**网 络 空 间 安 全 学 院**

**实践课程报告书**

**课程名称 　　程序设计基础实践**

**题 目 学生成绩管理系统**

**专业班级 21网络空间安全1班**

**组　长：学号** 2021441010122 **姓名**  **彭泽亮**

**组员１：学号** 2021441010116 **姓名**  **龙鑫智**

**组员２：学号** **姓名**

**指导教师** 牛熠 教授

**完成时间** 2022 年 3 月 19 **日 至**  2022 年 6 月 8 **日**

**一、系统概述**

**要求：**主要针对系统功能进行概述。

本系统使用C语言编写，实现对本科生和研究生的一系列信息的管理功能，内容包括增加、删除、修改、查询、计算、排序等。用户只需稍微修改记事本文件便可达到方便地管理学生信息的目的。

**二、系统功能结构图**

**要求：**[1]. 设计系统功能结构图，并针对每个功能模块用文字简单描述其功能。

　[2]. 注明小组各成员的合作分工情况，即合作完成部分、独立完成部分。图示

描述已自动生成

**三、系统数据结构说明**

**要求：**定义系统数据结构（如：链表、文件），链表需说明每项数据的含义，文件需说明文件类型。

文件：

demo.c 主函数源文件

dataStruct.h 存放数据结构的头文件

menu.h 存放菜单及一系列相关函数的头文件

lxz.h 组员完成的其他功能函数的头文件

underGraduate.h 本科生测试数据

postGraduate.h 研究生测试数据

**enum** Sex{female,male};          // 0为女，1为男

typedef struct underGraduate    //本科生结构体

{

    int id;                     //学号

    char name[16];              //姓名

    enum Sex sex;               //性别

    char profession[30];        //专业

    char class[30];             //班级

    int score[6];               // 0-5分别为 数学 英语 C语言 总分 班排名 校排名

    struct underGraduate \*next; //下一个节点

}SUG;

**typedef** **struct** postGraduate     //研究生结构体

{

**int** id;                    //学号

**char** name[16];             //姓名

**enum** Sex sex;              //性别

**char** profession[30];       //专业

**char** class[30];            //班级

**char** direction[30];        //研究方向

**char** teacher[16];          //导师

**int** score[5];              // 0-4分别为综合课程 论文 总成绩 班排名 校排名

**struct** postGraduate **\***next; //下一个节点

}SPG;

**四、自定义函数设计**

**要求：**[1]. 必须按成员依次列出每位成员完成的函数设计。

[2]. 函数描述格式：序号、函数首部、参数列表、返回值、实现功能、算法流程图、设计者。

--------------------------------------------------------------------------------

1．×××函数

* 设计者：学号+姓名
* 函数首部：
* 参数列表：[1] aaa，表示……

[2] aaa，表示……

[3] ……

* 返回值：
* 实现功能：
* 算法流程图：

--------------------------------------------------------------------------------

【**示例**】

1．fact函数

* 设计者：20204101101张三
* 函数首部：int fact(int n)
* 参数列表：[1] int n，表示求解阶乘的正整数。
* 返回值：int型，表示n!。
* 实现功能：计算并返回n!。
* 算法流程图：如图4-1所示。

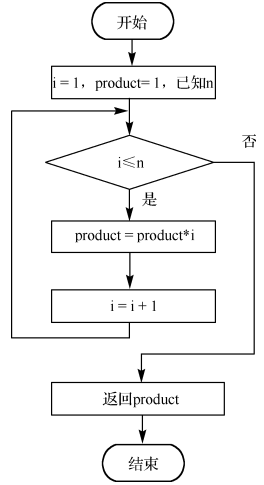


图4-1 求解n!的算法流程图

**五、运行界面**

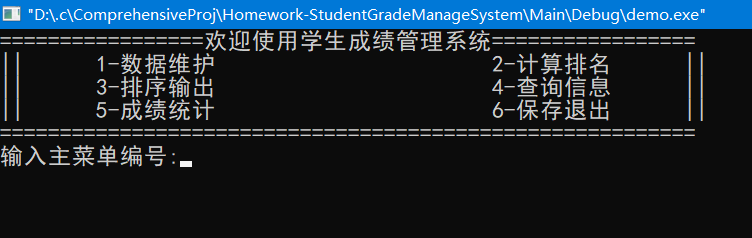
**要求：**[1]. 必须按成员列出每位成员完成程序测试的运行界面，注明成员的学号姓名。

[2]. 必须按菜单系统的顺序，依次给出各项功能的运行界面截图，每个截图必须有测试数据。

--------------------------------------------------------------------------------

一．主菜单

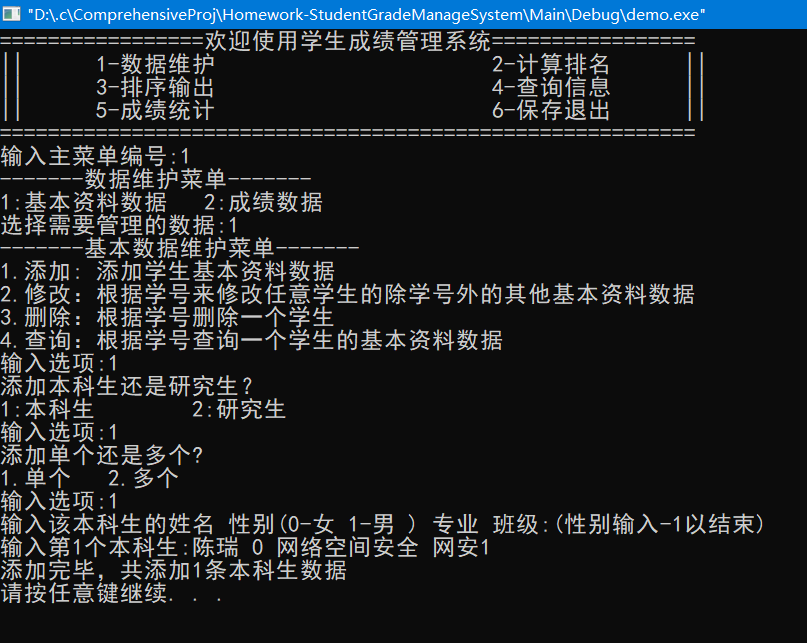
<主菜单运行界面截图>

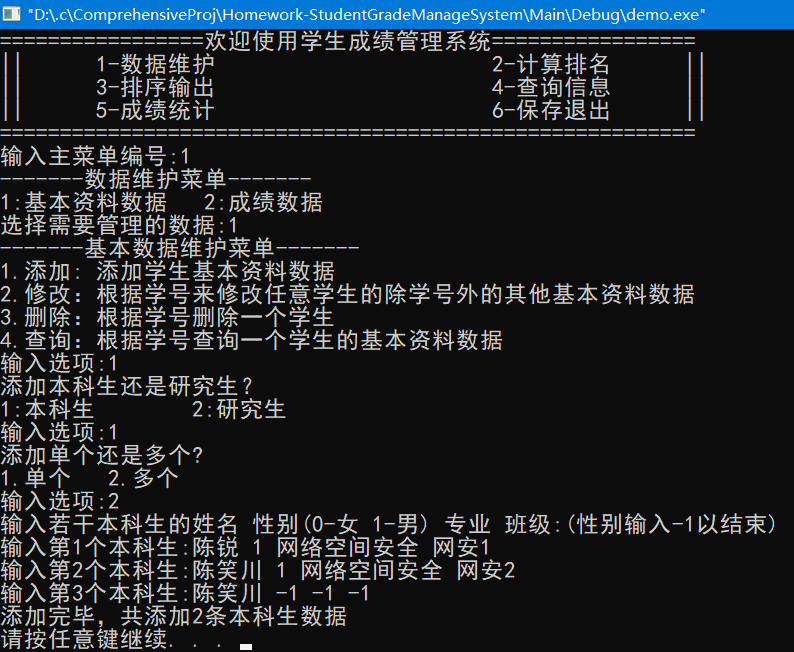


1. **数据维护：**

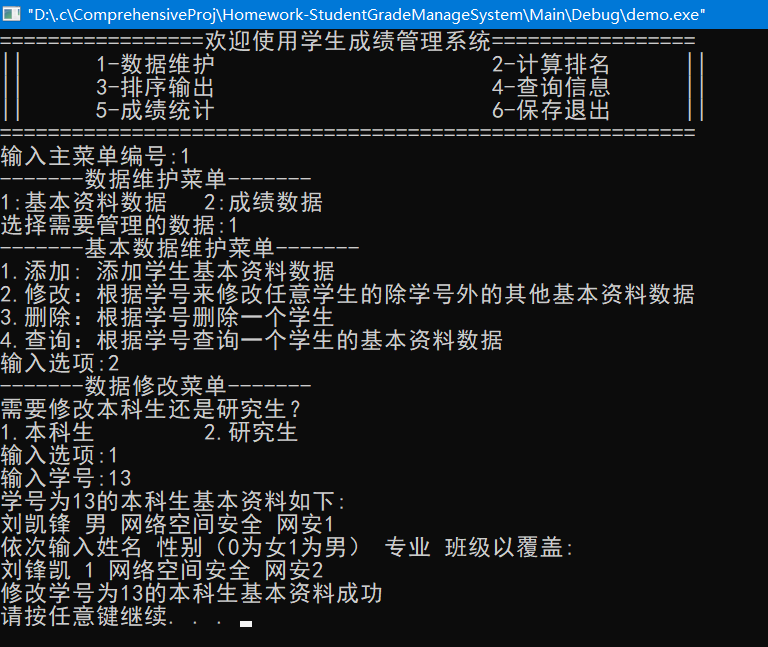
**1.基本数据：**

学生添加界面：

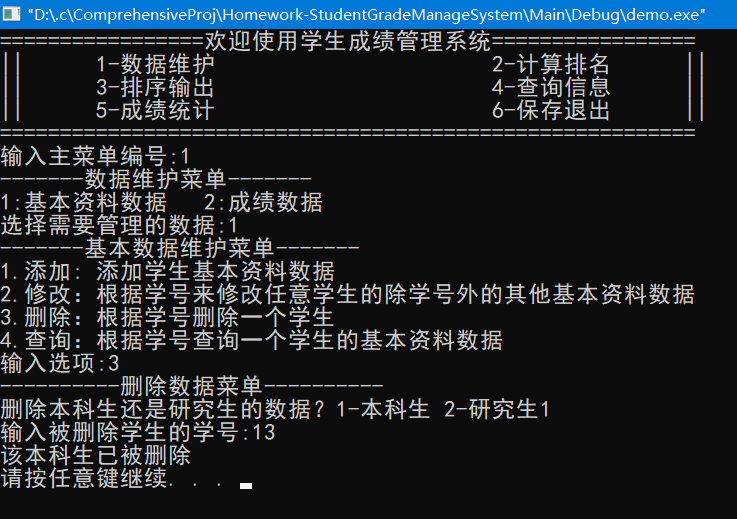




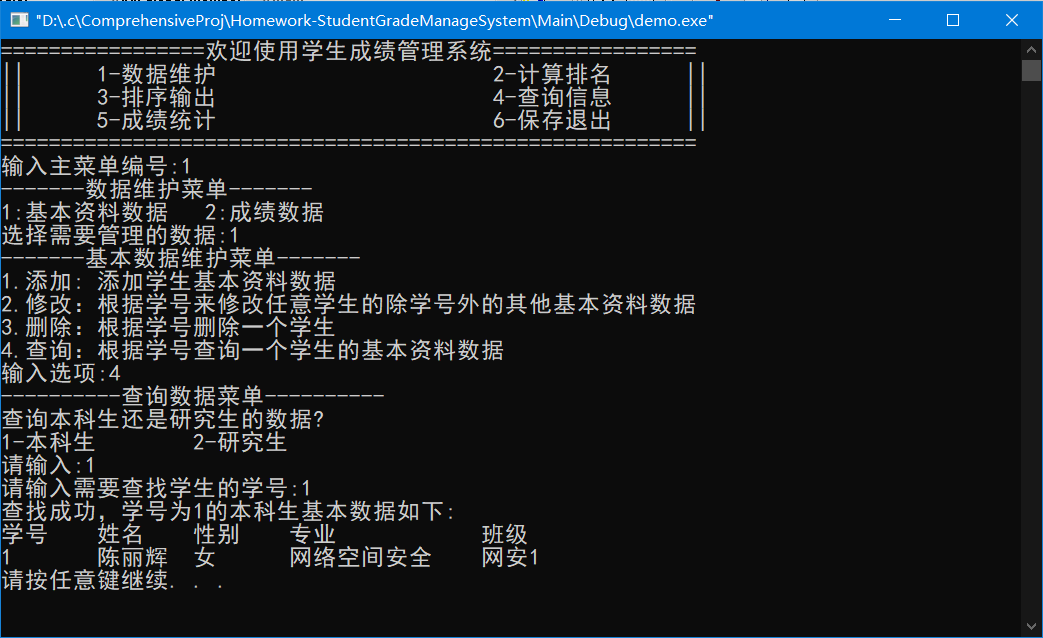
资料修改界面：



资料删除界面：

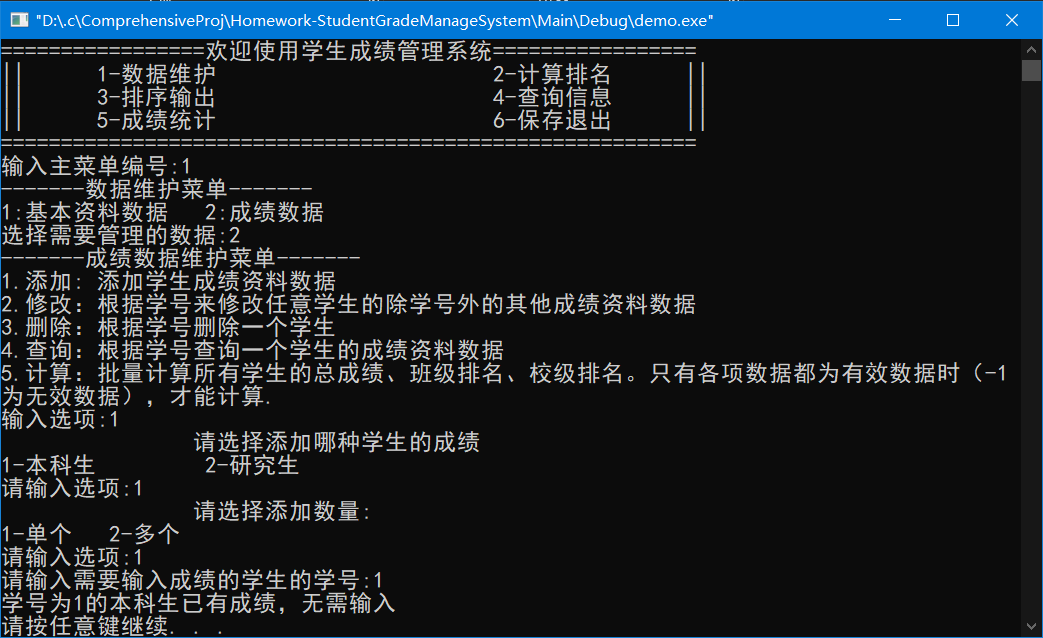


资料查询界面：

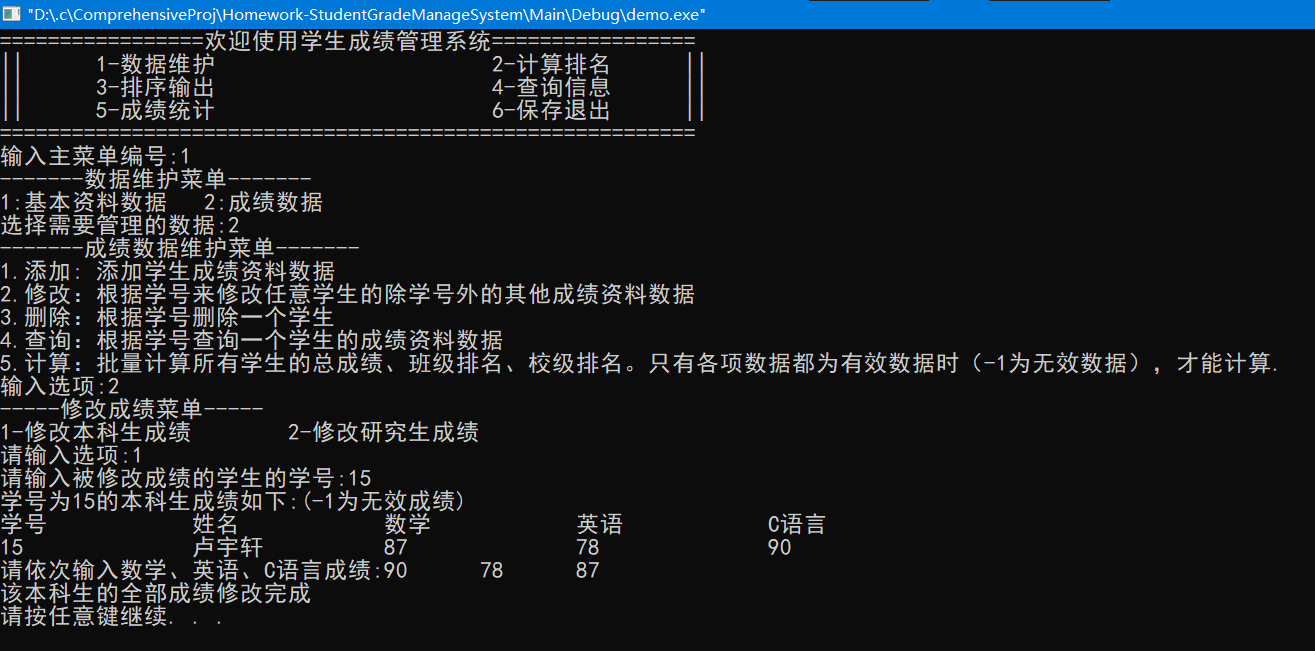


**2.成绩数据：**

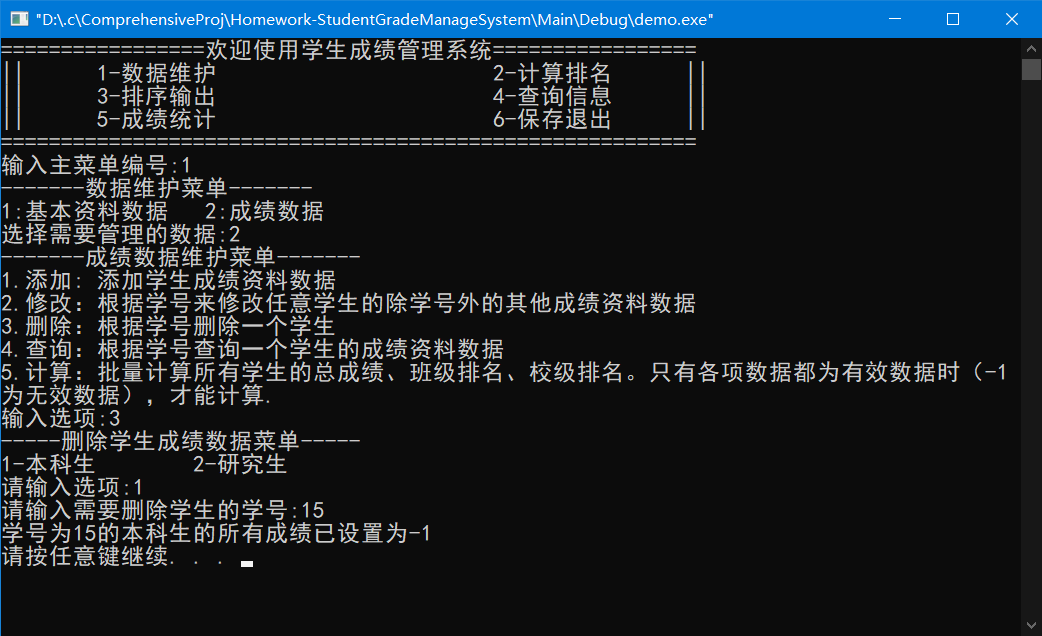
添加界面：



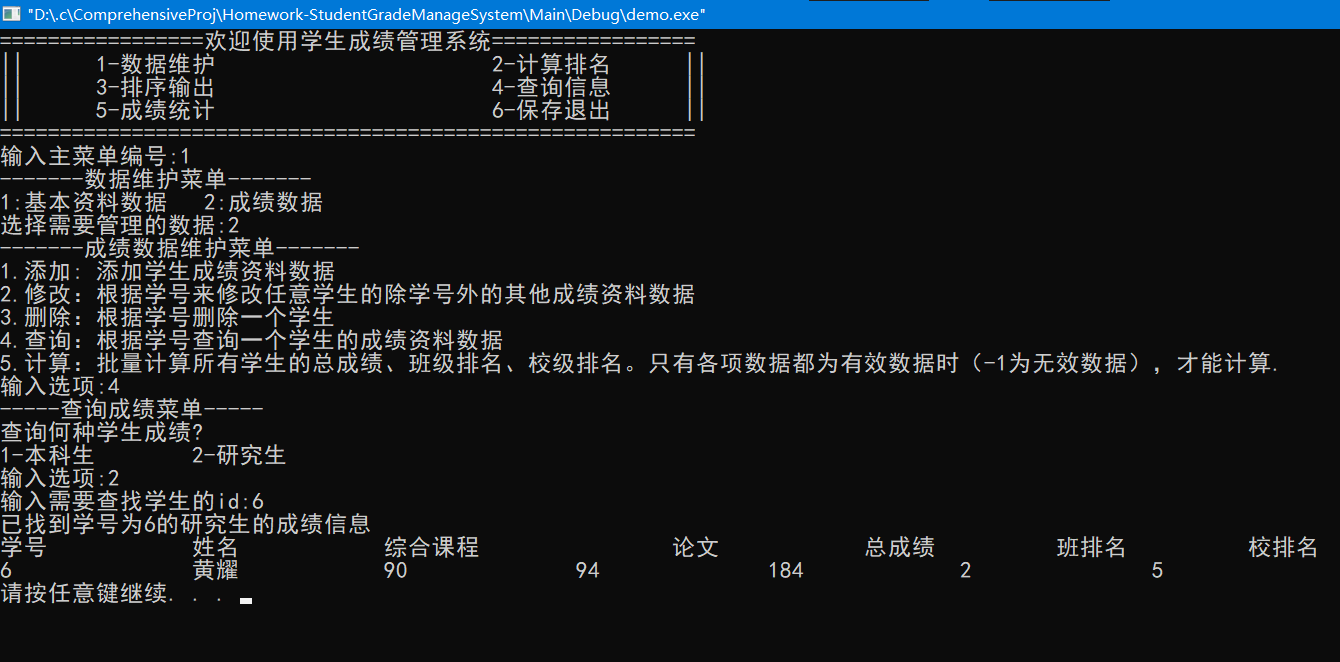
修改界面：



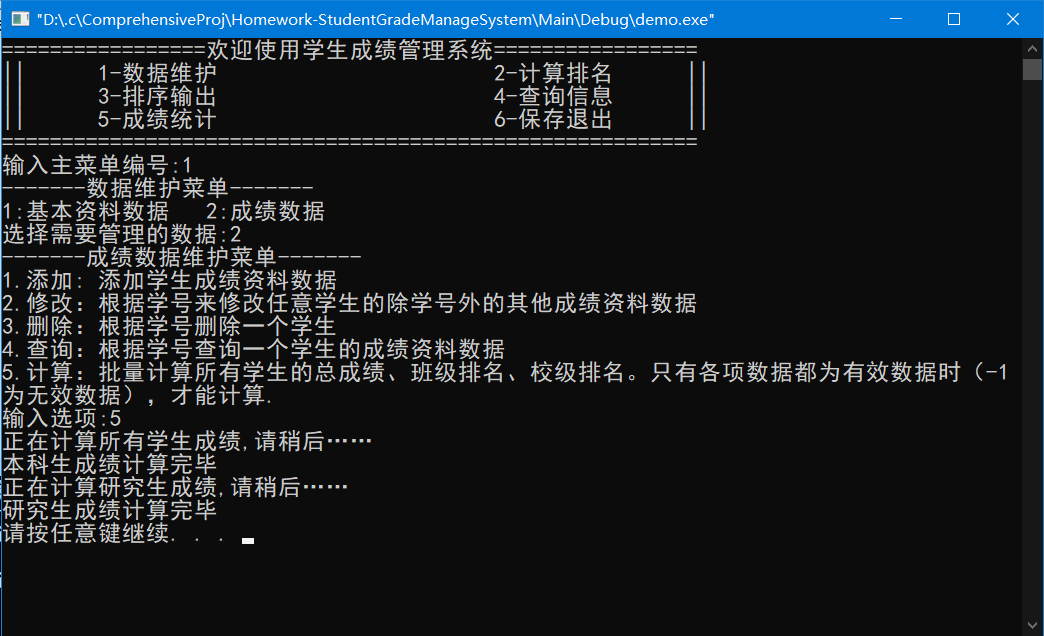
删除界面：



查询界面：

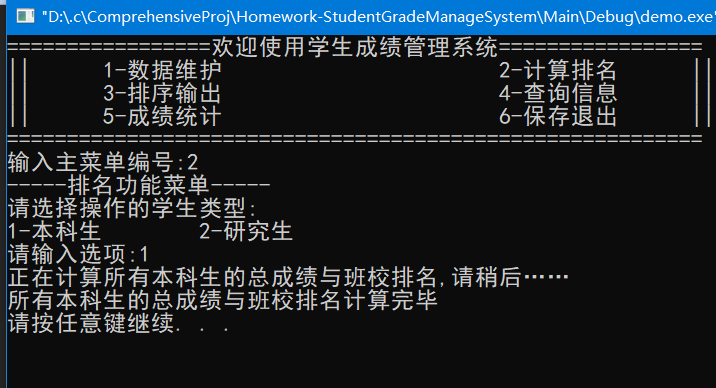


计算界面：

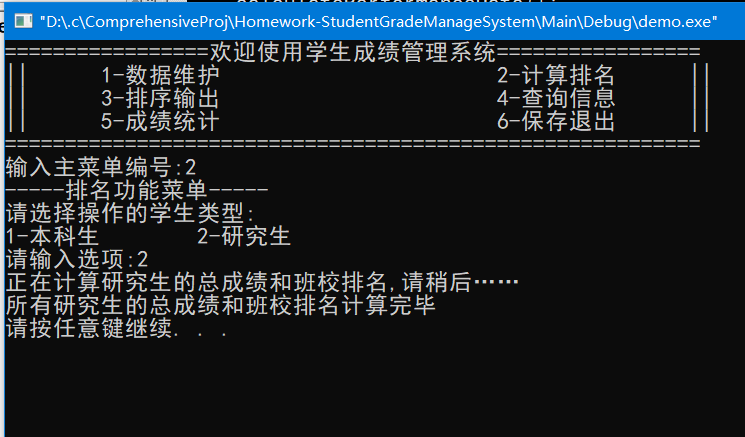


1. **计算排名**

1.计算本科生

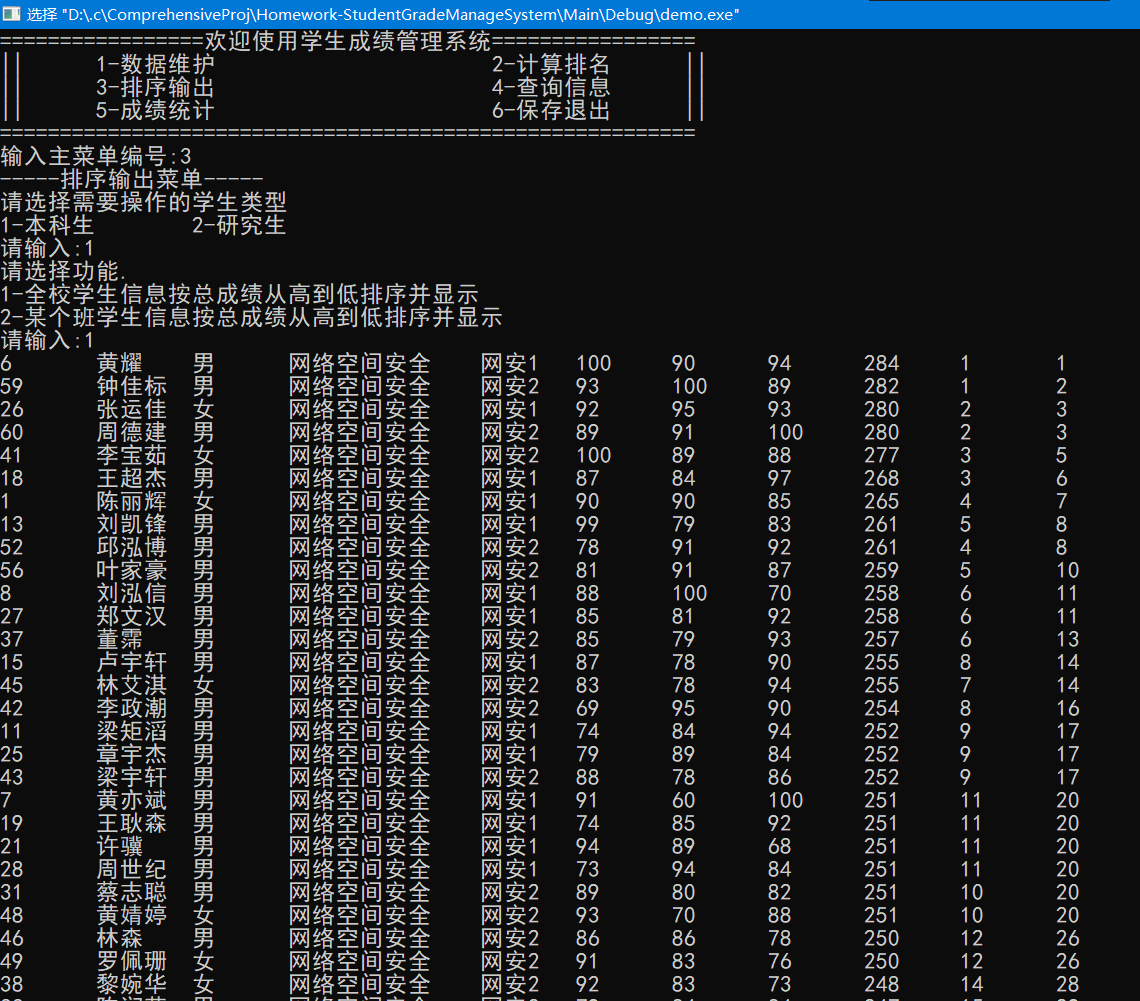


2.计算研究生

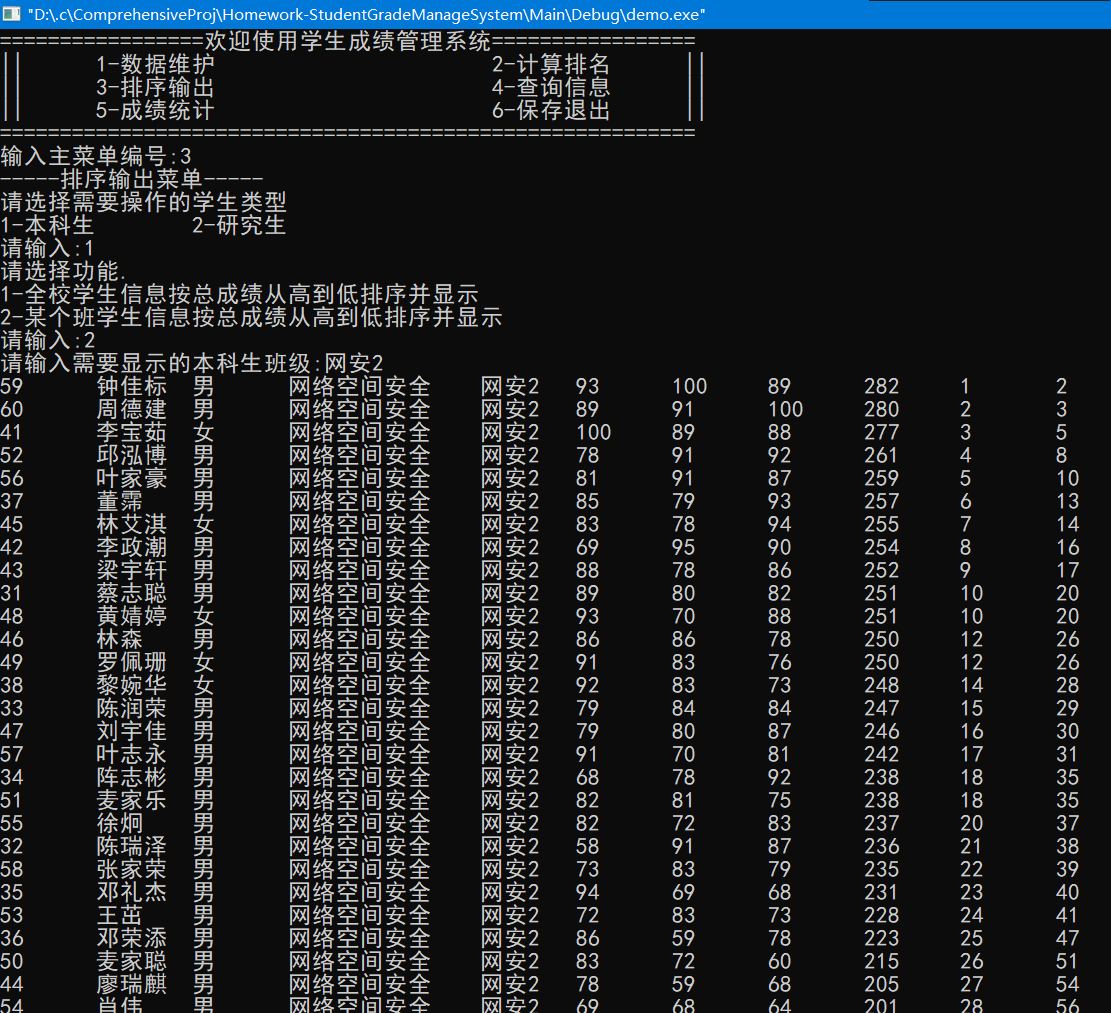


1. **排序输出**

1.输出全校学生信息



2.输出指定班级学生信息



1. **查询信息**
2. 分页输出

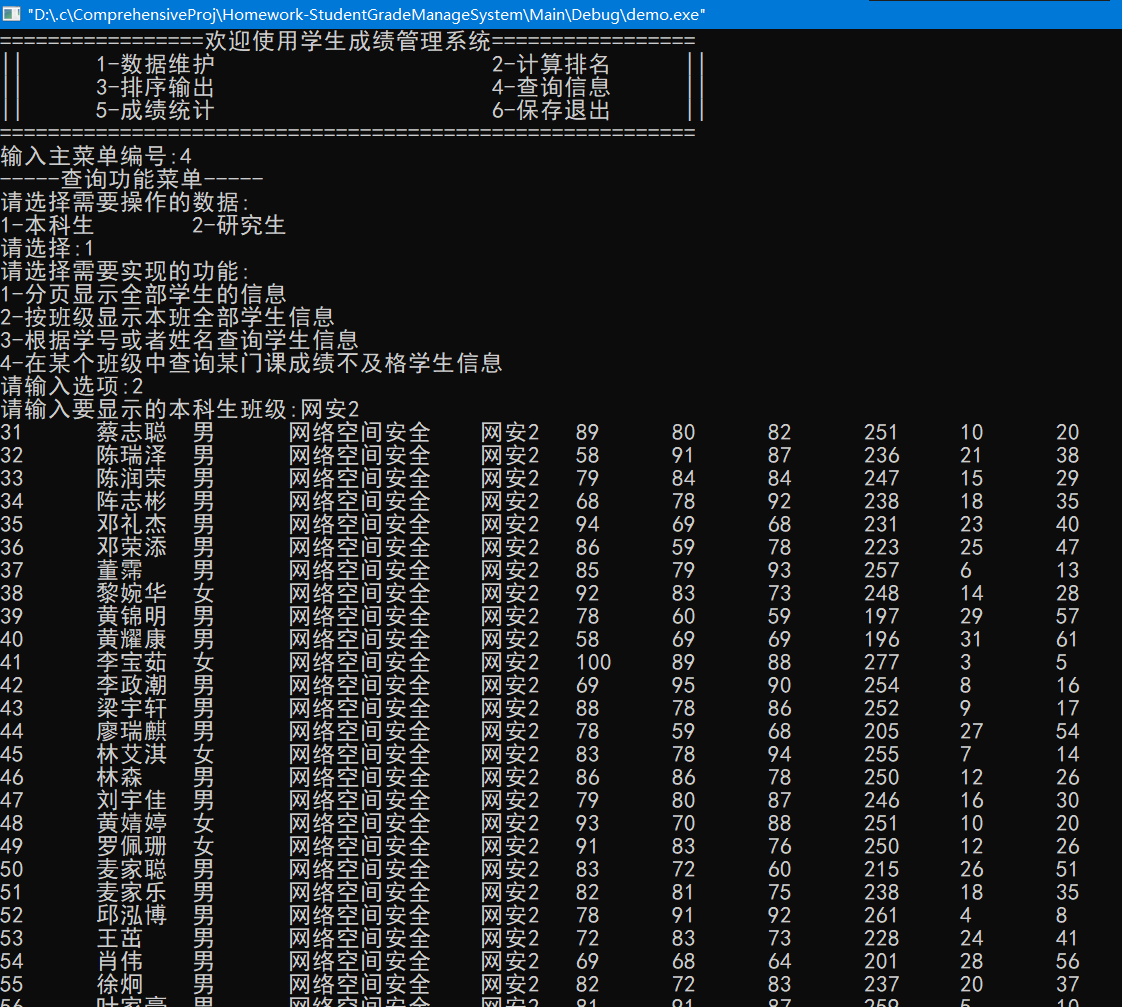
电脑萤幕画面

描述已自动生成图示, 示意图

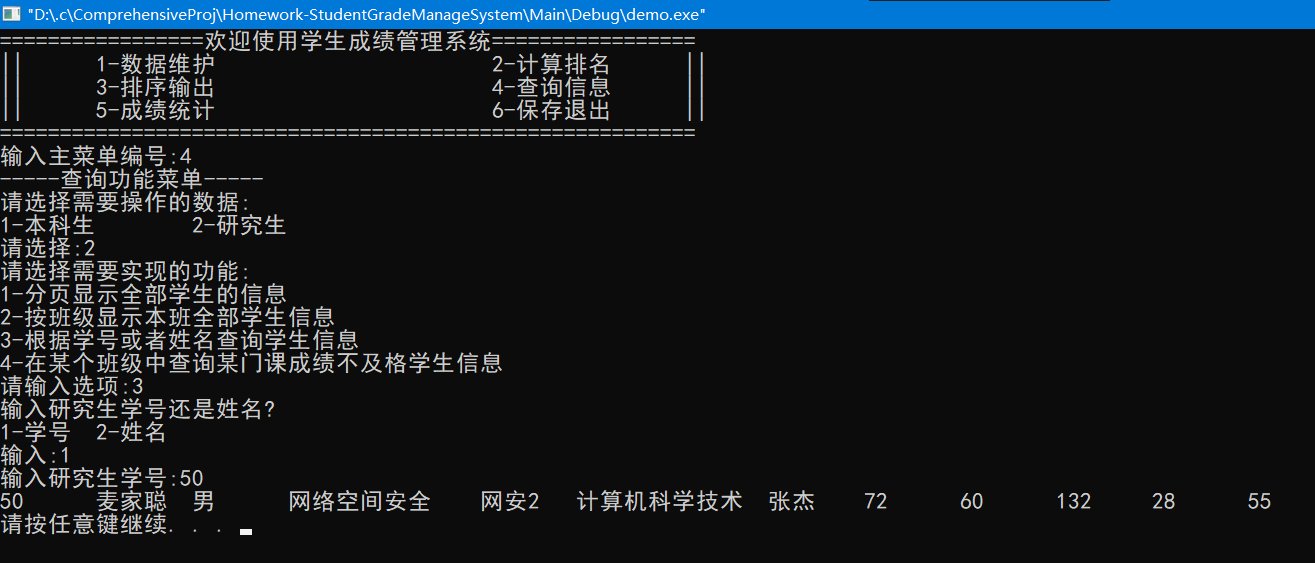
中度可信度描述已自动生成图片包含 图形用户界面

描述已自动生成

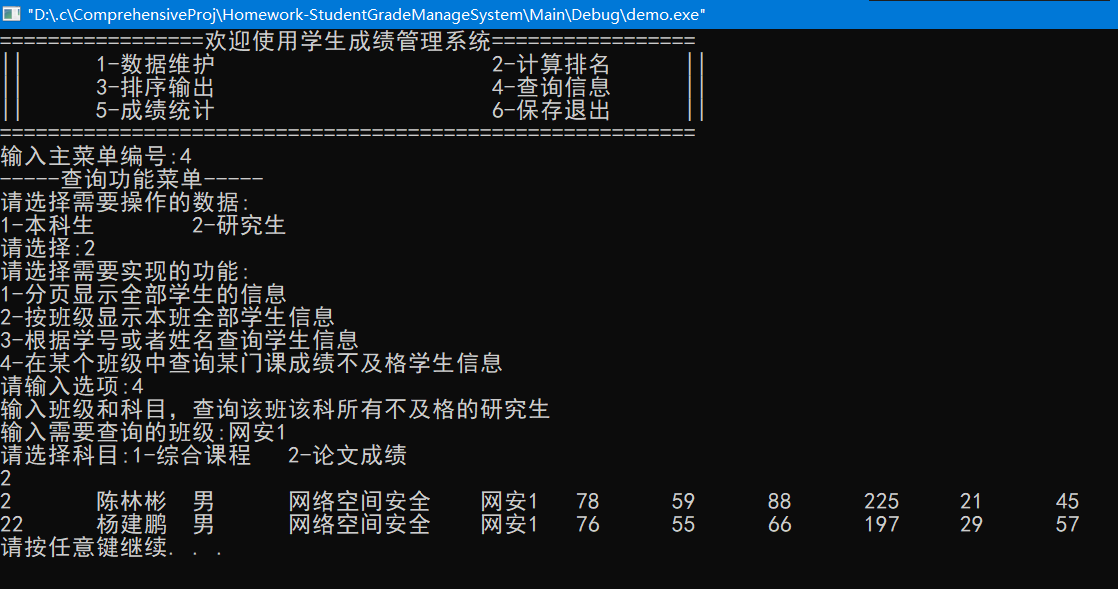
1. 按班级显示



1. 根据学号或姓名显示信息

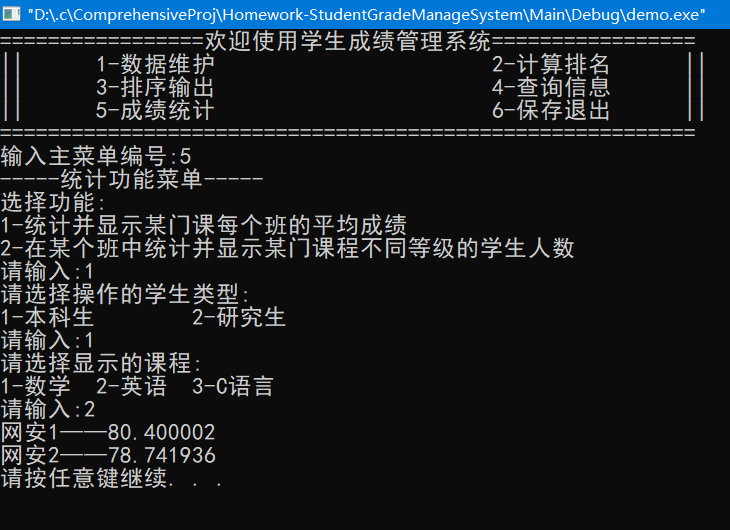


1. 查询某个班某门课不及格学生信息

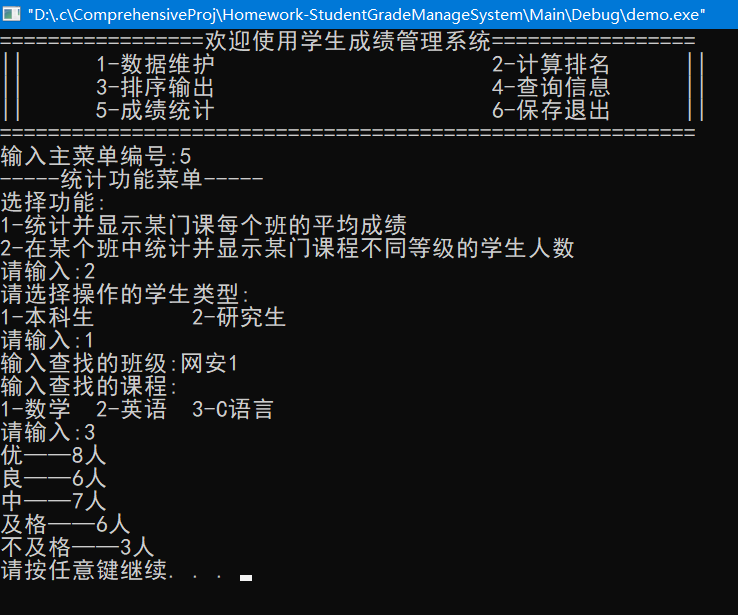


1. 成绩统计

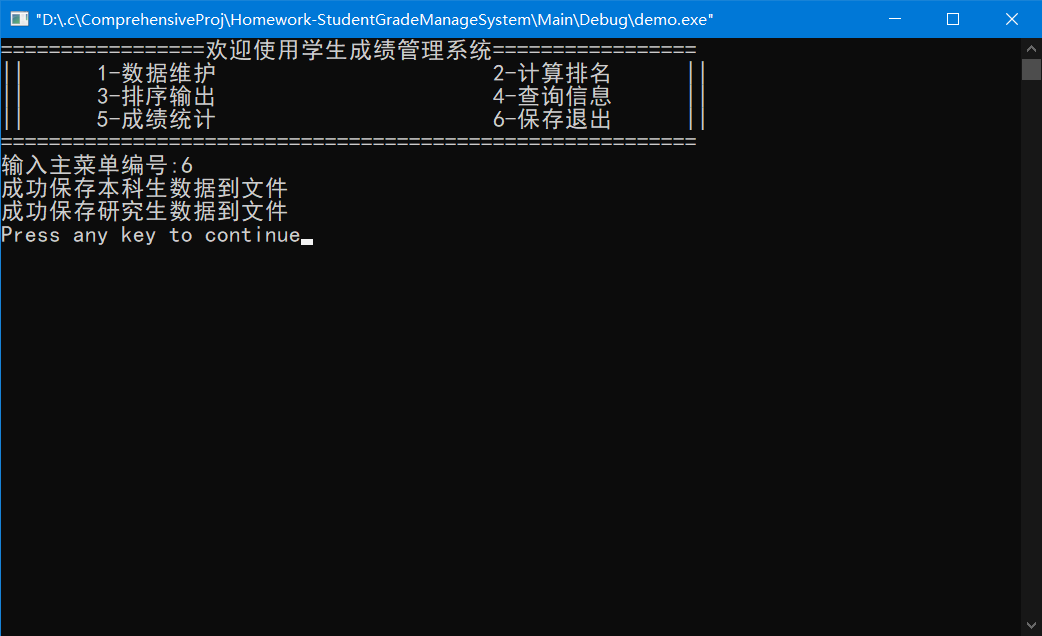
1.计算并显示某门课每个班平均成绩



2.统计某个班并显示某门课不同等级的学生人数



1. 保存并退出



--------------------------------------------------------------------------------

**六、总结**

**要求：**[1]. 必须描述本次综合实践各成员如何组织、配合、协作完成任务的过程及感想。

[2]. 必须按成员依次列出各成员在本次综合实践中所碰到的难点及解决方法、完成后的感想等。

本次作业，组长和成员各司其职，组长主要负责基本资料数据的维护管理模块和成绩数据的维护管理模块；成员主要负责查询数据模块和数据排序模块以及数据统计模块。总的来说，两人扬长避短，组长思维谨密，因此负责数据维护模块，而成员想法多，思维敏捷，因此负责编写查询、排序、统计等功能的函数。

网课期间，我们经常通过微信电话讨论如何写这个大作业，互相指导，各抒己见。

组长：

难点不多，可以通过沟通解决大部分问题，但是一旦某个环节缺乏交流便会出现重复造轮子的现象。

解决办法：多沟通多交流，让思想碰撞出火花。

感想：身为领头人除了执行力，更重要的是统筹修调的能力，与成员充分交流才会有默契的配合。

成员：

碰到的难点：暂时没发现有什么难的，就是代码长，排版不太美观。

对难点的解决方法：由于没遇到难点，故没有解决方法。

感想：一个人的能力是有限的，与人合作能实现1+1大于2的效果。

--------------------------------------------------------------------------------

**七、程序清单附件**

**要求：**必须分成员版面清楚地列出每位成员在本次综合实践中完成的程序清单，注明学号与姓名。

**组长完成的程序清单：**

/\* 自定义函数 \*/

**int** **checkSUGScore**(SUG **\****pSUG*); //检查本科生成绩(数学、英语、C语言)是否有无效成绩（-1），如果有则返回1， 否则返回0

**int** **checkSPGScore**(SPG **\****pSPG*); //检查研究生成绩(综合课程、论文)是否有无效成绩（-1），如果有则返回1， 否则返回0

**void** **printSUGTitle**();//输出本科生基本信息组成部分

**void** **printSPGTitle**();//输出研究生基本信息组成部分

**void** **printSUGBaseData**(SUG **\****pSUG*); //输出一个本科生的基本信息

**void** **printSPGBaseData**(SPG **\****pSPG*); //输出一个研究生的基本信息

**void** **printSUGInfo**(SUG **\****pSUG*); //输出一个本科生的全部信息

**void** **printSPGInfo**(SPG **\****pSPG*); //输出一个研究生的全部信息

**void** **calculateSUGTotalScore**();//计算本科生总成绩

**void** **calculateSPGTotalScore**();//计算研究生总成绩

/\* 自定义函数 \*/

/\* 1.A.基本数据管理函数声明 \*/

**void** **BaseData\_maintenance**();//基本数据维护菜单

**void** **addBaseData**();//添加数据的菜单

/\*\*参数说明：stu为要添加的学生类型， 1-本科生，2-研究生;

 \* quantity为添加模式，1-添加单个，2-添加多个（重复执行单个添加)

 \* \*/

**void** **addStudent**(**int** *stu*, **int** *quantity*);

**void** **modifyBaseData**();//数据修改菜单 用于输入学生类型及学号

**void** **modifyStudent**(**int** *stu*, **int** *id*); //参数说明:stu为学生类型  1-本科生 2-研究生; id:待修改学生的学号

**void** **deleteBaseData**();//删除数据菜单

**void** **deleteStudent**(**int** *stu*, **int** *id*); //删除数据函数:学号相符则释放该节点内存

**void** **deleteUStu**(**int** *id*); //根据学号删除一个本科生（释放节点内存）

**void** **deletePStu**(**int** *id*); //根据学号删除一个研究生（释放节点内存）

**void** **inquireBaseData**();//查询数据菜单

**void** **inquireStu**(**int** *stu*, **int** *id*); //根据接收的学生类型和学号（形参）进行查找并输出基本数据

/\* 1.A.基本数据管理函数声明 \*/

/\* 1.B.成绩管理部分声明 \*/

**void** **PerformanceData\_maintenance**();//成绩数据管理菜单

**void** **inputPerformanceData**();//输入成绩菜单

**void** **inputScore**(**int** *stu*, **int** *quantity*); //参数说明:stu:1-本科生 2-研究生 quantity:1-单个 2-多个

**void** **modifyPerformanceData**();//修改成绩菜单

**void** **modifyScore**(**int** *stu*, **int** *id*); //根据学号修改学生的成绩

**void** **deletePerformanceData**();//删除数据菜单

**void** **deleteScore**(**int** *stu*, **int** *id*); //根据学号删除一个学生成绩数据（相关数据设置成-1） stu:1-本科生 2-研究生

**void** **inquirePerformanceData**();//查询成绩菜单

**void** **inquireScore**(**int** *stu*, **int** *id*); //根据学号查找学生成绩 stu:1-本科生 2-研究生 id:需要查找学生的学号

**void** **calculatePerformanceData**();//计算成绩函数

/\* 1.B.成绩管理部分声明 \*/

/\* 2.计算排名部分声明 \*/

**void** **Calculate\_rank**();//排名功能菜单

**void** **calculateSUGRank**();//计算所有本科生的总成绩和班校排名

**void** **calculateSPGRank**();//计算所有研究生的总成绩和班校排名

/\* 2.计算排名部分声明 \*/

/\* 3.排序输出部分声明 \*/

**void** **Sort\_print**();//3.排序功能菜单

/\* 3.排序输出部分声明 \*/

/\* 4.查询功能部分声明 \*/

**void** **Inquire\_information**();//4.查询功能菜单

/\* 4.查询功能部分声明 \*/

/\* 5.统计功能菜单声明 \*/

**void** **Performance\_statistic**();//5.统计功能菜单

/\* 5.统计功能菜单声明 \*/

**成员1完成的程序清单：**

void classSort(struct underGraduate\* Head, char Class[]);

//[本科生]查询某个班学生信息（按排名先后输出）

void classShow(struct underGraduate\* Head, char Class[]);

//[本科生]查询某个班级学生信息（按学号先后输出）

void classShow(struct underGraduate\* Head,char Class[]);

//[本科生]查询某个班级学生信息（按学号先后输出）

void schoolSort(struct underGraduate\* Head);

//[本科生]输出所有学生信息（按排名先后输出）

void showAll(struct underGraduate\* Head);

//[本科生]分页输出所有学生信息（按学号）

void showOne1(struct underGraduate\* Head,int o);

//[本科生]按学号查询学生信息

void showOne2(struct underGraduate\* Head,char name[]);

//[本科生]按姓名输出学生信息

void showClassPass(struct underGraduate\* Head, char name[],int i);

//[本科生]查询某个班级某个科目不及格的学生信息

void countAverage(struct underGraduate\* Head, int i);

//[本科生]统计某门科目每个班的平均分并显示

void classCourse(struct underGraduate\* Head, char class[], int i);

//[本科生]统计并显示某个班某门科目不同等级的学生人数

void postClassSort(struct postGraduate\* Head, char Class[]);

//[研究生]查询某个班学生信息（按排名先后输出）

void postClassShow(struct postGraduate\* Head, char Class[]);

//[研究生]查询某个班级学生信息（按学号先后输出）

void postSchoolSort(struct postGraduate\* Head);

//[研究生]输出所有学生信息（按排名先后输出）

void postShowAll(struct postGraduate\* Head);

//[研究生]分页输出所有学生信息（按学号）

void postShowOne1(struct postGraduate\* Head, int o);

//[研究生]按学号查询学生信息

void postShowOne2(struct postGraduate\* Head, char name[]);

//[研究生]按姓名输出学生信息

void postShowClassPass(struct postGraduate\* Head, char class[], int i);

//[研究生]查询某个班级某个科目不及格的学生信息

void postCountAverage(struct postGraduate\* Head, int i);

//[研究生]统计某门科目每个班的平均分并显示

void postClassCourse(struct postGraduate\* Head, char class[], int i);

//[研究生]统计并显示某个班某门科目不同等级的学生人数\*/

**成绩评定表**

|  |  |  |  |
| --- | --- | --- | --- |
| 观测点 | 成绩 | | |
| 组长 | 成员1 | 成员2 |
| 学号： 姓名： | 学号： 姓名： | 学号： 姓名： |
| 解决方案、模块设计与结构化编程的完成情况  （权重0.80） |  |  |  |
| 团队组织、配合与协作情况  （权重0.20） |  |  |  |
| 总分 |  |  |  |