



**学 生 实 习 报 告**

课程编号： B01044

课程名称： 程序设计与数据结构综合实践

学 号： 31501365

姓 名： 黄鹏羽

专业班级： 软件工程1503

所在学院： 计算机与计算科学学院

报告日期：2018年8月24日

|  |  |  |  |
| --- | --- | --- | --- |
| 实习时间 | 20 17 －20 18 学年 □ 秋季学期 □ 春季学期 ■暑期短学期  自 年 月 日 至 年 月 日 | | |
| 实习类型 | □ 认识（社会调查）实习  ■ 教学（课程、专题、金工）实习  □ 毕业（综合）实习 | 实习地点 | 校内 |
| 实习单位评语： | | | |
| 实习单位（盖章）：  年 月 日 | | | |
| 实习指导（带队）教师评语： | | | |
| 实习总评成绩：  教师（签名）：  年 月 日 | | | |

注：学生的实习总结等文档附在本封面之后

学生档案管理系统

## 1 内容与要求

1。创建功能：初始从文件中读取学生信息(包括学号、姓名、性别等)和学生的课程信息(课程号、课程名、成绩等)添加到顺序表中。

2。显示功能：显示全部学生的信息（包括各门课的成绩）或显示某一类学生的信息（如显示党员或男生或生源地是杭州的等等）。

3。输入某课程成绩功能：按学号、成绩输入选修某门课的所有学生的成绩

4。统计平均成绩功能：计算每个学生的平均成绩。

5。查找功能：按姓名或学号查找学生记录，并显示。

6。排序功能：按学号或平均成绩进行排序，并显示。

7。插入功能：插入一条学生记录。

8。删除功能：删除指定学生的记录。

## 2 总体设计

### 2.1 创建模块

创建学生信息时，程序会先判断fileFlag标志是否为1，若为1则表示已经读取过文件信息，则输出“您已经创建过学生信息了！请不要重复操作！”。若fileFlag标志不为1，则读取目录下的student.txt和course.txt文件，若文件读取成功，则逐行将文件中的信息添加到顺序表中，添加完成后关闭文件并输出创建成功！

### 2.2 显示模块

程序先判断顺序表的长度是否为0，若为0则表示没有学生信息，输出"暂无学生信息！"，若不为0，则遍历顺序表中的学生信息并显示，显示学生信息时，若学生不存在课程信息则输出"该学生还没有课程信息"且平均成绩输出为”暂无”。

### 2.3 输入成绩模块

首先用户先输入学号，程序根据学号去查找该学生在顺序表中的位置，若未找到则输出"没有该学生！"，若找到则判断该学生的课程链表是否为空，若为空则执行createCourse函数，初始化并插入课程信息，若不为空则执行insertCourse函数，在课程链表的表头位置插入课程信息，最后输出"添加成功！"。

### 2.4 统计成绩模块

首先输入学生的学号，程序根据学号去查找该学生在顺序表中的位置，若未找到则输出"没有该学生！"，若该学生的课程链表为空，则输出该学生还没有课程信息，否则遍历该学生的课程信息，然后计算并输出该生的平均成绩。

### 2.5 查找模块

首先进入查找模块的子菜单，按1根据学号查找，按2根据姓名查找，按0返回上一级，选择1则用户需先输入学号，程序根据学号去查找该学生在顺序表中的位置，若未找到则输出"没有该学生！"，若找到则显示该学生的所有信息，选择2则用户先输入姓名，序根据学号去查找该学生在顺序表中的位置，若未找到则输出"没有该学生！"，若找到则显示该学生的所有信息。

### 2.6 排序模块

首先程序先判断当前学生顺序表长度是否为0，若为0则输出"当前没有学生信息！"，若不为0则进入子菜单，按1根据学号排序，按2根据平均成绩排序，按0返回上一级。若选择1则根据学号对顺序表进行选择排序法，然后再遍历输出所有学生的所有信息。若选择2则根据平均成绩对顺序表进行选择排序法，然后再遍历输出所有学生的所有信息。

### 2.7 插入模块

首先用户先输入学号、姓名、性别等学生信息，然后调用insertStudent函数插入到顺序表的末尾，最后输出"插入成功！"。

### 2.8 删除模块

先输入学号，程序根据学号去查找该学生在顺序表中的位置，若未找到则输出"没有该学生！", 若找到则执行deleteStudent函数删除该学生信息，输出"删除成功！"。

## 3 详细设计

### 3.1 数据的存储结构

typedef struct CNode { // 课程信息

char cNumber[10]; // 课程号

char cName[20]; // 课程名

int point; // 成绩

int length; // 链表长度

struct CNode \*next; // 指针

}CNode, \*CourseList;

typedef struct { // 学生信息

char number[10]; // 学号

char name[10]; // 姓名

char gender[10]; // 性别

char birth[10]; // 生日

char from[10]; // 生源地

char face[10]; // 政治面貌

char mobile[15]; // 手机号

char address[30]; // 地址

CNode \*head; // 课程链表头指针

}Student;

typedef struct { // 学生信息顺序表

Student \*elem; // 学生信息

int length; // 顺序表长度

}StudentList;

### 3.2 函数实现

int initCourseList (CourseList &L) // 初始化课程表

int insertCourse(CourseList &L, int i) // 插入课程， i为插入的位置

int addCourse(CourseList &L, int i, char num[10], char name[20], int score) // 添加课程, i为插入的位置， num为课程号, name为课程名, score为成绩

void createCourse(CourseList &L) // 创建课程信息

void travelCourse(CourseList L) // 遍历课程信息

int countAver(Student stu) // 统计某个学生所有课程的平均成绩

int initStudentList(StudentList &L) // 初始化学生顺序表

int getStudent(StudentList L, int i, Student &e) // 获取某个学生的信息,

int locateStudent(StudentList L, char sNumber[10]) // 根据学号查找某个学生在顺序表中的位置

int searchStudentByName(StudentList L, char sName[10]) // 根据姓名查找某个学生在顺序表中的位置

void displayStudentInfo(StudentList L, int location) // 显示某个学生的信息

int insertStudent(StudentList &L, int i, Student e) // 插入单个学生信息

int deleteStudent(StudentList &L, int i) // 删除某个学生的信息

void sortByStudentNumber (StudentList &L) // 将线性表根据学号排序

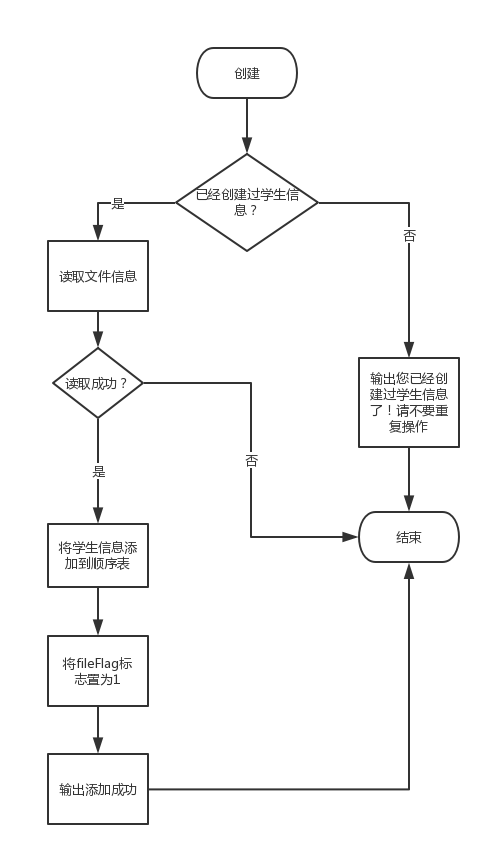
void sortByAver(StudentList &L) // 将线性表根据平均成绩排序

### 3.3 创建模块

先判断fileFlag的值是否为1，若为1则输出"您已经创建过学生信息了！请不要重复操作！"，若不为1则用fopen函数以只读的方式读取根目录下student.txt和course.txt文件。

判断文件指针fp1和 fp2是否为空，若为空则输出"文件读取失败!"，若不为空，则通过while循环读取fp1文件并将每行数据读入并使用insertStudent函数插入到表尾直至文件读取完毕。读取完fp1文件后再通过while循环读取fp2文件，先读取一行课程信息，然后通过读入的学号再根据locateStudent函数去查找该学生在顺序表中的位置并返回，然后通过addCourse函数将课程号、课程名、成绩等课程信息插入到该学生信息中的课程信息链表的头部。最后用fclose关闭 fp1和fp2文件，输出"创建成功！"并将fileFlag的值置为1，然后使用system("CLS")来清屏。

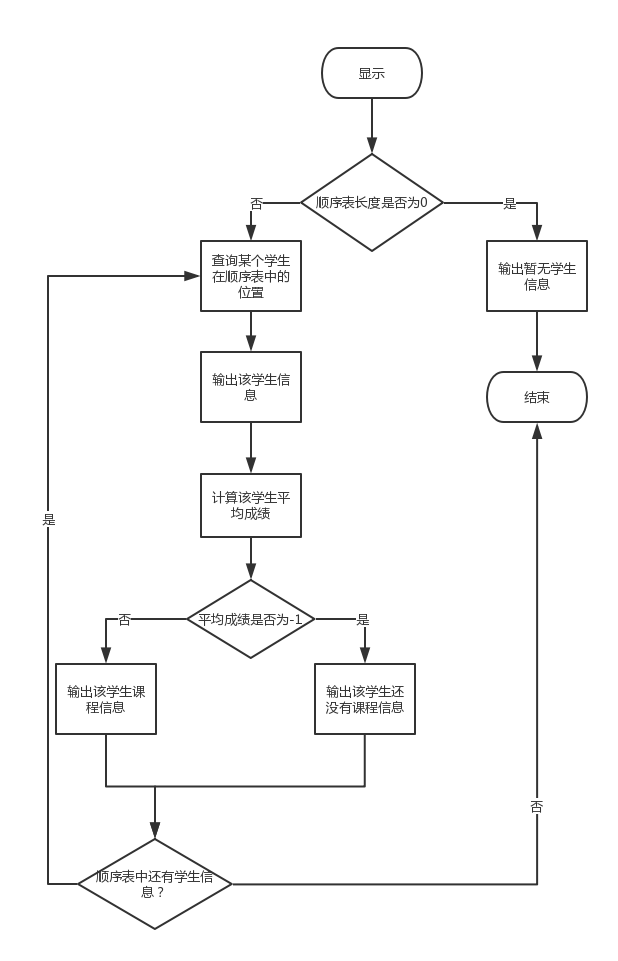
流程图如下：



### 3.4 显示模块

首先判断学生信息顺序表的长度是否为0，若为0则输出"暂无学生信息！"并清屏，若不为0，则进入for循环，循环次数为顺序表的长度值，然后调用displayStudentInfo函数获取当前学生的信息并输出，然后使用countAver函数计算该学生的平均成绩，若返回值为-1则输出"该学生还没有课程信息"，若返回值不为-1则输出该学生的相关课程信息，最后system("CLS")清屏。

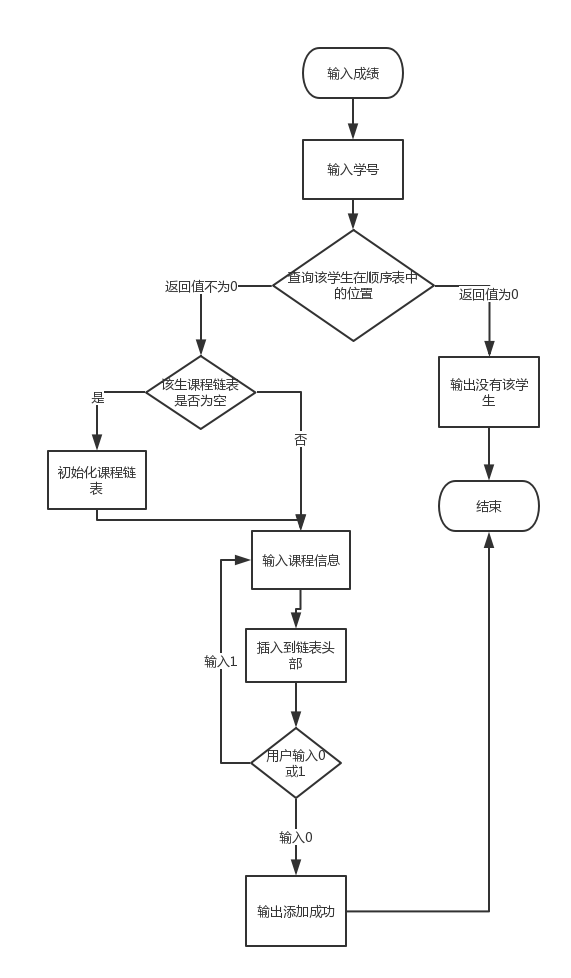
流程图如下：



### 3.5 输入成绩模块

首先先提示用户输入学生学号，然后通过学号根据locateStudent函数查找该学生在顺序表中的位置，若返回值为0则输出"没有该学生！"，若不为0则判断该学生的课程链表是否为空，若为空则调用createCourse函数先初始化课程链表然后循环输入课程号、课程名、成绩直到用户按0退出为止，若不为空则调用insertCourse函数环输入课程号、课程名、成绩插入到课程链表头部直到用户按0退出为止，最后输出"添加成功！"并system("CLS")清屏。

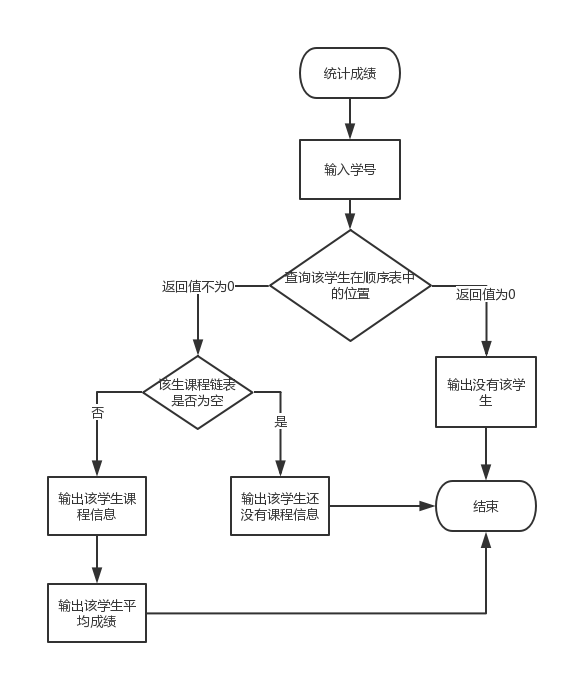
流程图如下：



### 3.6 统计成绩模块

首先先提示用户输入学生学号，然后通过学号根据locateStudent函数查找该学生在顺序表中的位置，若返回值为0则输出"没有该学生！"，若不为0则调用travelCourse函数，若链表为空则输出"该学生还没有课程信息"，若不为空则编立该学生相关的课程信息，然后再调用countAver计算该学生的平均成绩并输出，然后system("CLS")清屏。

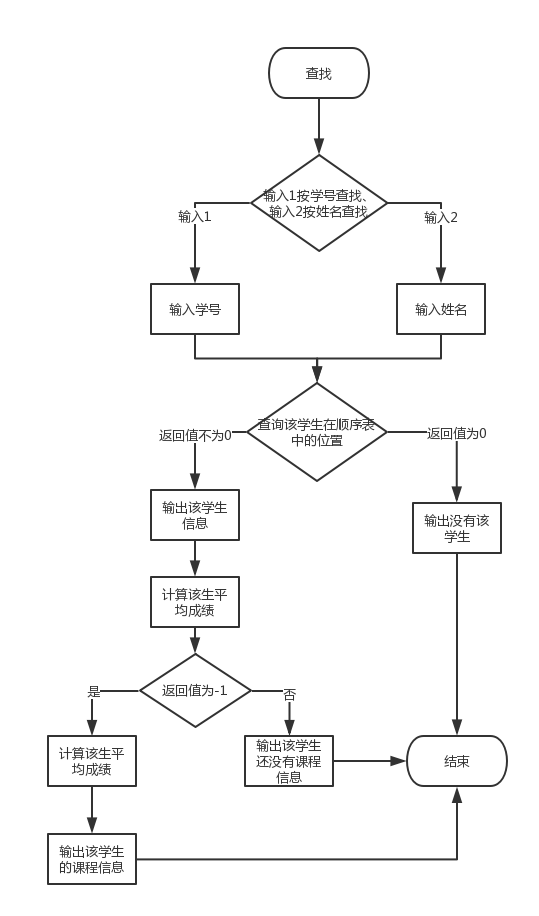
流程图如下：



### 3.7 查找模块

首先进入查询子菜单，按1根据学号查找，按2根据姓名查找，按0返回上一级，用户若选择1，再输入学号，则通过学号根据locateStudent函数查找该学生在顺序表中的位置，若返回值为0则输出"没有该学生！"，若返回值不为0则调用displayStudentInfo函数获取该学生的信息并输出，然后使用countAver函数计算该学生的平均成绩，若返回值为-1则输出"该学生还没有课程信息"，若返回值不为-1则输出该学生的相关课程信息。用户若选择2，再输入姓名，则通过姓名根据searchStudentByName函数查找该学生在顺序表中的位置，若返回值为0则输出"没有该学生！"，若返回值不为0则调用displayStudentInfo函数获取该学生的信息并输出，然后使用countAver函数计算该学生的平均成绩，若返回值为-1则输出"该学生还没有课程信息"，若返回值不为-1则输出该学生的相关课程信息。最后system("CLS")清屏，若用户选择0则退出循环返回主菜单。

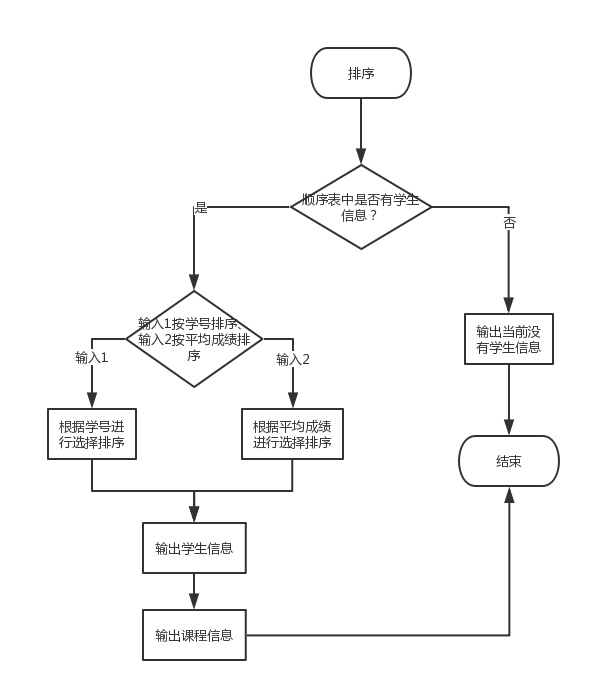
流程图如下：



### 3.8 排序模块

首先先判断当前学生信息顺序表的长度是否为0，若为0则输出"当前没有学生信息！"，若不为0则进入排序子菜单。按1根据学号排序，按2根据平均成绩排序，按0返回上一级，户若选择1，则程序调用sortByStudentNumber函数根据学号对顺序表进行选择排序法，通过嵌套for循环，让第0位的元素与后面每个元素逐个比较并交换位置，然后再让第1位的元素与后面每个元素逐个比较并交换位置，以此类推直到顺序表尾，然后通过for循环调用displayStudentInfo函数遍历输出顺序表中的所有学生信息和相对应的所有课程信息。若用户选择2, 则程序调用sortByAver函数对根据平均成绩对顺序表进行选择排序法，由于平均成绩不是student结构体中包含的变量，所以定义两个变量分别为fAver、sAver并设初值为0，用来记录每次循环时当前学生的平均成绩。然后通过嵌套for循环，让第0位的元素与后面每个元素逐个比较并交换位置，并且交换平均成绩，以此类推直到顺序表尾。然后通过for循环调用displayStudentInfo函数遍历输出顺序表中的所有学生信息和相对应的所有课程信息。最后system("CLS")清屏。

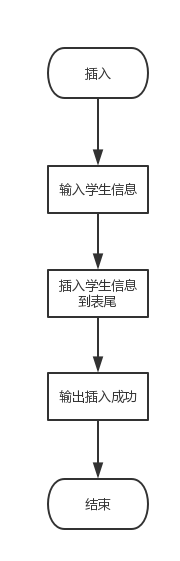
流程图如下：



### 3.9 插入模块

程序提示用户分别输入学号、姓名、性别、生日等信息，输入完成后调用insertStudent函数将当前学生信息插入到顺序表末尾并将顺序表长加1，然后输出"插入成功！"并system("CLS")清屏。

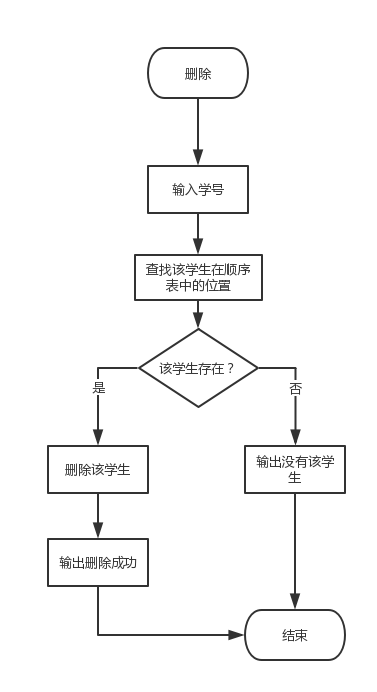
流程图如下：



### 3.10 删除模块

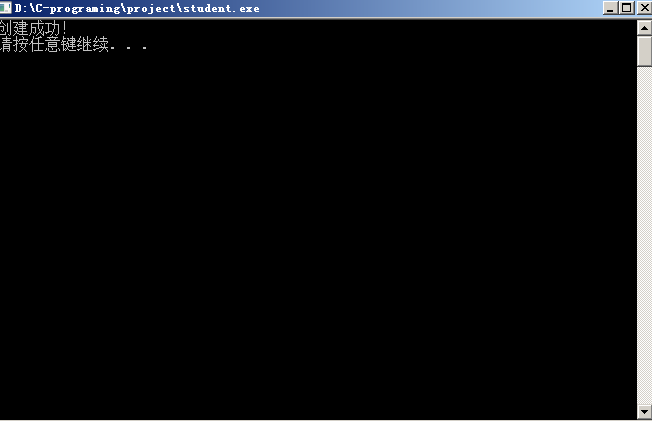
首先提示用户输入学号，用户输入学号后，通过学号根据locateStudent函数查找该学生在顺序表中的位置，若返回值为0则输出"没有该学生！"，若返回值不为0则调用deleteStudent函数删除该学生的信息并将顺序表长度减1，然后输出"删除成功！"并system("CLS")清屏。

流程图如下：



## 4 程序测试

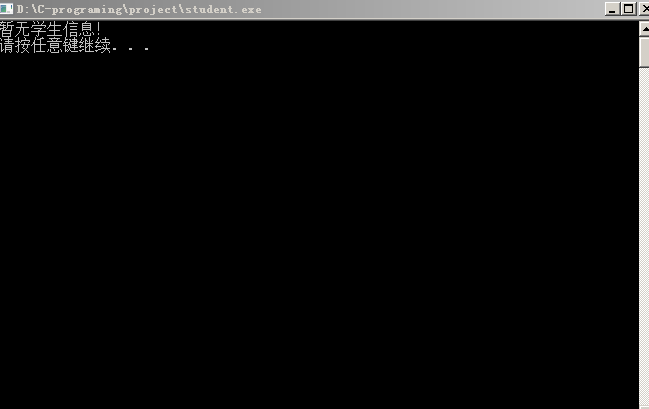
创建：



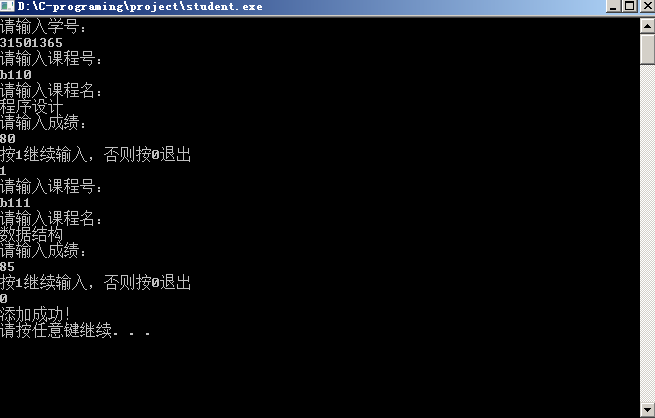
然后显示创建的信息：



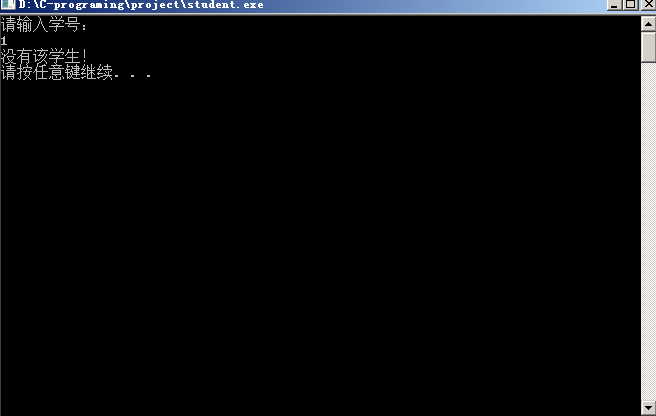
若没有学生信息：



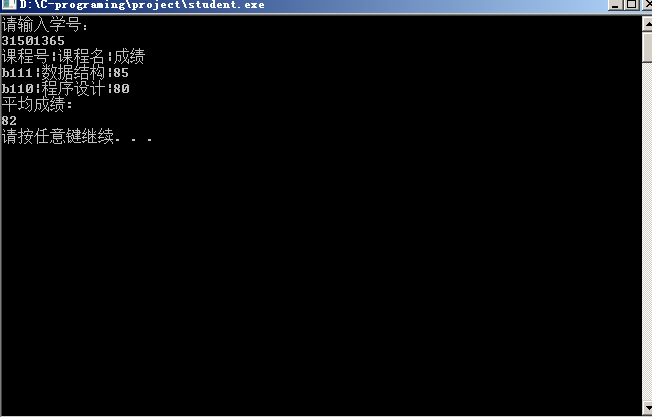
输入成绩：



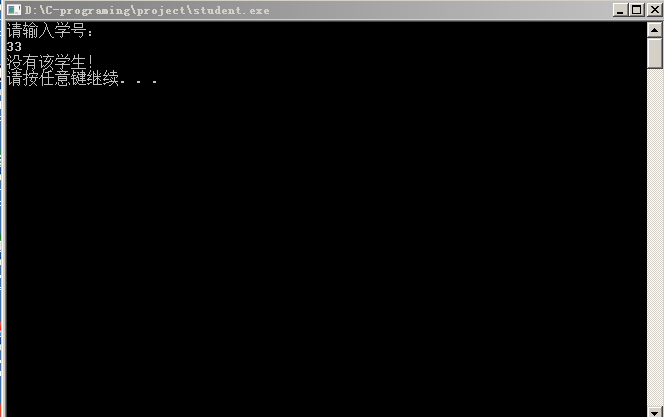
若学号输入错误：



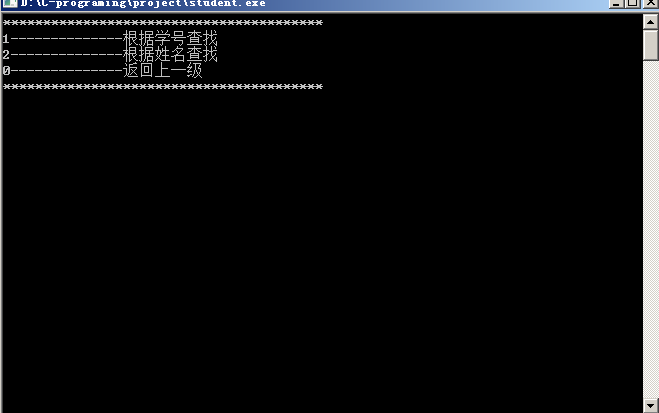
统计成绩：



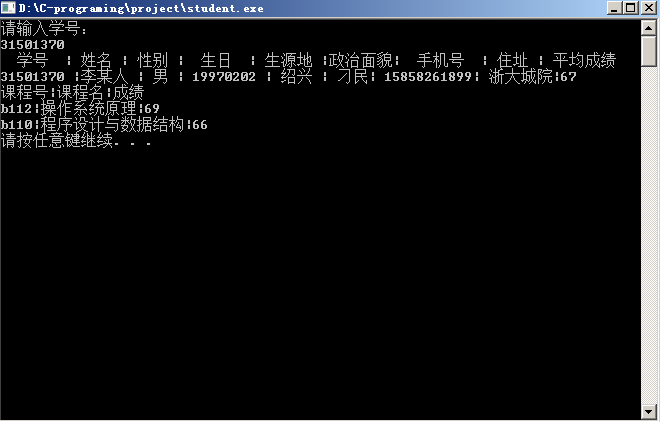
若学号输入错误：



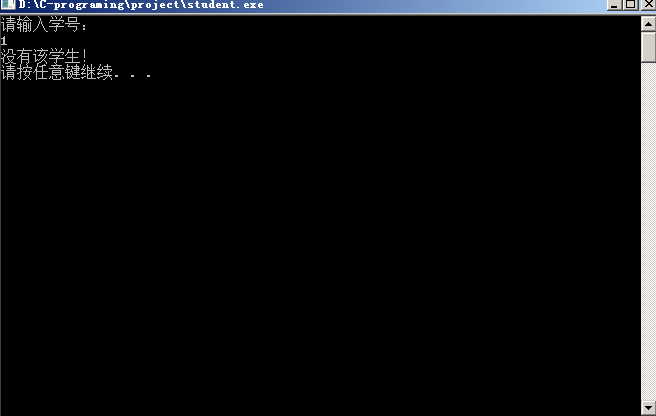
查找，先显示子菜单：



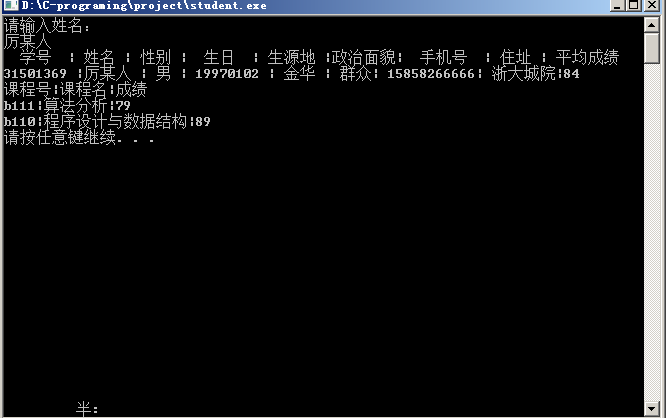
根据学号查找：



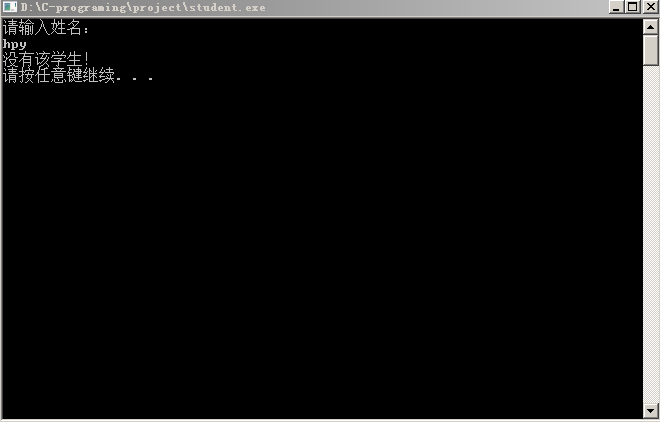
若学号输入错误：



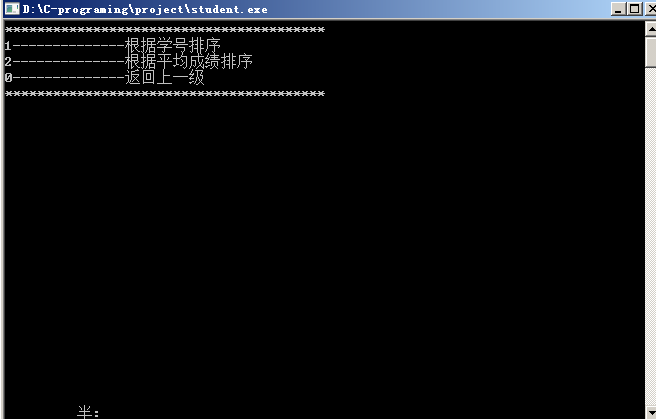
根据姓名查找：



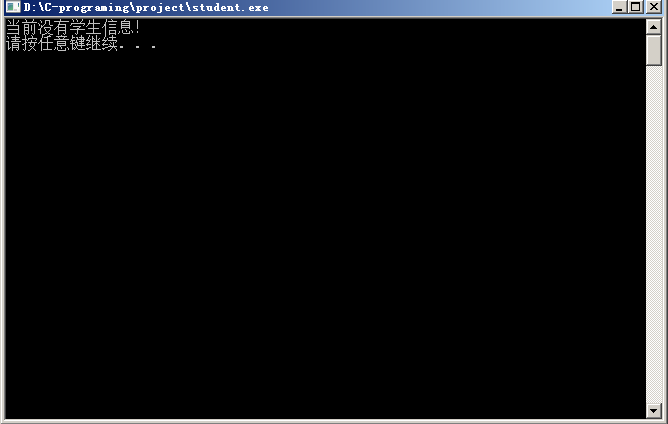
若姓名输入错误：



排序，先显示子菜单：



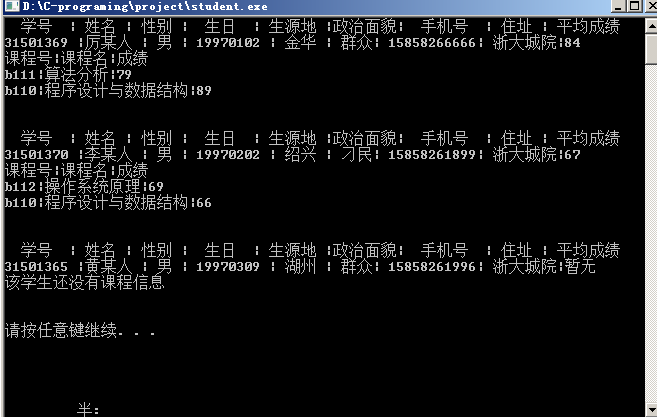
若还没有学生信息：



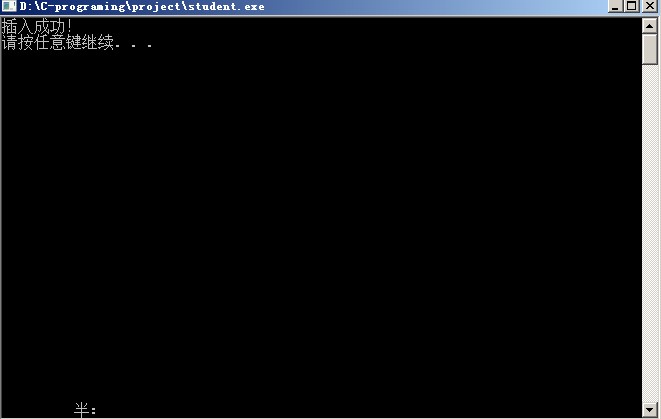
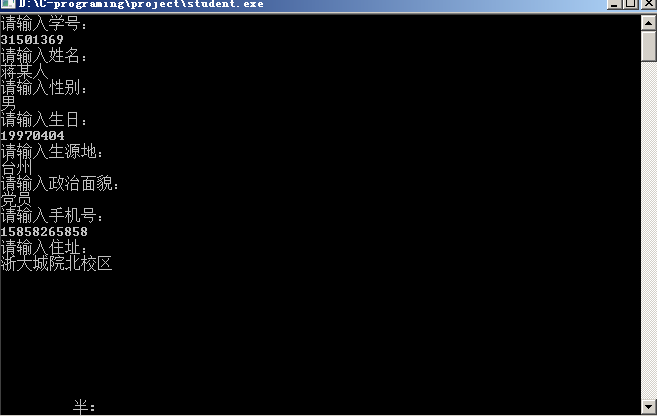
根据学号排序：



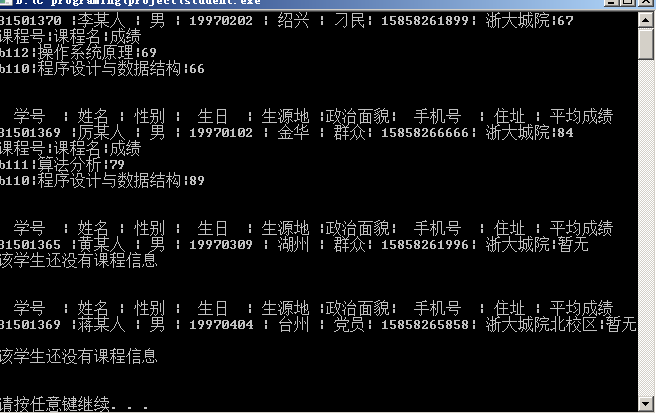
根据平均成绩排序：



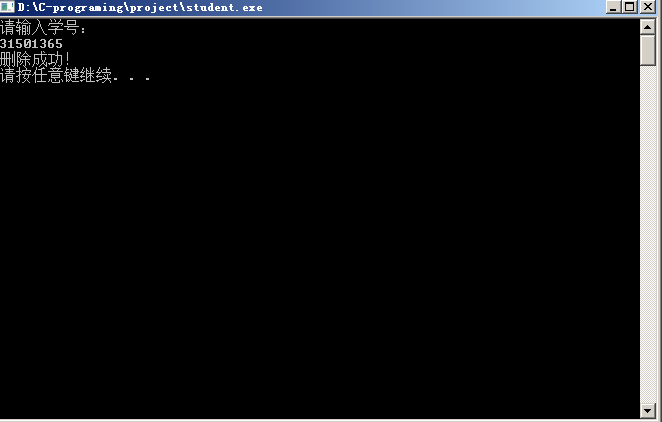
插入：



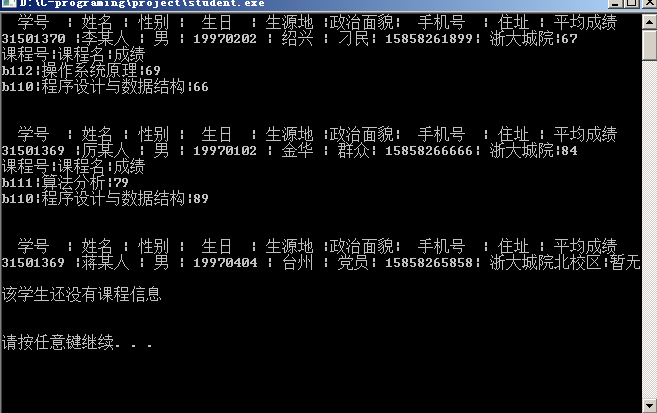
可以看到刚插入的信息：



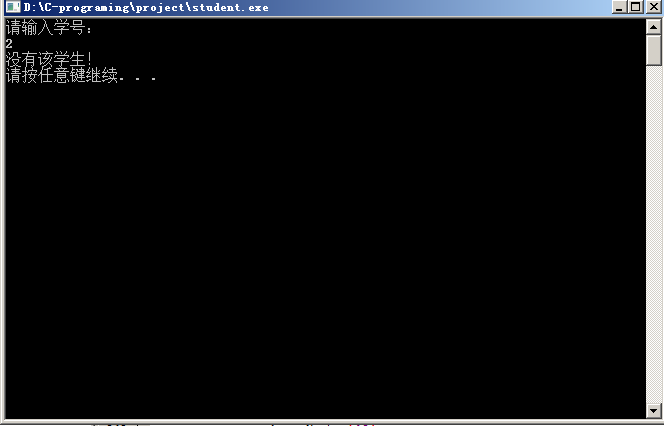
删除：



可以看到31501365这条记录被删除了:



若学号输入错误：



## 5 总结

通过一周多的短学期实践，让我巩固了算法和数据结构的基础，重新学习了最基础最底层的操作，让我对编程有了一个更加深刻的认识。通过实践，我也发现了自己很多的不足之处，平时不太会用编译器自带的调试工具，让我吃了不少苦头，这次实践，很好的训练了我调试程序的能力，对今后的日子来说，是不小的帮助。在这次实践中，我也学习到了不少C语言的标准库函数用法，也感受到了不同编程语言之间的差异，对我的编程思维是有很大的帮助，并且独立完成一个这样的题目，对我来说也有不小的成就感。希望在今后的日子里能够再接再厉。