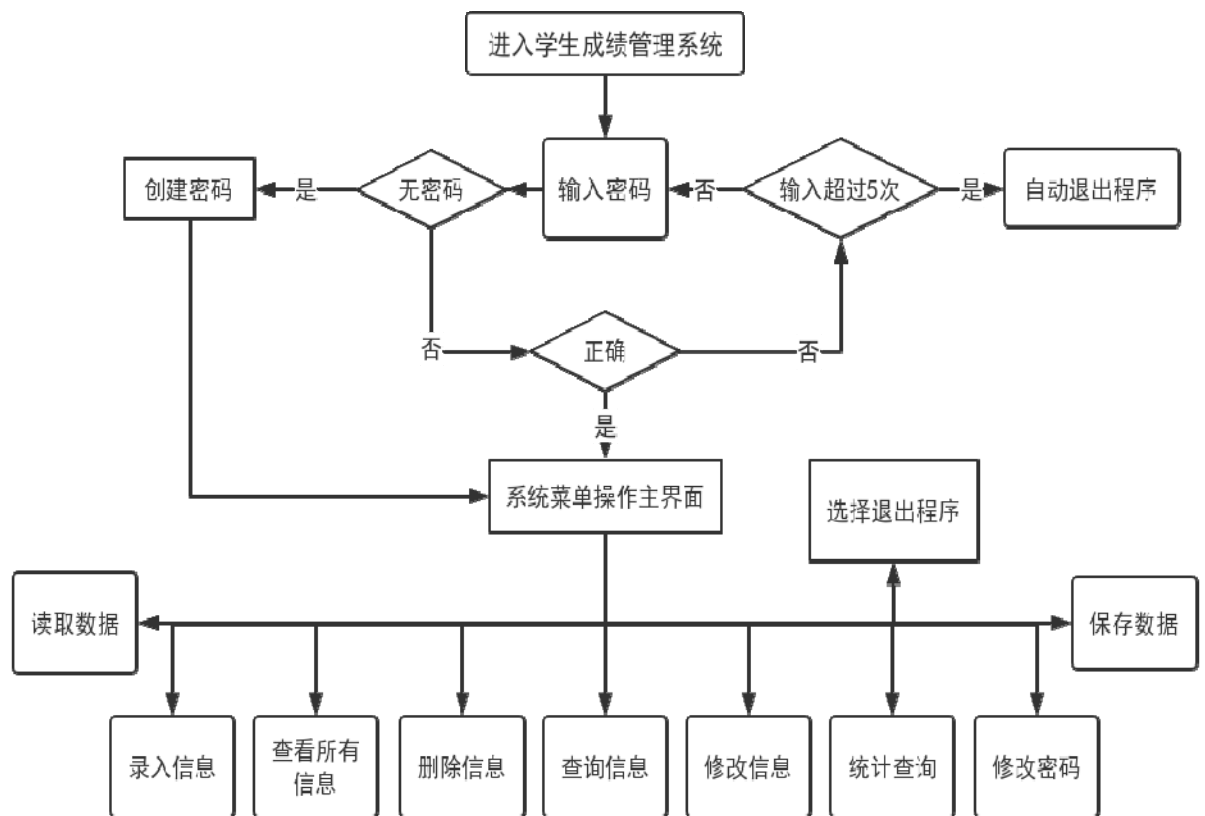
### 六、设计心得

一个星期前已经有了基本的样本，密码系统是近期加的。我没有选择用链表做，而是采用了指针数组（较简洁易懂）。写完后一直在优化（代码冗余（有些东西是计算机内部的结果，可以不必输出给用户看，此问题也是简洁问题；刚进入程序时在主函数中先读入数据，之后的对数据的相关操作都应在结构体上完成，在最后退出时，

## C 语言程序设计课程设计报告

如果想保存刚才对数据所做的操作，再在菜单中选择保存数据，不必在每次调用功能函数后都保存一次），循环条件等逻辑问题（读取文件时未考虑文件为空的情况，应该在取数据之前先进行判断），人性化操作问题（例：删除和修改时，当学号合法时，若两次找不到则自动退出至主菜单，原来的情况是只有找到了才会退出，不然则一直停留在那里），UI 问题（应该尽量使可视化的数据明了简洁的呈现给用户），难点是灵活熟练使用对文件操作的相关函数和对各种数据类型的输入输出的合理性做出准确的判断，我保存文件使用的是绝对路径（用宏定义方便修改），最后在写报告时，测试函数的修改功能出了些 BUG，就是修改了其中一门课程的成绩后没有将新的平均分保存起来，总的来说，我对相关的输入都做了必要的检测，不过在检验程序的抗压能力时我还是希望老师能够手下留情的哈！

### 七、简单流程图



## C 语言程序设计课程设计报告

---

附：

指导教师评语：

设计报告成绩：

指导教师(签字): \_\_\_\_\_

2018 年 6 月 29 日