**《高级语言程序设计》**

**课程设计报告**

**题 目 学生成绩管理系统**

**专业及班级 计算机1701**

**姓 名 叶文滔**

**学 号 1711640118**

2018年 6 月 24 日

# 系统功能描述

该学生成绩管理系统主要实现了学生成绩的管理功能，其中包括学生的输入、新增、删除、搜索、修改、排序、统计、保存和读取功能，并通过较为直观的用户界面显示这些数据，实现了一个在 Windows 和 Linux 平台上通用的精简的学生成绩管理系统。本系统以结构体×链表作为核心数据类型，可保存高达超大量人员及长达 18 位的人名、学号、三科成绩，仅需对程序进行少量修改即可增加更多的成绩列，使得该程序具有较佳的可以执行。

# 总体设计

## 功能模块设计

### 数据输入模块

该模块用于全新输入学生成绩数据。若此前已输入过数据，则会提示用户是否覆盖数据。

### 数据删除模块

该模块用于删除内存中的学生成绩数据。

### 数据搜索模块

该模块用于从内存中搜索指定学号、姓名、的学生成绩数据。

### 数据修改模块

该模块用于修改内存中的学生成绩数据。

### 数据插入模块

该模块用于在内存中新增学生成绩数据。

### 数据统计模块

该模块用于统计各科的学生成绩数据，包括总分、平均分等。

### 数据排序模块

该模块用于对内存中的学生成绩数据进行总分/平均分降序排序。

### 数据保存模块

该模块用于保存内存中的学生成绩数据到磁盘的当前目录。

### 数据读取模块

该模块用于从磁盘中读取通讯录数据到内存。此模块不单独显示，而是作为程序进入时的温馨提示自动调用。

### 数据显示模块

该模块用于显示内存中的通讯录数据。在进行数据删除、数据搜索、数据插入‘数据排序、读盘操作时，会自动调用该模块，以实时跟踪数据变化。

### 语言选择模块

该模块用于从英语、粤语（非书面）、汉语中选择一项作为程序的显示语言。此模块不单独显示，而是在程序进入时自动调用。

## 数据结构设计

结构体定义：

struct student

{

char num[MAXNUMLENGTH]; /\*学号\*/

char name[MAXNAMELENGTH]; /\*姓名\*/

int cgrade; /\*C语言成绩\*/

int mgrade; /\*数学成绩\*/

int egrade; /\*英语成绩\*/

int total; /\*总分\*/

float ave; /\*平均分\*/

int mingci; /\*名次\*/

};

typedef struct typeStudent STUDENT;

其中 MAXNUMLENGTH、MAXNAMELENGTH 默认值均为 18

链表定义

struct link

{

STUDENT data;

struct link \*next;

};

## 函数功能描述

int main() // 程序主函数，包括从进入到退出的整个流程

struct link \*InputRecord(struct link \*head) // 负责数据输入模块的功能

struct link \*DeleteRecord(struct link \*head) // 负责数据删除模块的功能

void SearchRecord(struct link \*head) // 负责数据搜索模块的功能

void ModifyRecord(struct link \*head) // 负责数据修改模块的功能

void InsertRecord(struct link \*head) // 负责数据插入模块的功能

void CountRecord(struct link \*head) // 负责数据统计模块的功能

void Swap(struct link \*stuA, struct link \*stuB) // 实现两个结构体的数据交换

struct link \*ReadByIndex(struct link \*head, int index) // 实现按顺序读取链表第 index 项

void SortRecord(struct link \*head) // 负责数据排序模块的功能

void SaveRecord(struct link \*head) // 负责数据保存模块的功能

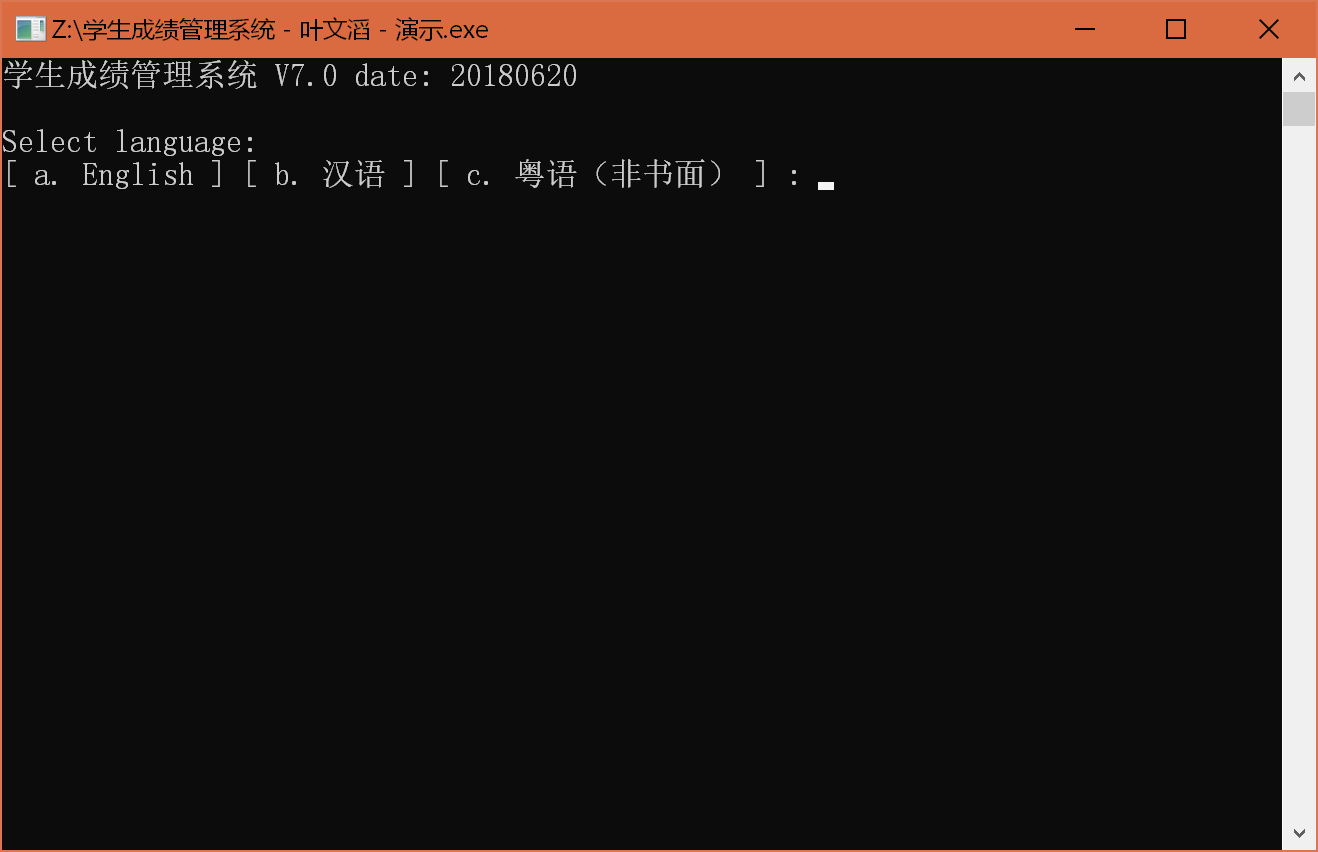
void DisplayRecord(struct link \*head, int ClearScreen) // 负责数据显示模块的功能

# 整体设计

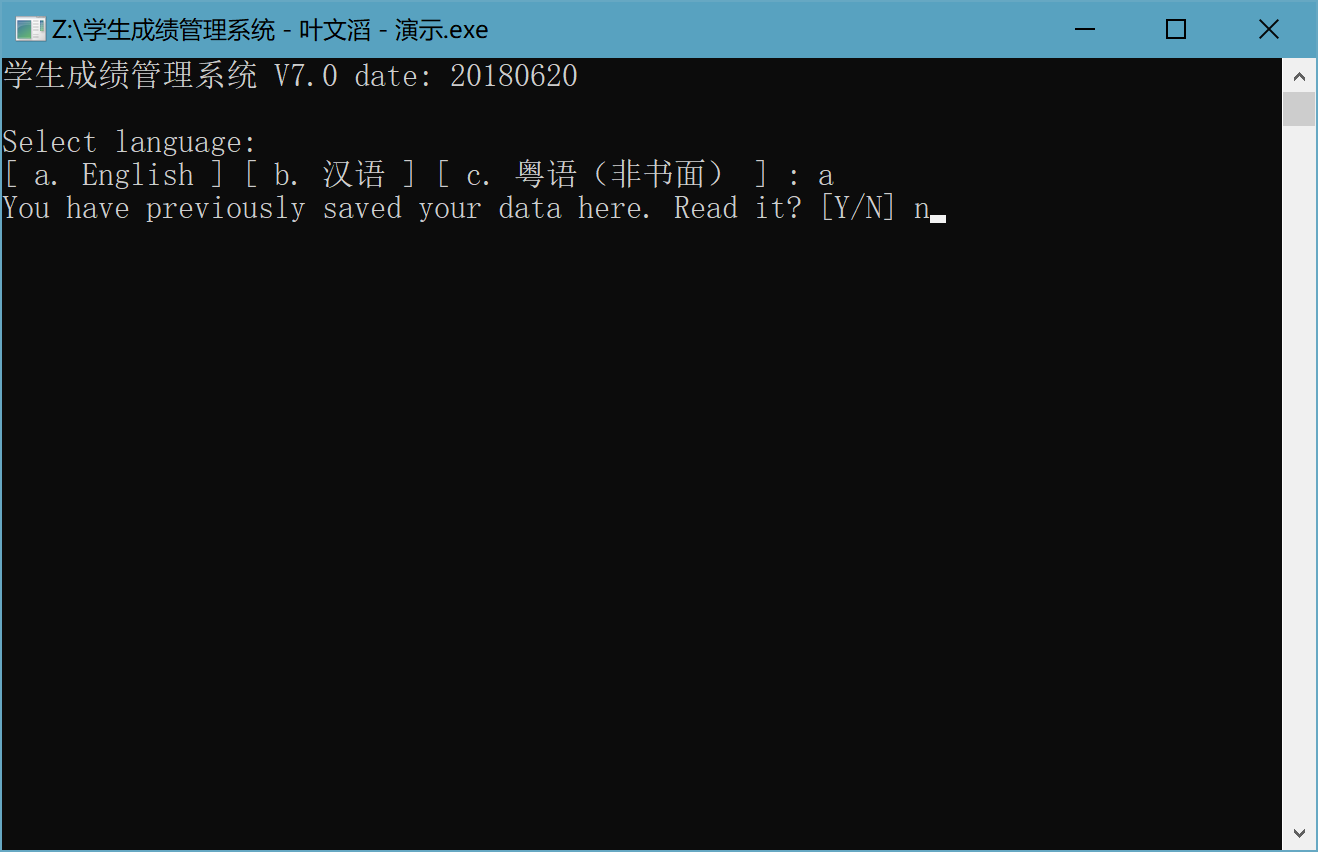


# 详细设计及运行结果

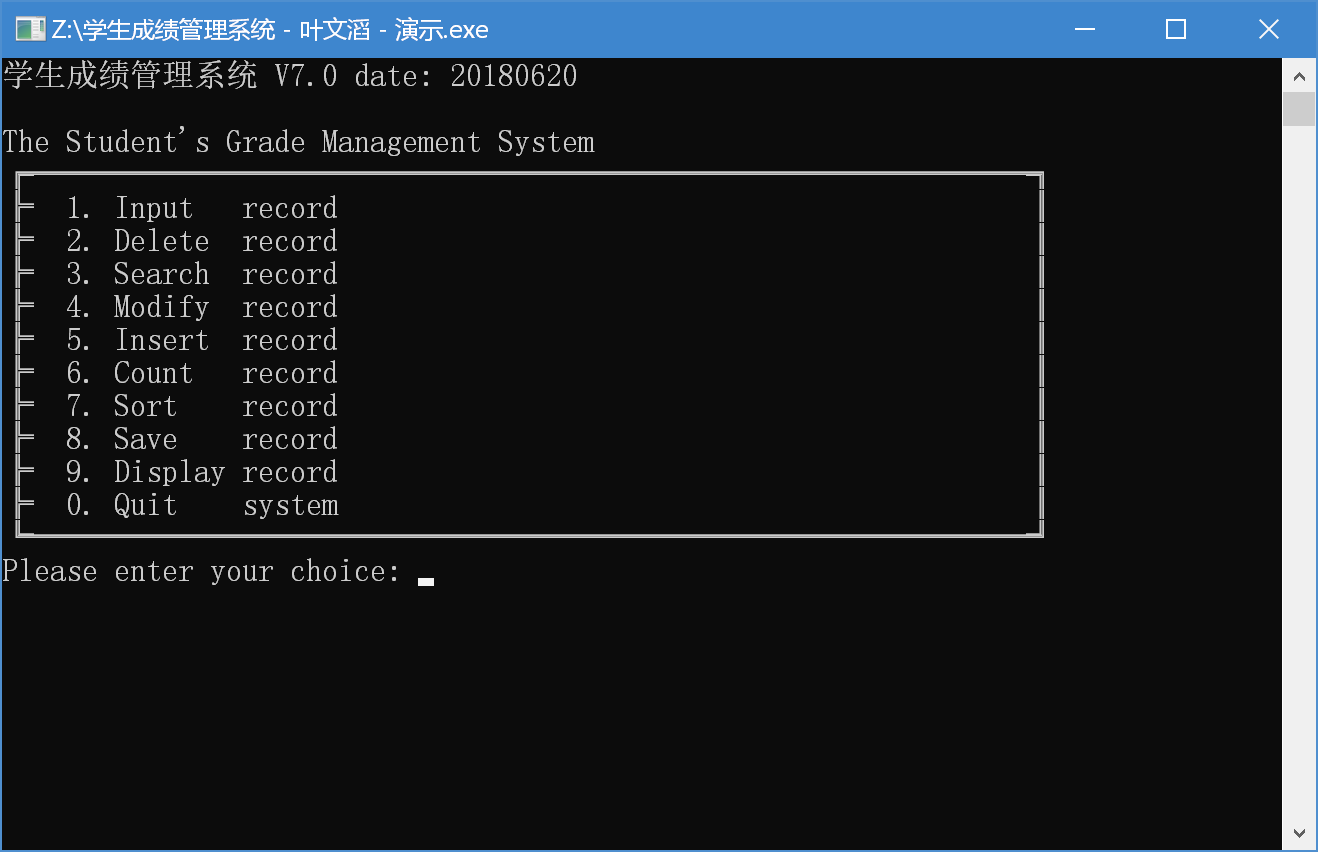
程序进入后，会首先询问语言。此处我们选择英语。



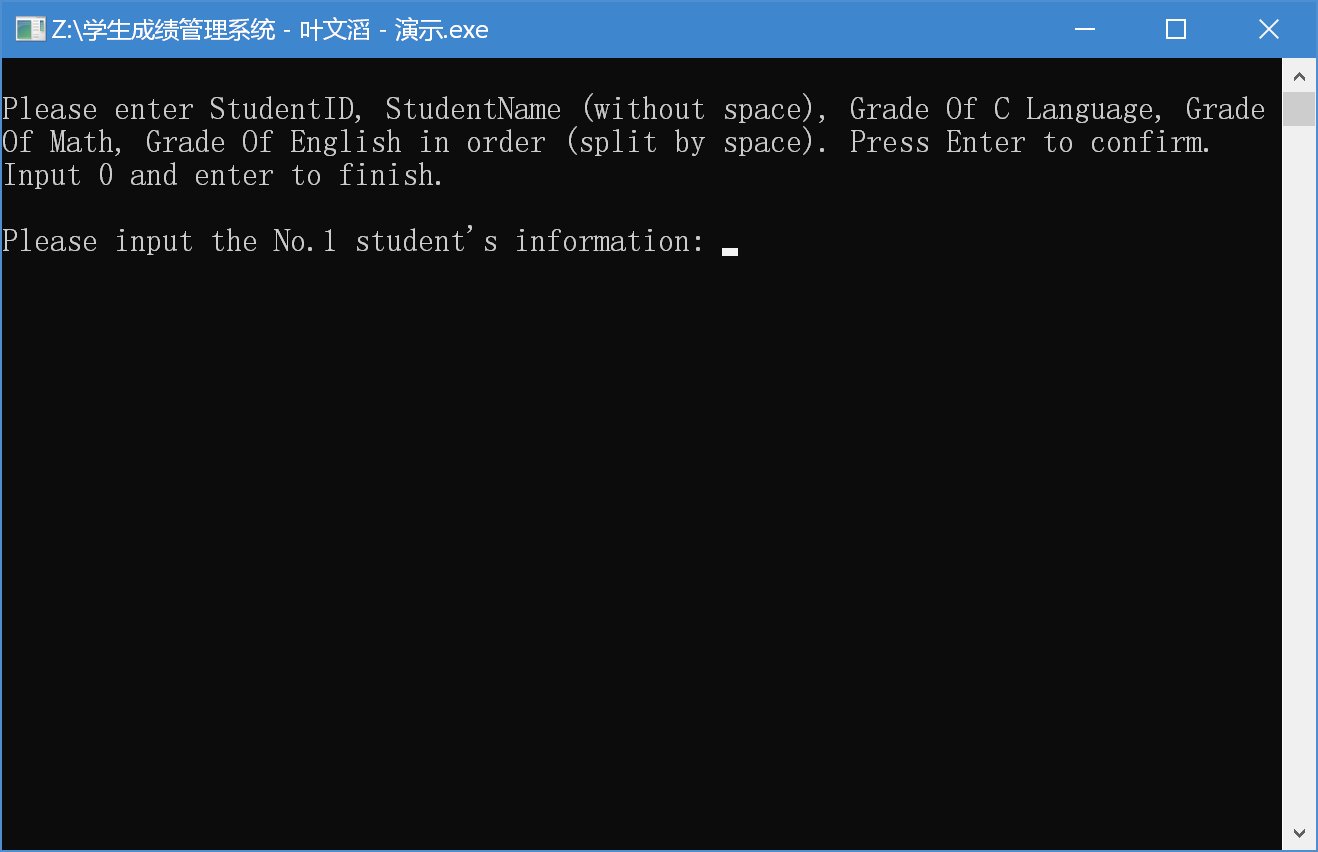
如果此前已有存档记录，程序会询问是否需要读档。此处我们选择否。



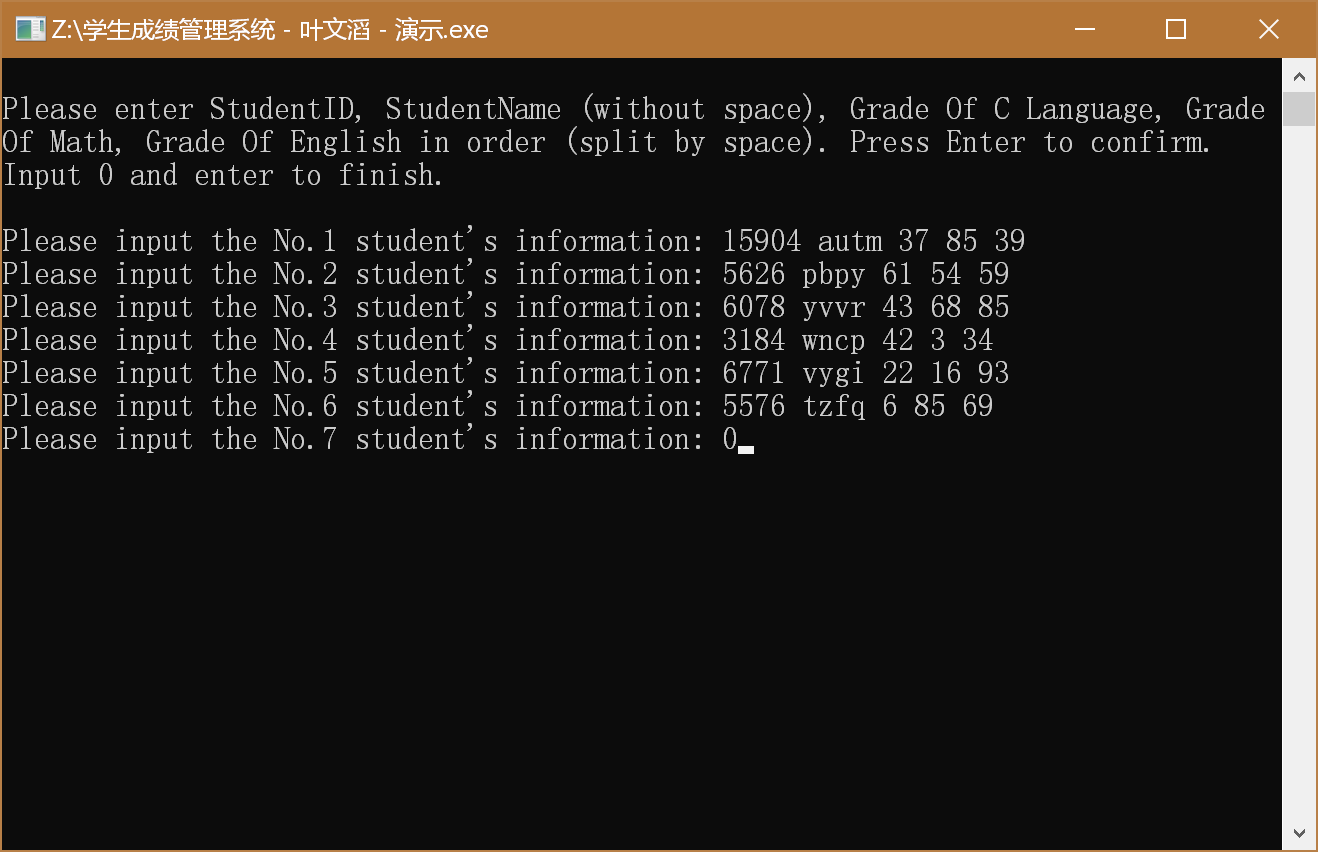
接下来将进入程序主界面。



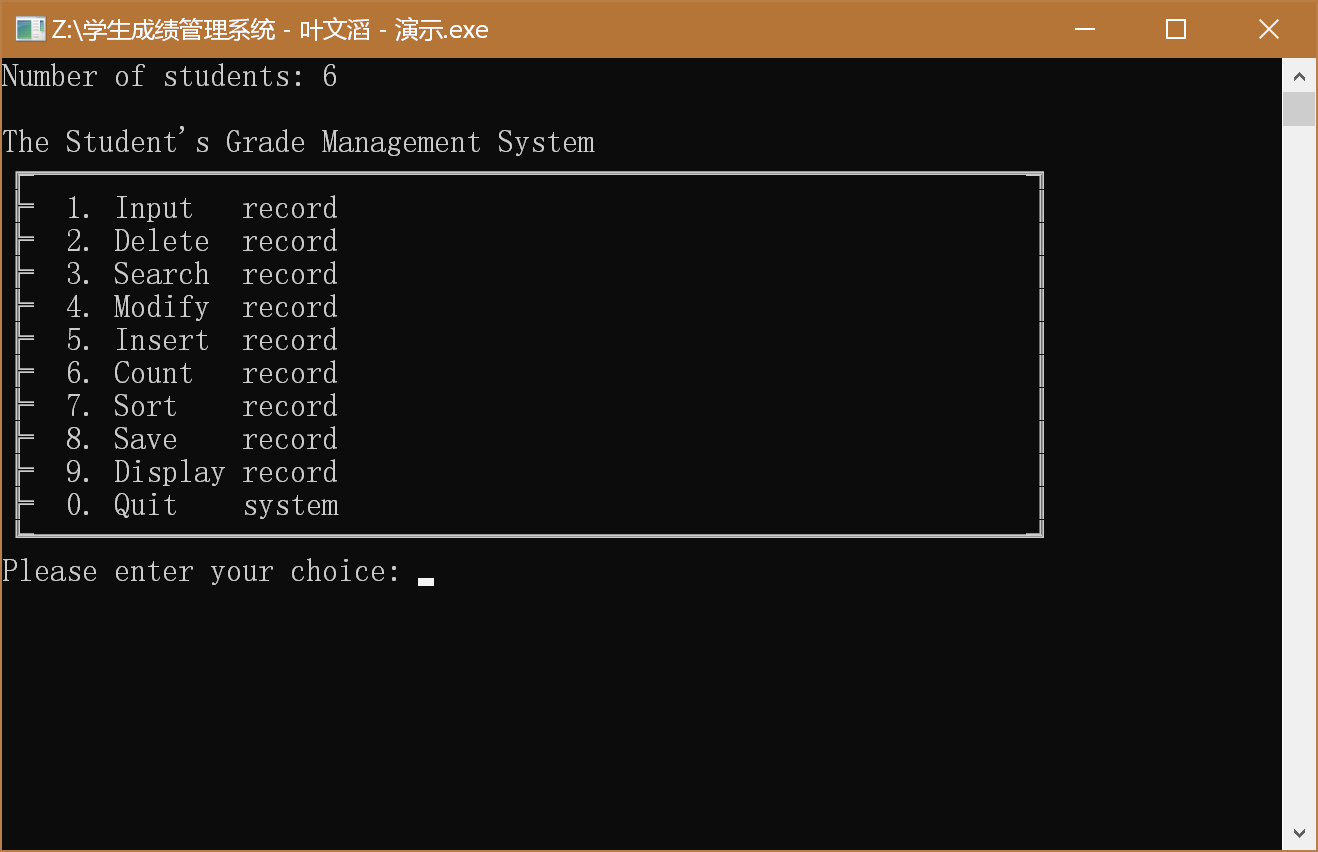
首先我们需要输入数据，输入 1 进入此功能。



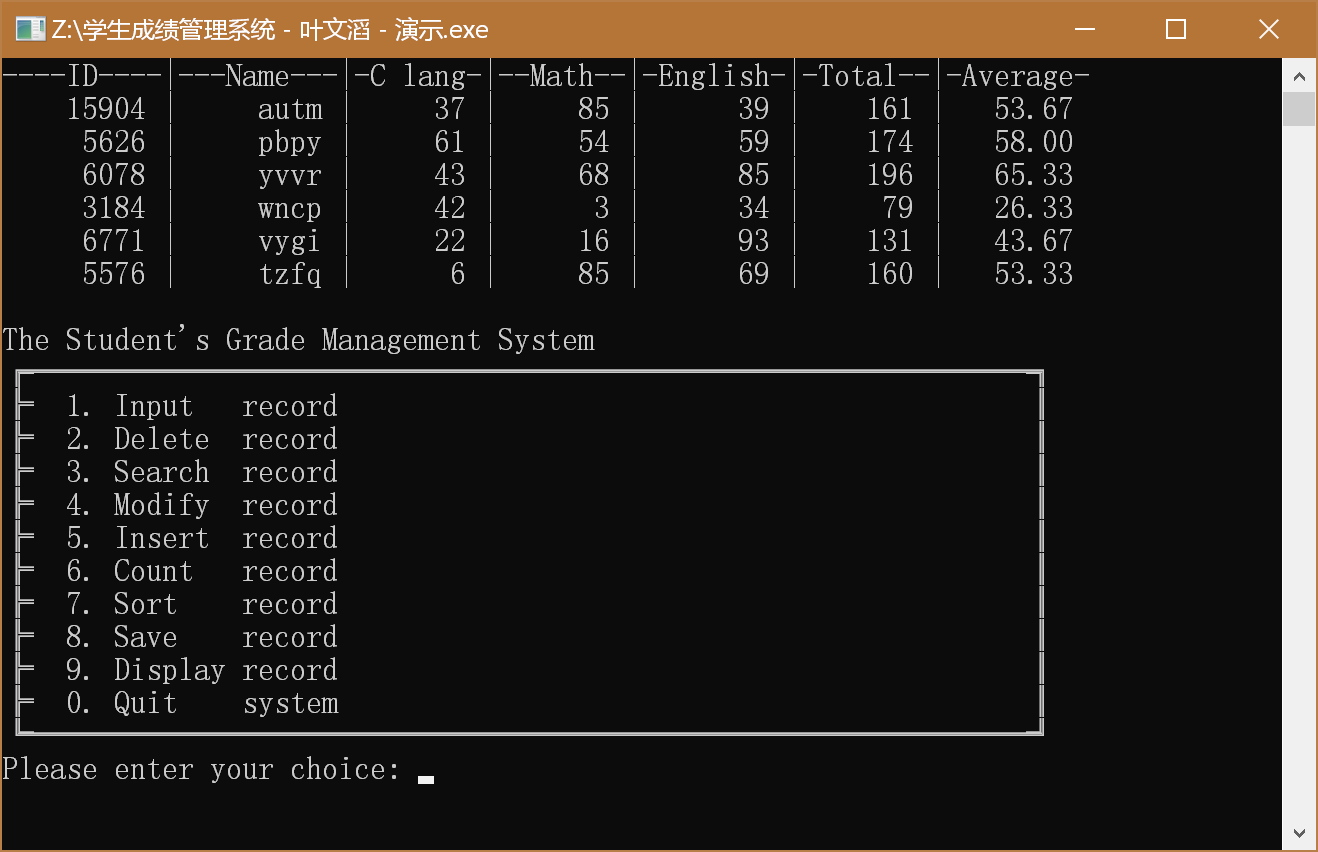
根据程序提示输入数据，并在最后输入 0 以结束输入。



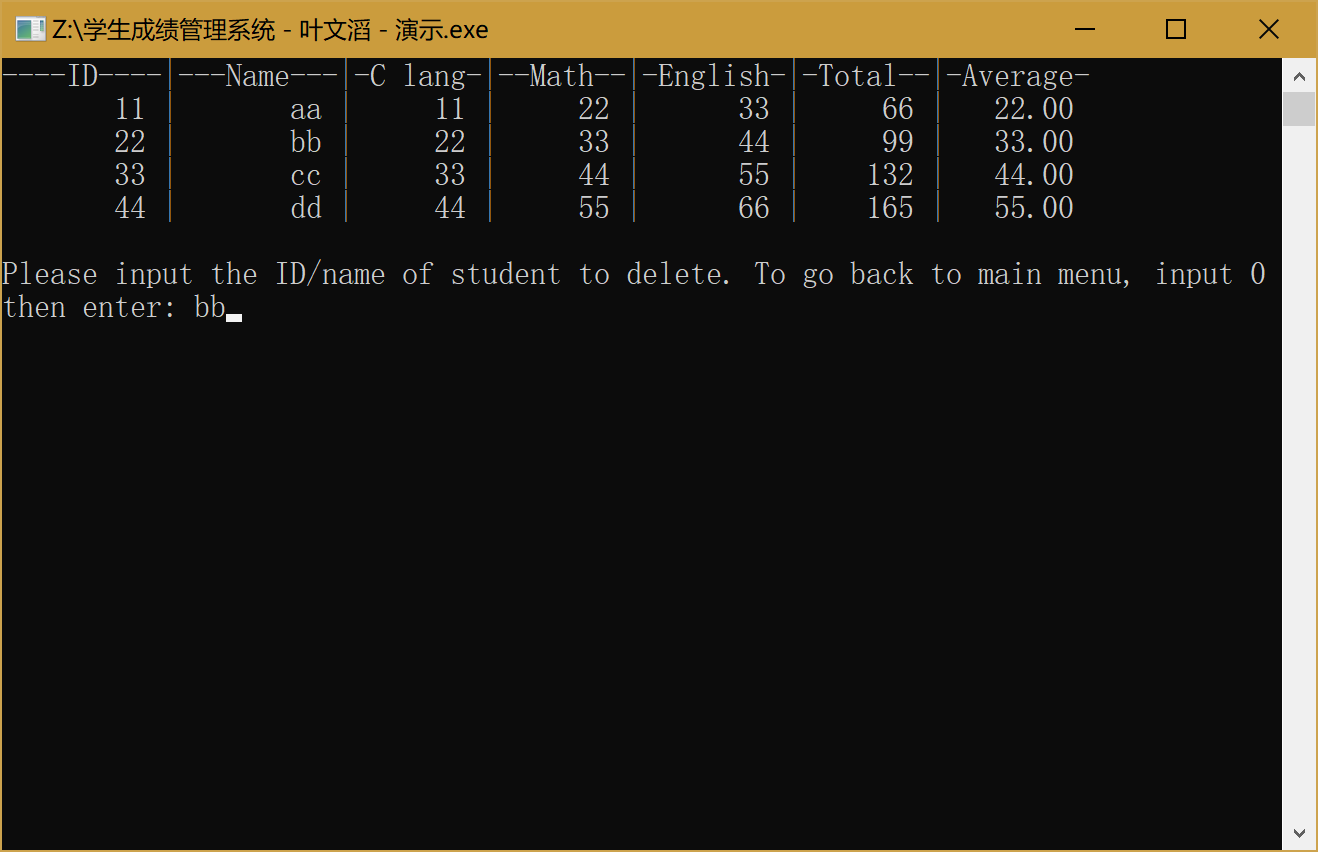
输入结束后，程序自动提示学生数量。



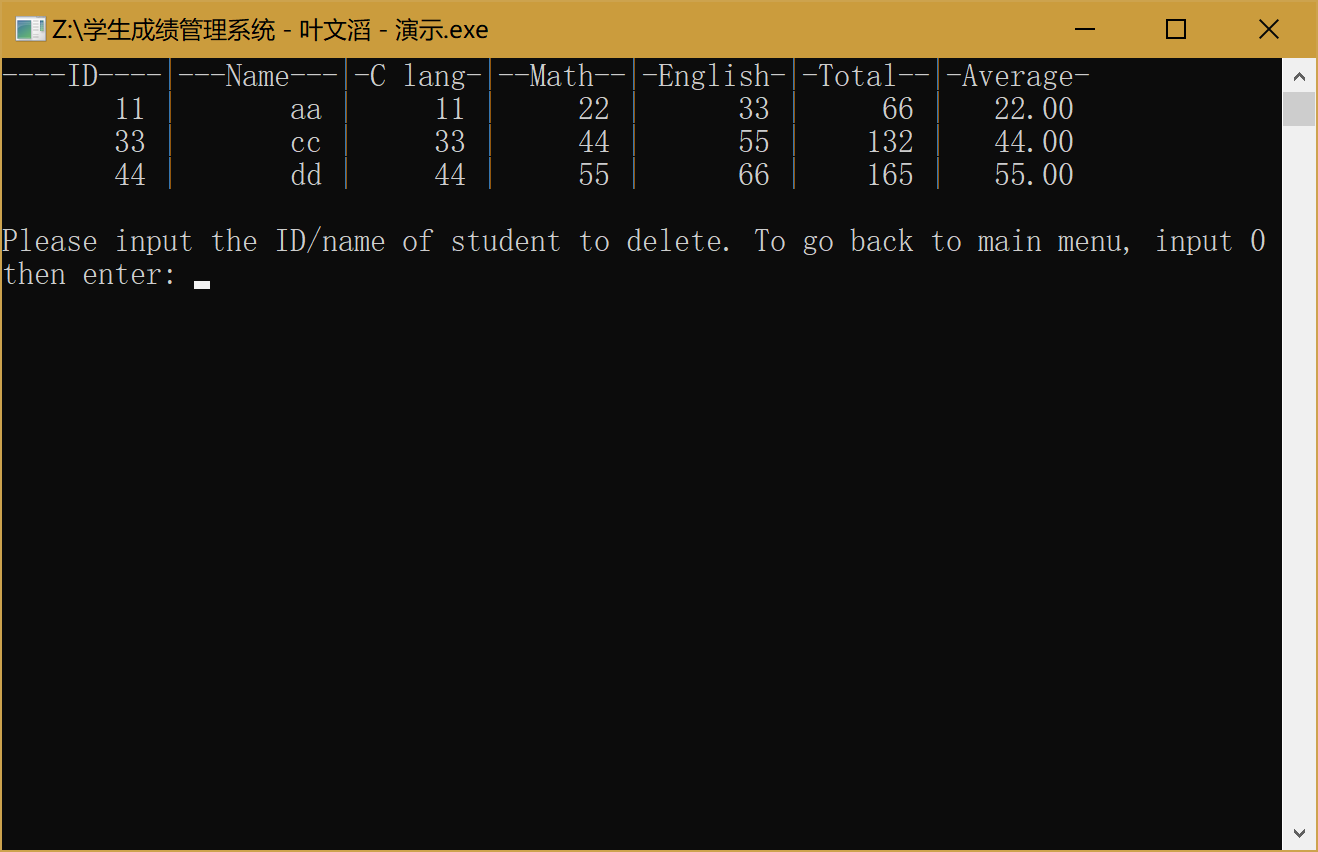
输入 9 以显示当前数据。



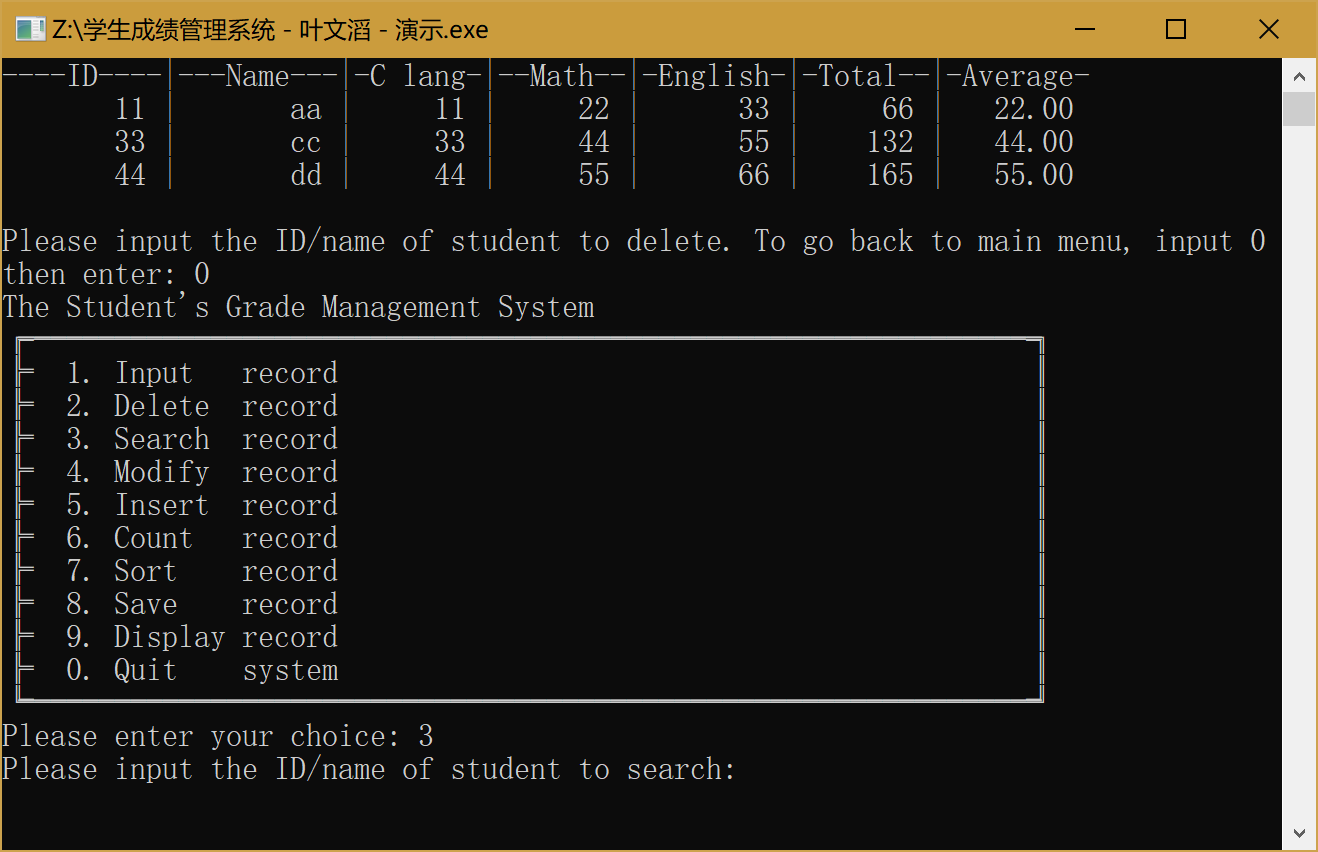
若需要删除学生，输入 2 进入删除功能，然后输入该学生的学号/姓名以删除。



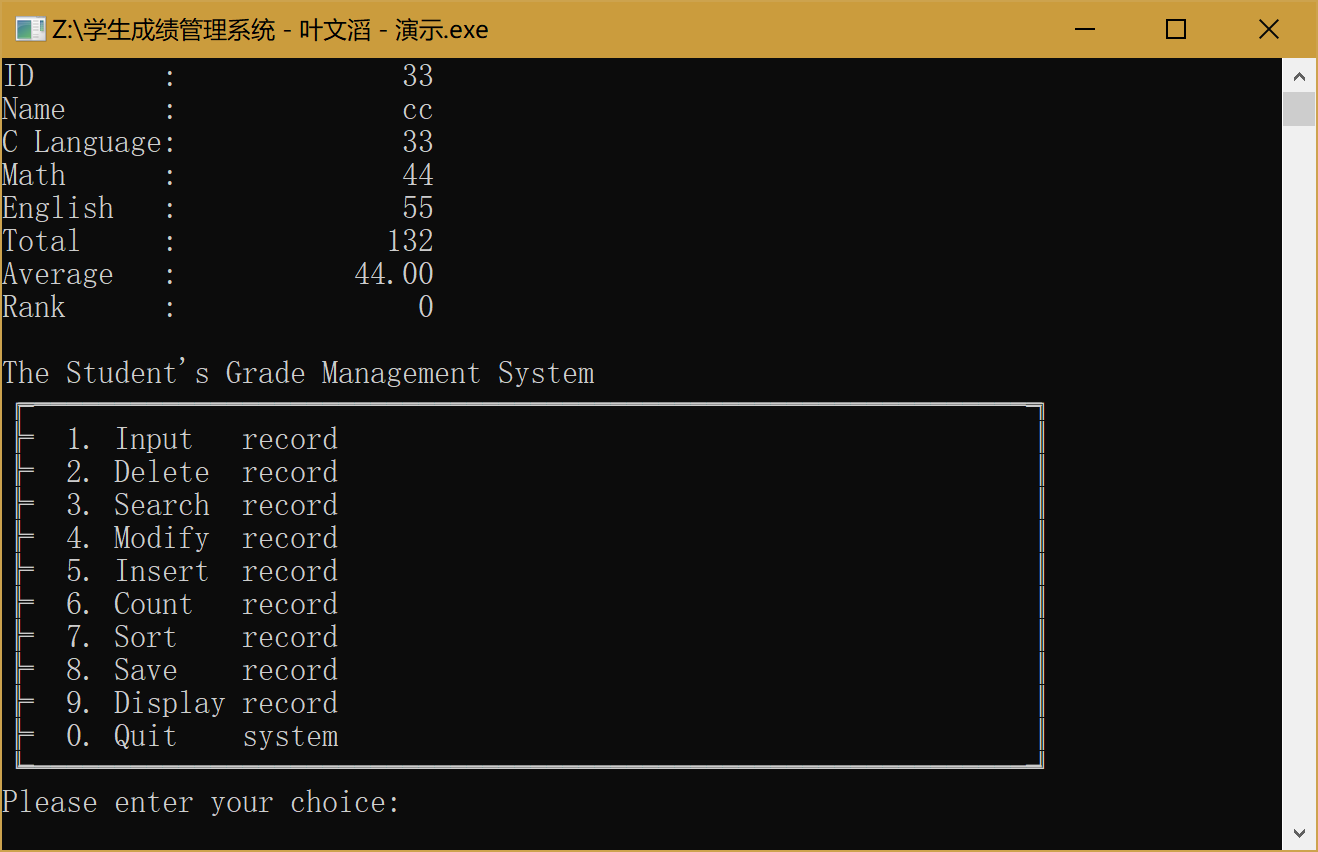
删除结果。



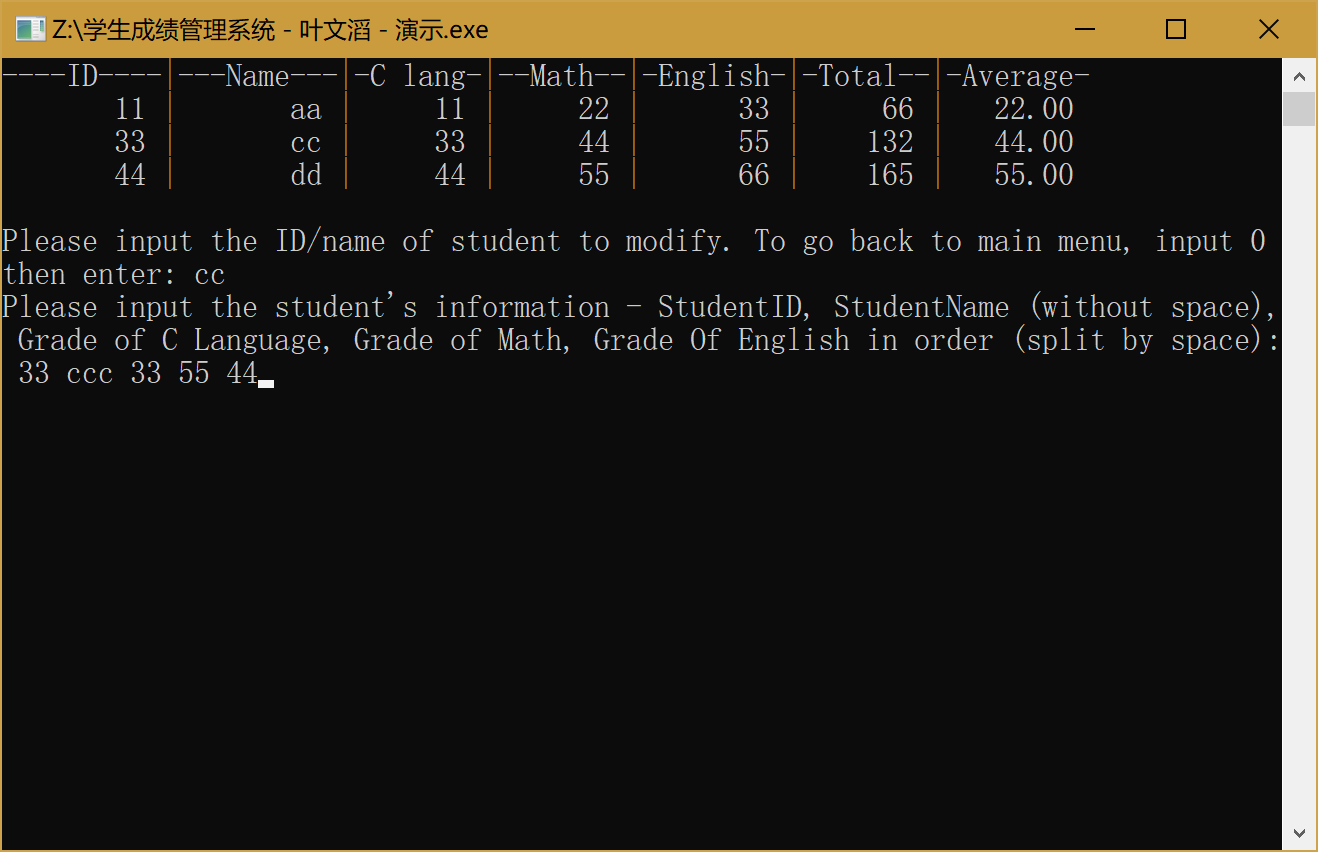
搜索模块与删除模块界面类似。



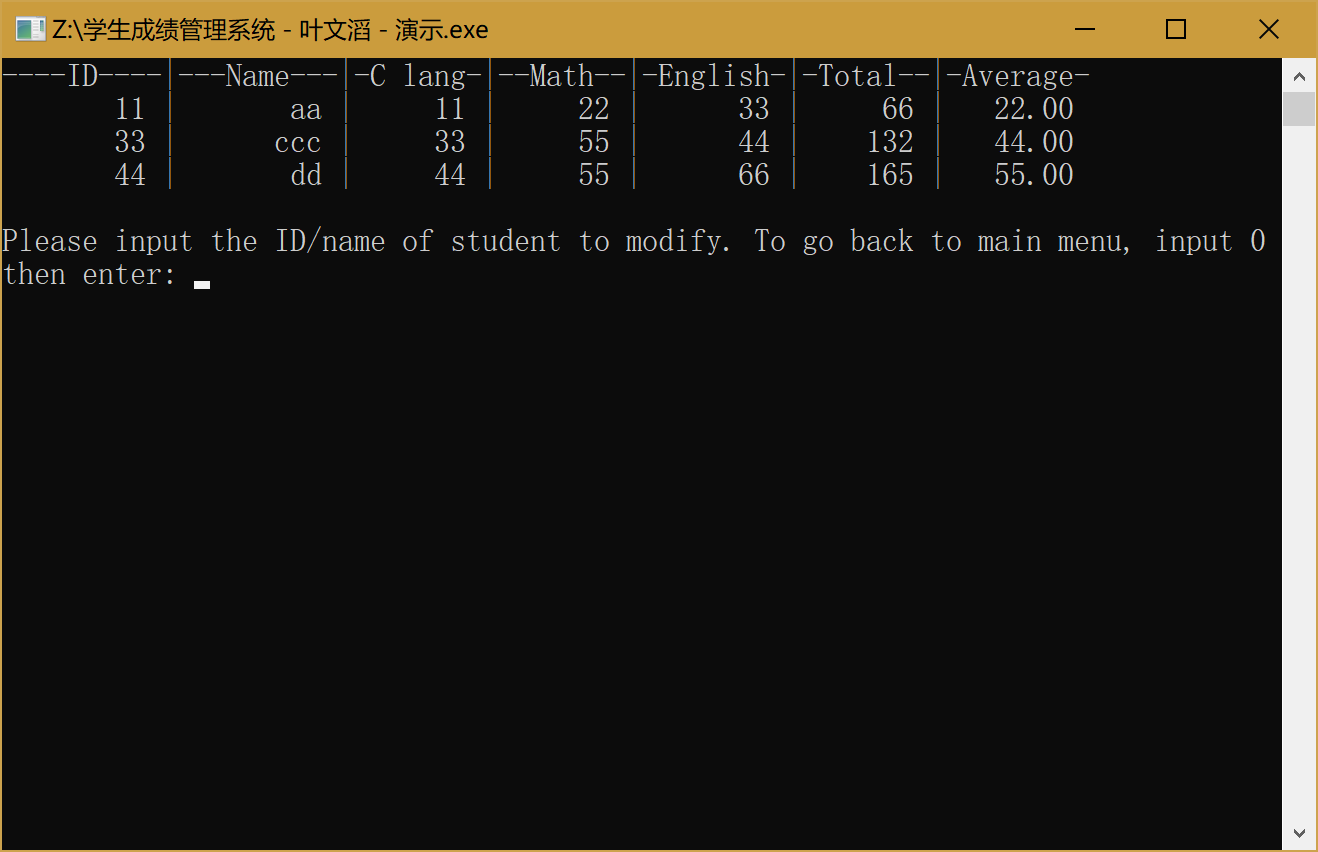
输入学号/姓名以查询对应人员。



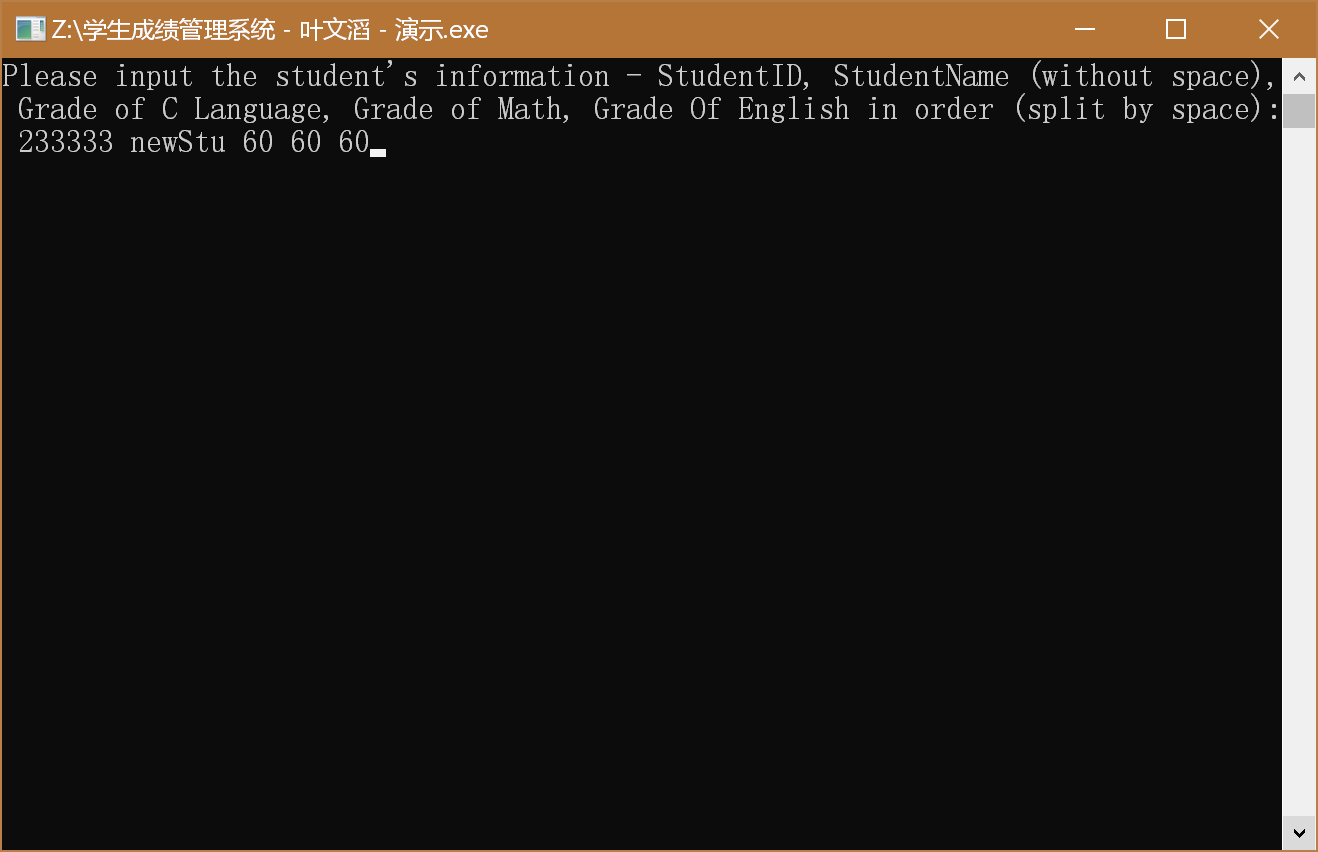
修改模块类似。



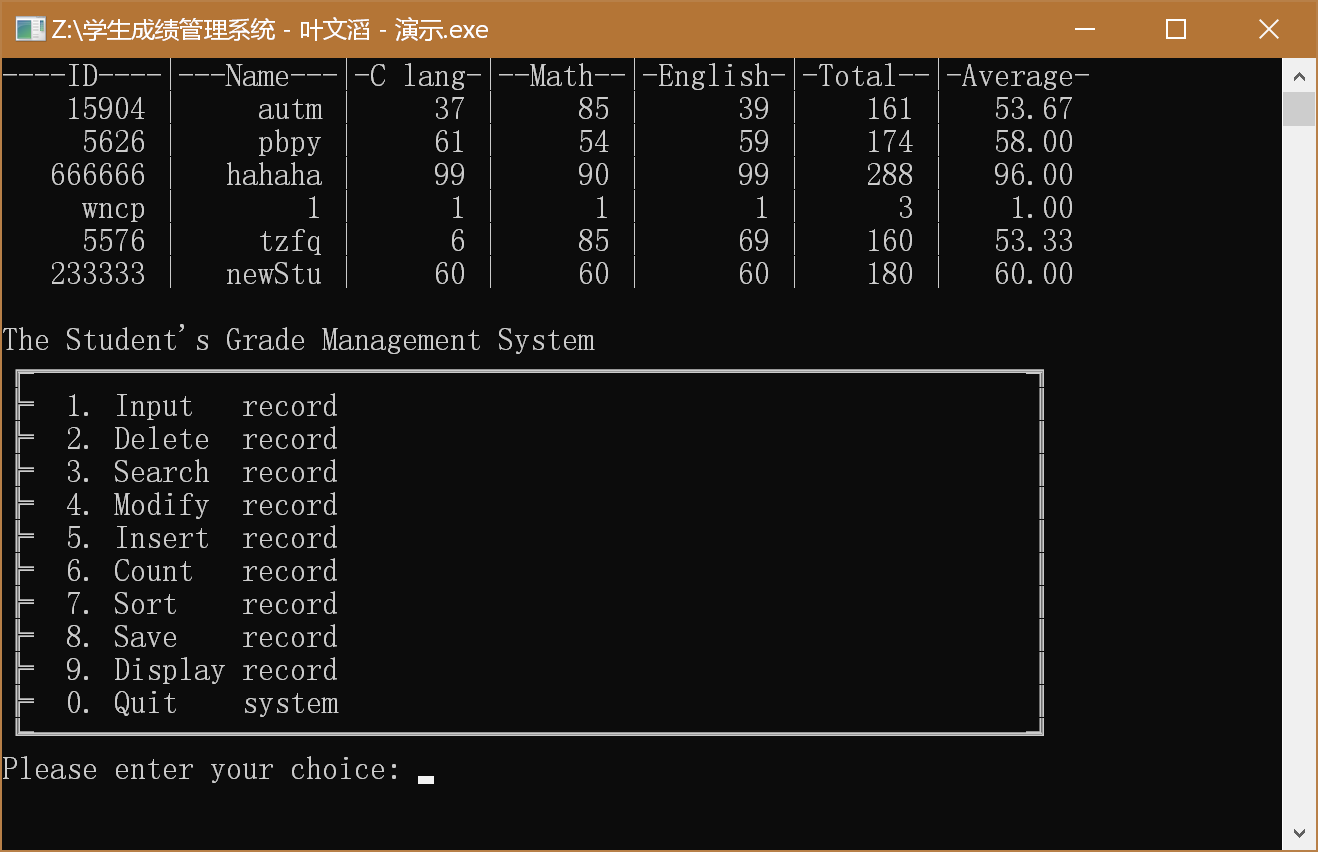
修改结果。



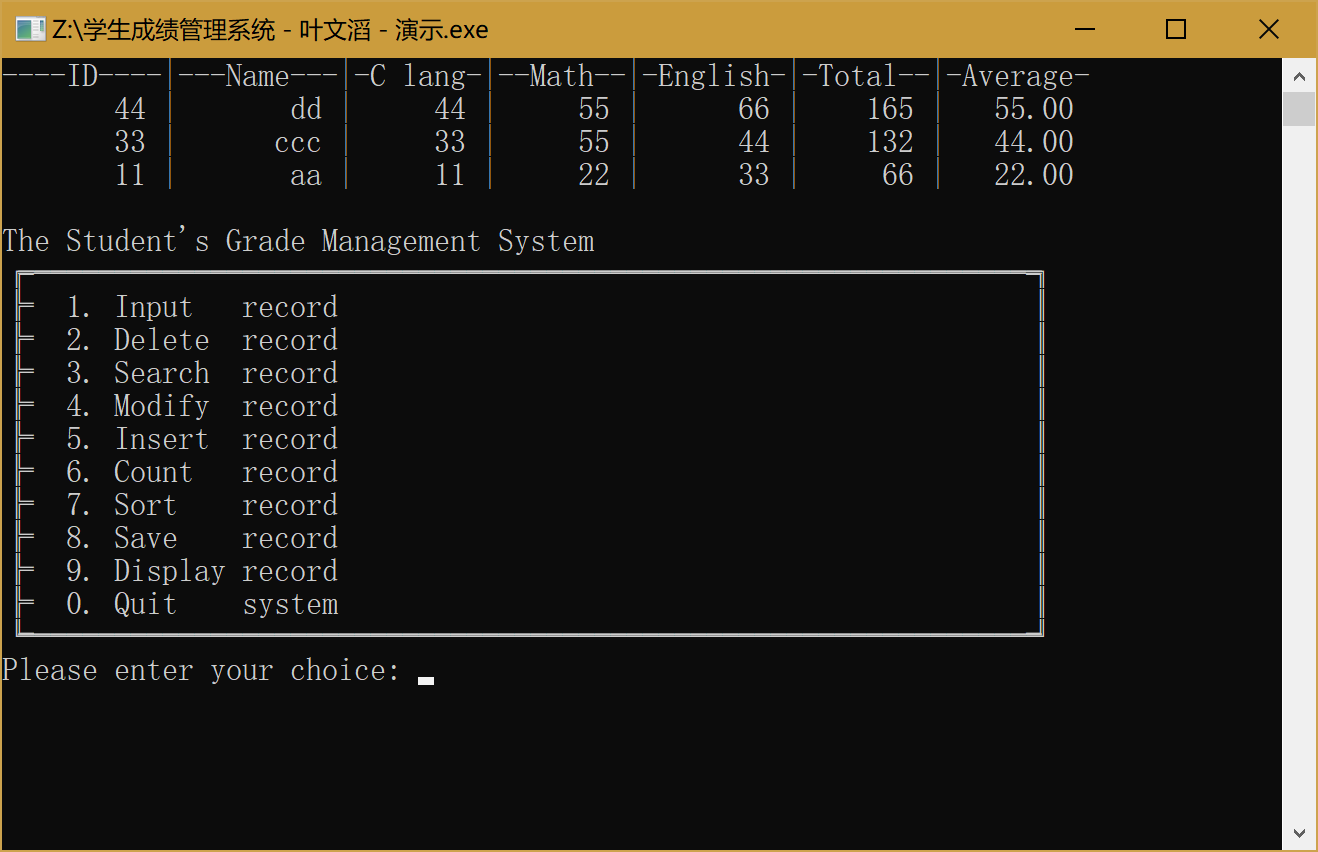
插入功能界面类似。



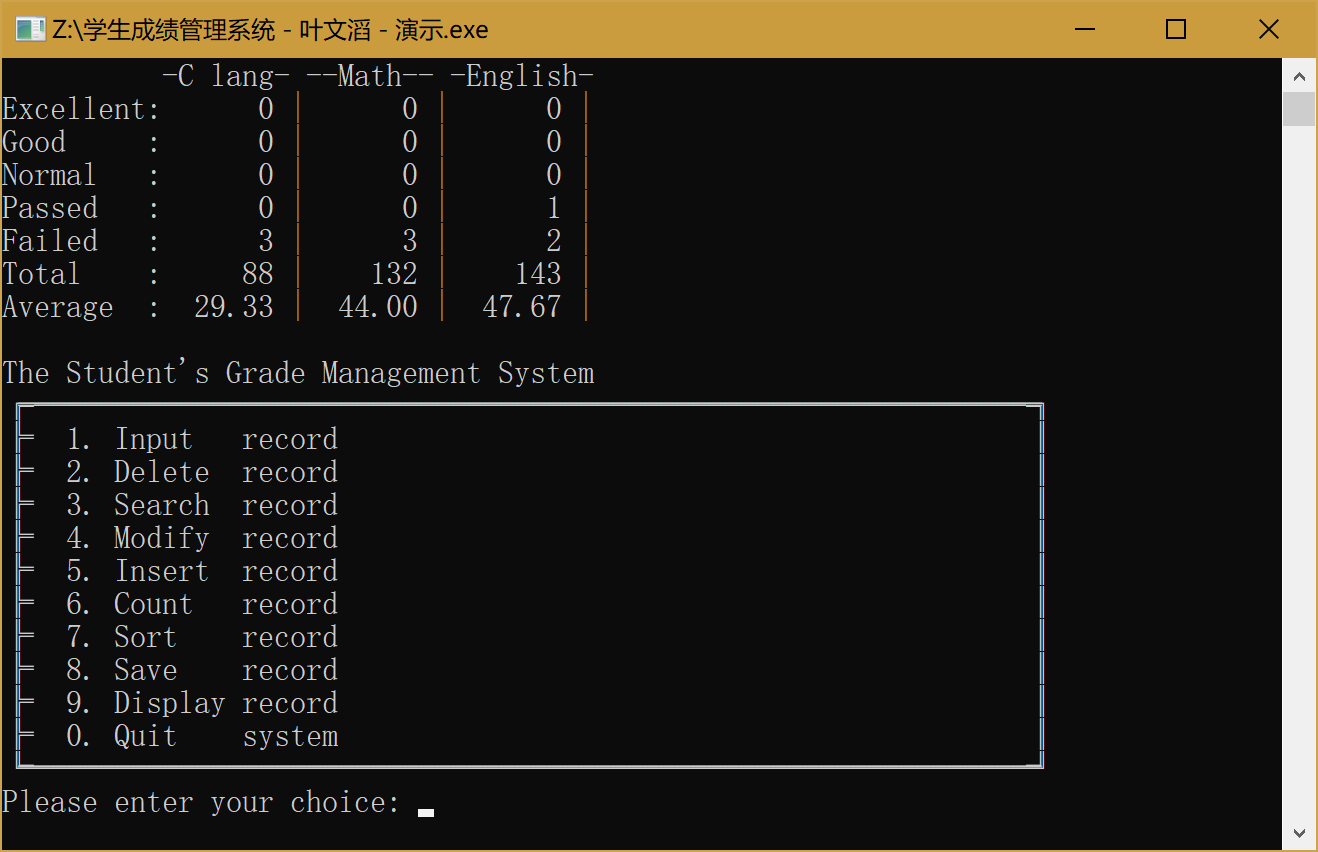
插入结果



排序功能将对总分进行降序排序。



统计功能将统计各科总分和平均分



保存功能。



退出程序界面。



# 设计总结

　　这次的程序设计，与其说是在实训周内完成的，不如说是花了一个学期完成的，因为程序内所涉及到的界面和部分数据结构继承自之前的 6.0 版本作品，而 6.0 版本包括了从该门课程开始到文件操作章节的所有内容，这些都是需要花时间学习的。在 7.0 版本中，核心数据结构由结构体数组更改为结构体×链表，这是一个较大的改动。在程序的编写、修改、除错过程中，我学会了很多，也积累到了很多经验，这些经验无论是对 C 语言的学习还是日后其他语言的学习都十分有帮助，收获颇多。由于时间有限，本程序仍有一些不足，如在搜索和插入功能中仍未能实现“输入 0 回到主菜单”这种较为方便的操作方式。希望这些缺陷在日后的学习中能够被逐渐攻破。