

# 重庆师范大学

## 《数据结构课程设计》

课程名称： 数据结构

题 目： 学生成绩信息管理系统

学 院： 计算机与信息科学学院

专业年级： 计算机科学与技术 2019 级

小组组长： 赵宇湘

小组成员： 谢鑫、张益

指导教师： 张万里 职称： 讲师

2022 年 1 月 5 日

## 目录

一、设计目的/需求分析.....	- 1 -
二、问题描述.....	- 1 -
三、基本要求.....	- 1 -
四、概要设计.....	- 2 -
1、算法思路.....	- 2 -
2、工作分配.....	- 2 -
3、程序模块.....	- 2 -
五、主程序.....	- 7 -
六、运行截图.....	- 11 -
七、总结.....	- 23 -

# 学生成绩信息管理系统

## 一、设计目的/需求分析

设计目的：

设计一个学生成绩信息管理系统，可以方便地对学生和学生成绩进行有效管理，提高工作效率。

需求分析：

本系统根据学生成绩信息管理的需求，而建立一个“学生成绩信息管理系统”，以方便对成绩的各项管理操作。本系统能对学生成绩信息进行输入和输出；能添加学生成绩信息；能根据学生的学号或姓名来查询和修改该学生的成绩信息；能按学号或姓名进行排序，并显示学生信息、成绩等；能准确计算出学生课程成绩的平均分和及格率。设置合理的数据结构，每个学生记录包含的信息要有学号、姓名和各门课程的成绩。

## 二、问题描述

设计一个学生成绩信息管理系统，要求设计合理，美观，并实现以下功能：

- (1) 从键盘输入学生的学号、姓名、课程成绩，输入下一个数据时会出现提示继续输入还是退出。
- (2) 可以插入学生成绩信息。
- (3) 可以按学号或姓名查找学生成绩信息，并对该信息进行删除或修改。
- (4) 可以按学号大小或姓名首字母对学生信息进行排序。
- (5) 可以输出所有学生的成绩信息。
- (6) 可以分别求出课程的平均分和及格率。

## 三、基本要求

(1) 学生成绩信息管理系统包括：学生学号、姓名、各科课程的成绩。能实现成绩信息的输入，成绩信息的插入、查找、删除、修改、排序、输出，以及求出每个学生成绩的平均分和及格率。

(2) 系统能实现的操作功能如下：

- ①成绩信息输入/插入功能：将学生的学号、姓名、成绩录入。
- ②成绩信息修查找能：该功能主要是对某个学生进行查找，考虑到可能存在同名

同姓的学生，故该功能可设两个辅助查找功能——即按学号查找以及按姓名查找。

③成绩信息删除和修改功能：此功能主要是能通过学号或姓名查找到某个学生修改或者删除该学生的信息。

④成绩信息输出：该功能主要是能将所有学生的信息都显示出来。

⑤成绩信息排序：该功能主要是对全部学生进行排序，该功能可设两个辅助排序功能——即按学号排序以及按姓名首字母排序。

⑥求课程平均成绩：该功能主要是求出所有学生每门课程的平均成绩。

⑦求课程成绩及格率：该功能主要是统计每门课程成绩的及格率，大于 60 分的为及格，小于 60 分的为不及格，最后算出及格率。

## 四、概要设计

### 1、算法思路

用 C++ 进行编写，采用线性链表结构，带头结点的单链表存储结构，定义节点类型 `ListNode`，每个节点包括数据域 `data` 和指针域 `next`。定义头指针 `Link List`。用尾插法建立带头节点的管理系统链表算法，用线性查找法遍历查找下一个节点，用冒泡排序法实现升序排序，还有包括整个框架的构思，图等的运用，最终构成了整个程序。

### 2、工作分配

- （1）框架设计：由张益进行构思，同时参考其他人的意见。
- （2）系统设计：由赵宇湘、谢鑫进行框架的编写。
- （3）程序设计：由赵宇湘进行各项功能的编辑。
- （4）程序调试：由赵宇湘进行程序的调试与调整。
- （5）文档制作：由赵宇湘完成文档的编辑工作。
- （6）PPT 制作：由张益完成 PPT 的编辑工作。
- （7）答辩视频制作：由谢鑫完成答辩视频的录制工作。

### 3、程序模块

“学生成绩信息管理系统”包括 9 个功能模块，各个功能模块的总体设计图如下：

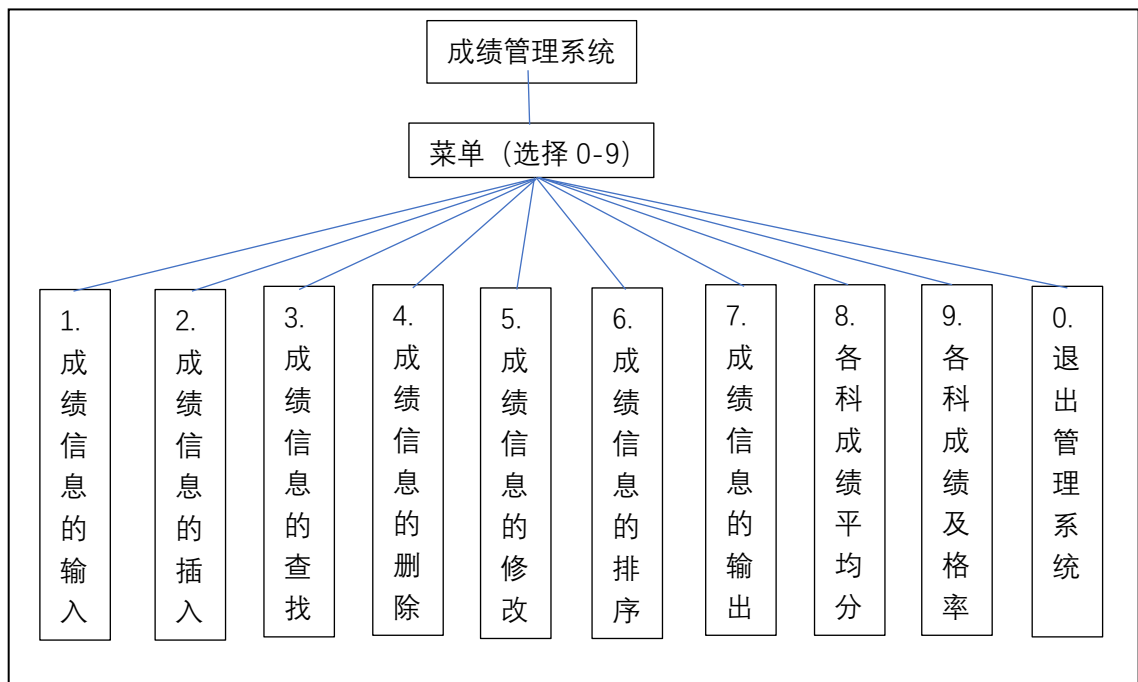


图 1 功能模块总体设计图

成绩信息的输入功能：用 CreateList 函数建立管理系统的链表；

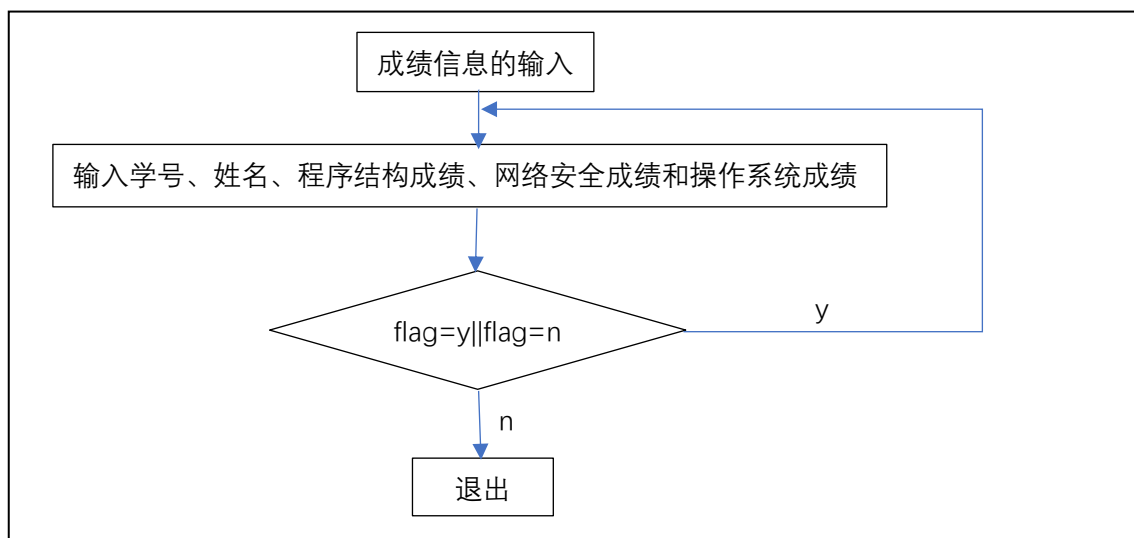


图 2 成绩信息输入算法流程图

成绩信息的插入功能：用 InsertNode 函数实现管理系统信息的插入；

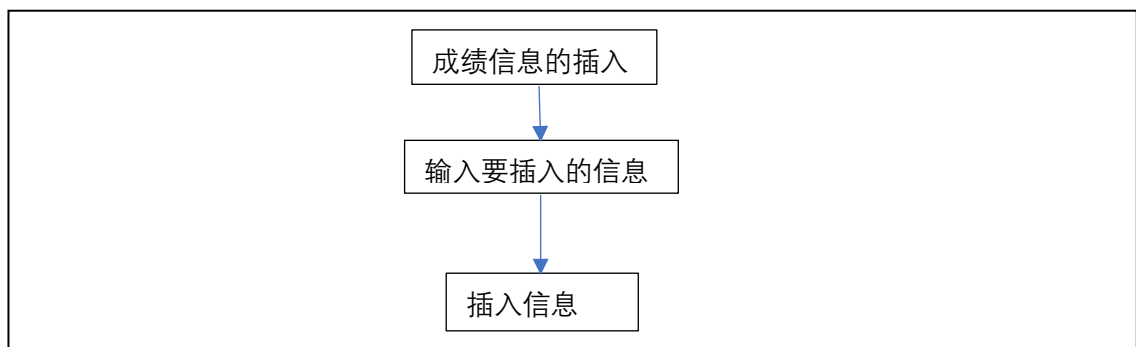


图 3 成绩信息插入算法流程图

成绩信息的查找功能：用 ListFind 函数实现管理系统信息的查找；

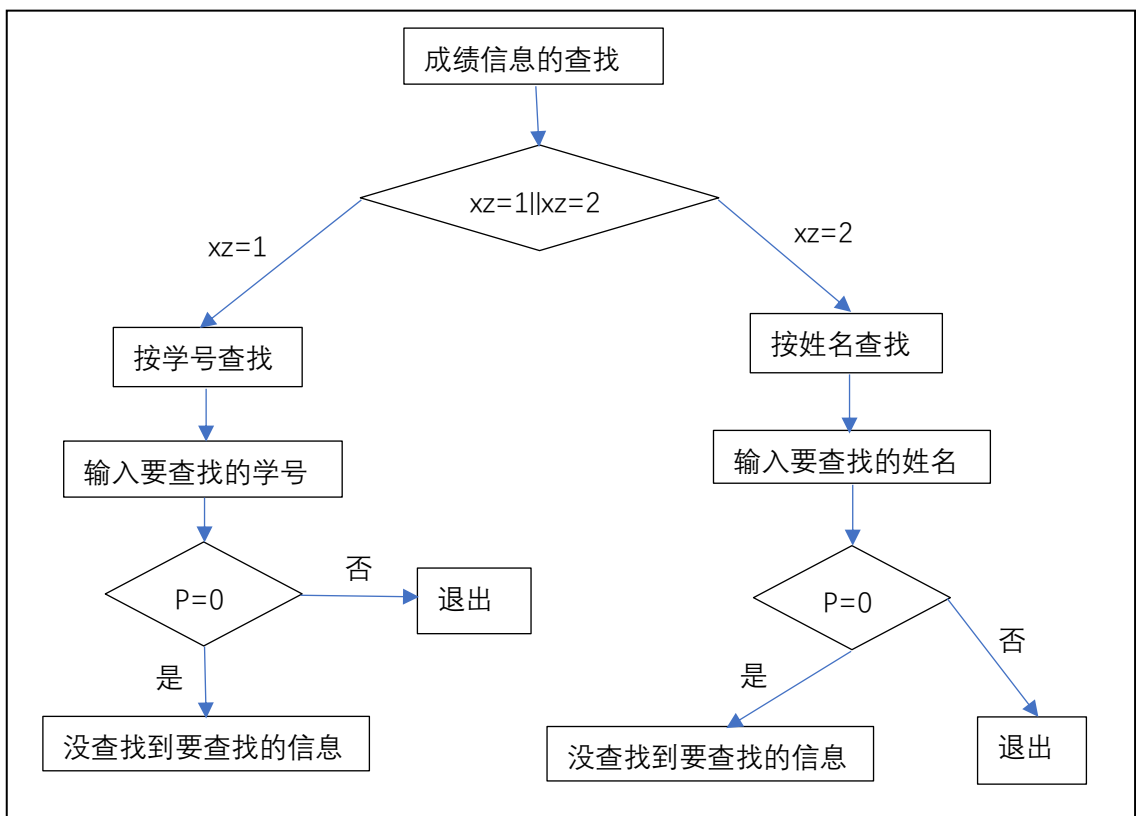


图 4 成绩信息查找算法流程图

成绩信息的删除功能：用 DelNode 函数实现管理系统多余信息的删除；

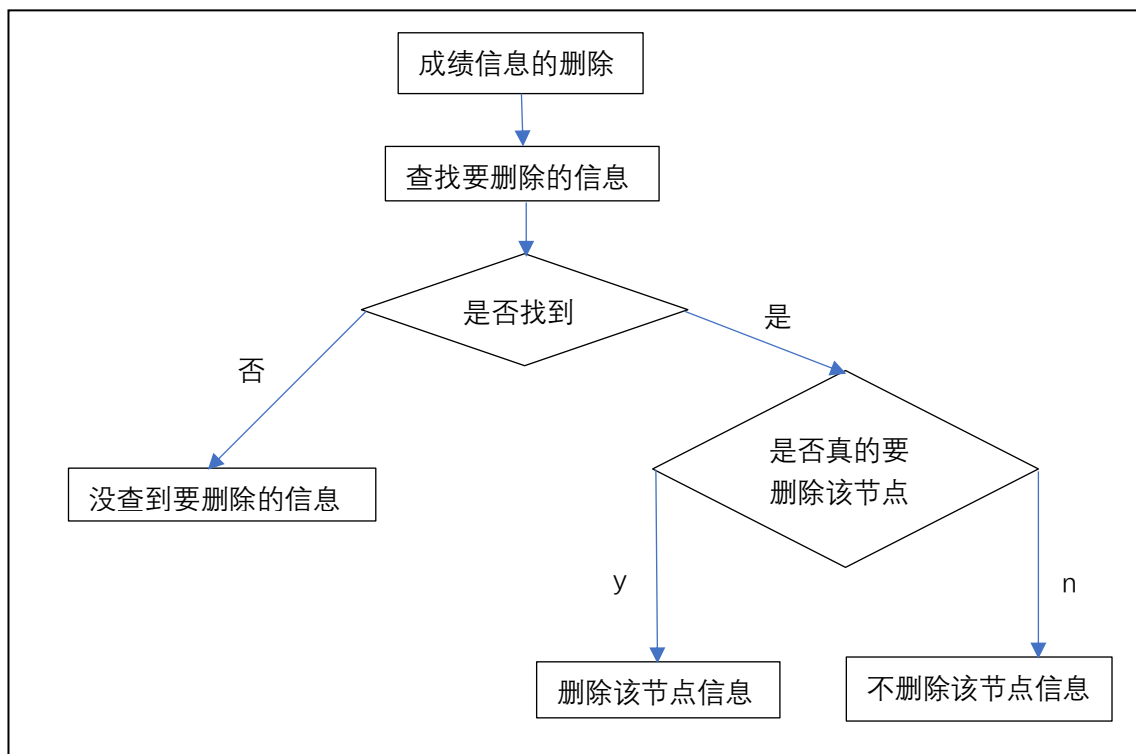


图 5 成绩信息删除算法流程图

成绩信息的修改功能：用 ChangeNode 函数实现管理系统信息的修改；

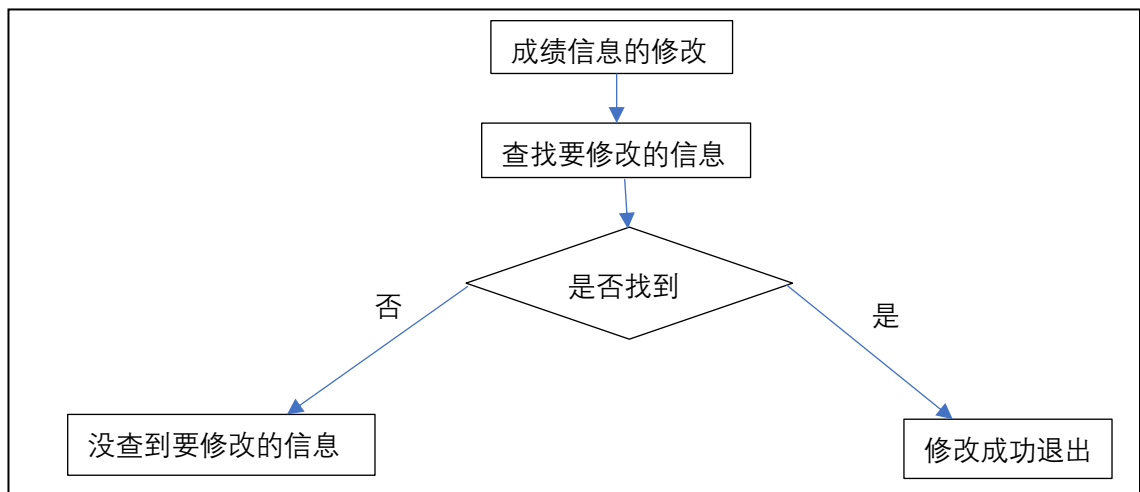


图 6 成绩信息修改算法流程图

成绩信息的排序功能：用 Sorting\_order 函数实现管理系统信息的排序；

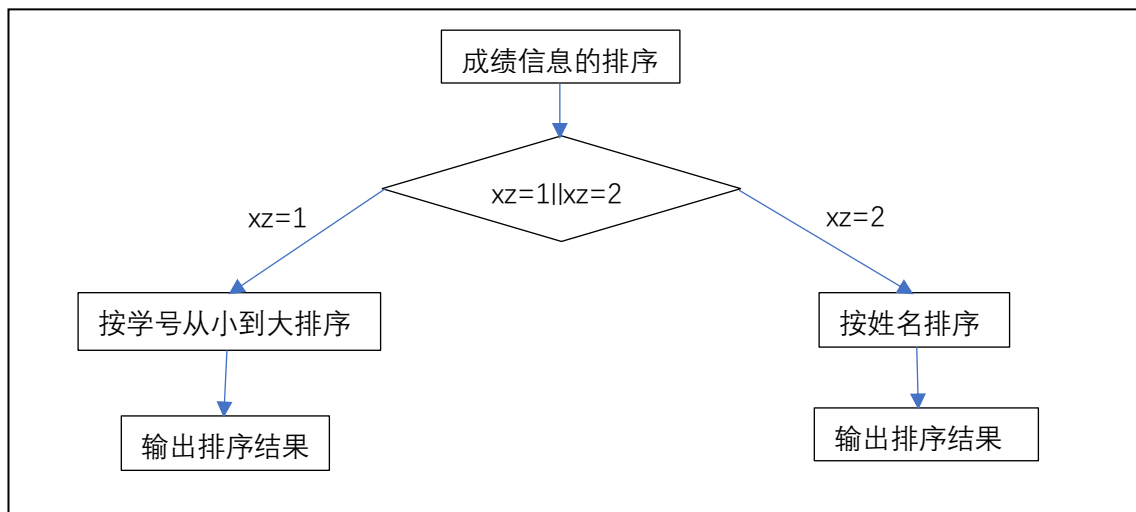


图 7 成绩信息排序算法流程图

成绩信息的输出功能：用 PrintList 函数输出链表信息；

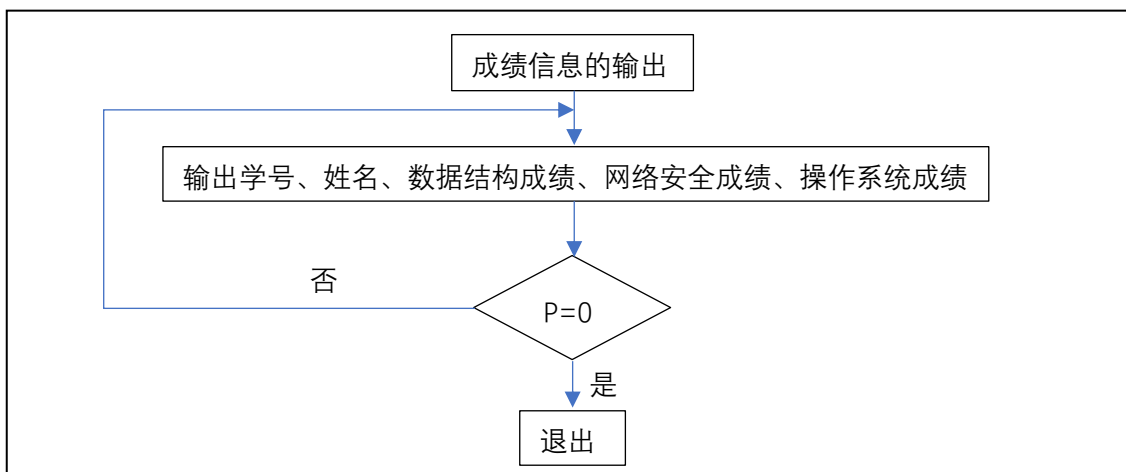


图 8 成绩信息输出算法流程图



求各科成绩平均分功能：用 `average` 函数实现管理系统中各科成绩求平均值；

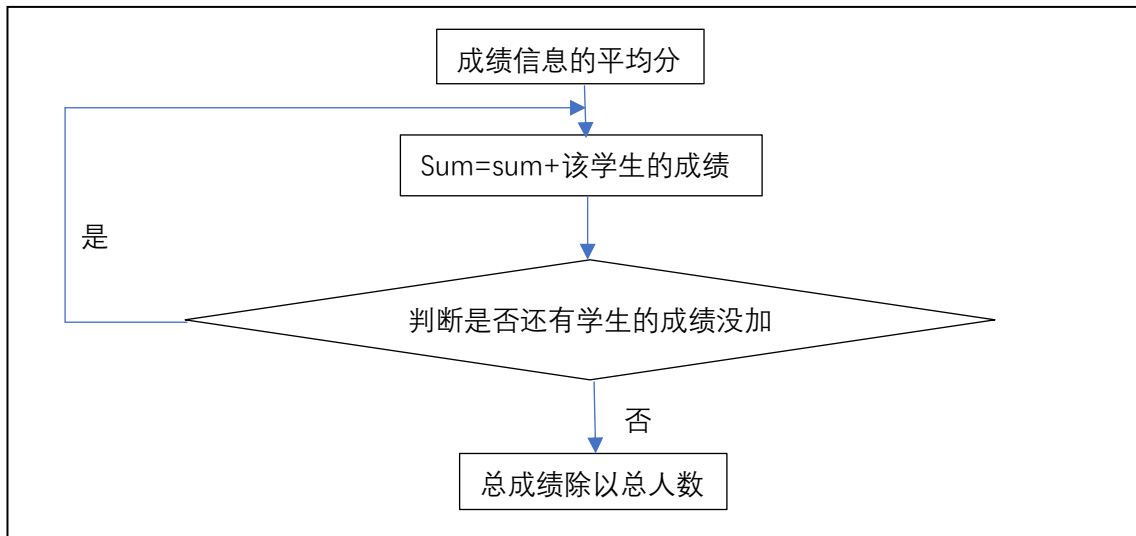


图 9 求平均成绩算法流程图

求各科成绩及格率功能：用 `pass_rate` 函数实现管理系统中各科成绩的通过率；

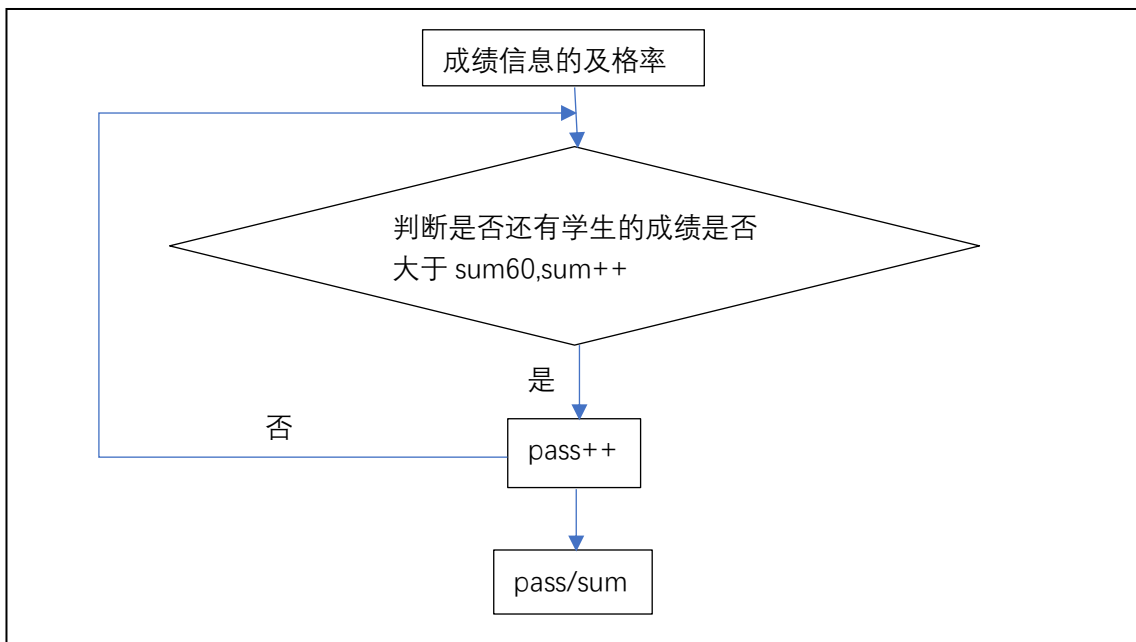


图 10 求成绩及格率算法流程图

## 五、主程序

程序 1. 学生的成绩信息的链表存储结构

```

typedef struct{      //管理系统节点类型
    char num[10];    //学号

```

```

    char name[20]; //姓名
    float score[3]; //各科成绩
    float average; //平均成绩
    float pass_rate; //成绩及格率
}DataType;
typedef struct node{ //节点类型定义
    DataType data; //节点数据域
    struct node *next; //节点指针域
}ListNode;
typedef ListNode *LinkList;
LinkList head;

```

程序 2. 用尾插法建立带头节点的管理系统链表算法

//用尾插法建立管理系统链表函数

LinkList CreateList(void)

```

{ //尾插法建立带头节点的管理系统链表算法
    LinkList head=(ListNode *)malloc(sizeof(ListNode)); //申请头节点
    ListNode *p,*rear;
    int flag='y'; //是否结束标志
    rear=head; //尾指针初始指向头节点
    while(flag=='y')
    {p=(ListNode *)malloc(sizeof(ListNode)); //申请新节点
        printf("请顺序输入学号、姓名、数据结构成绩、网络安全成绩和操作系统成绩\n");
        printf("-----\n");

        scanf("%s%s%f%f%f",p->data.num,p->data.name,&p->data.score[0],&p->data.score[1],&p->data.score[2]);

        rear->next=p; //新节点连接到尾节点之后
        rear=p; //尾指针指向新节点
    }
}

```

```

    printf("继续输入吗?(y/n):");
    getchar();
    scanf("%c",&flag); //读入一个标志数据
}

rear->next=NULL;    //终端节点指针域置空
return head;        //返回链表头指针
}

```

程序 3. 用线性查找法遍历查找下一个节点

```

ListNode *ListFind(LinkList head)
{    //有序链表上的查找
    ListNode *p;
    char num[10];
    char name[20];
    int xz;
    printf("=====\n");
    printf(" 1.按学号查询  \n");
    printf(" 2.按姓名查询  \n");
    printf("=====\n");
    printf(" 请选择: ");
    p=head->next;                //假定学生成绩表带头节点
    scanf("%d",&xz);
    if(xz==1){                  //按学号查找
        printf("请输入要查找者的学号: ");
        scanf("%s",&num);
        getchar();
        while(p&&strcmp(p->data.num,num)<0)
            p=p->next;
        if(p==NULL||strcmp(p->data.num,num)>0)
            p=NULL;
    }
}

```

```

    }
    else if(xz==2)                //按姓名查找
    {
        printf("请输入要查找者的姓名: ");
        scanf("%s",&name);
        while(p-&&strcmp(p->data.name,name)!=0)
            p=p->next;
    }
    return p;
}

```

程序 4. 用冒泡排序法实现升序排序

/管理系统的排序函数

```

ListNode *Sorting_order(LinkList head){
    ListNode *p,*q;
    int xz;
    printf("=====\\n");
    printf("1.按学号排序 \\n");
    printf("2.按姓名排序 \\n");
    printf("=====\\n");
    printf("    请选择: ");
    scanf("%d",&xz);
    if(xz==1){                //按学号排序
        p=head->next->next;
        head->next->next=NULL;
        while(p){
            q=p->next;
            InsertNode(head,p);
            p=q;
        }
    }
}

```

```

        return head;
    }
    if(xz==2){          //按姓名排序
        p=head->next->next;
        head->next->next=NULL;
        while(p){
            ListNode *p1,*p2;
            p1=head;
            p2=p1->next;
            q=p->next;
            while(p2 && strcmp(p2->data.name,p->data.name)<0)
            {
                p1=p2;
                p2=p2->next;
            }
            p1->next=p;
            p->next=p2;
            p=q;
        }
        return head;
    }
}

```

## 六、运行截图

1、利用尾插法建立成绩管理系统链表函数，实现成绩信息的输入：输入选项 1，分别输入学生的学号、姓名、数据结构成绩、网络安全成绩、操作系统成绩。

```

C:\Users\86157\Desktop\学生成绩信息管理系统.exe
学生成绩信息管理系统
=====
1. 成绩信息的输入
2. 成绩信息的插入
3. 成绩信息的查找
4. 成绩信息的删除
5. 成绩信息的修改
6. 成绩信息的排序
7. 成绩信息的输出
8. 成绩信息的平均分
9. 成绩信息的及格率
0. 退出管理系统
=====
请 选 择 0-9: 1
*****
*          成  绩  信  息  的  输  入          *
*****
请顺序输入学号、姓名、数据结构成绩、网络安全成绩和操作系统成绩
-----
3025 赵宇湘 80 90 78
继续输入吗?(y/n):y
请顺序输入学号、姓名、数据结构成绩、网络安全成绩和操作系统成绩
-----
3024 张益 90 89 67
继续输入吗?(y/n):y
请顺序输入学号、姓名、数据结构成绩、网络安全成绩和操作系统成绩
-----
3018 谢鑫 77 88 99
继续输入吗?(y/n):n
学生成绩信息管理系统

```

图 1 成绩信息输入数据图

检查输入的信息是否已经录入，输入选项 7，输出每个学生的信息：

```

学生成绩信息管理系统
=====
1. 成绩信息的输入
2. 成绩信息的插入
3. 成绩信息的查找
4. 成绩信息的删除
5. 成绩信息的修改
6. 成绩信息的排序
7. 成绩信息的输出
8. 成绩信息的平均分
9. 成绩信息的及格率
0. 退出管理系统
=====
请 选 择 0-9: 7
*****
*          成  绩  信  息  的  输  出          *
*****
学号      姓名      数据结构      网络安全      操作系统
-----
3025      赵宇湘      80.0          90.0          78.0
-----
3024      张益        90.0          89.0          67.0
-----
3018      谢鑫        77.0          88.0          99.0
-----

```

图 2 查看成绩信息输入情况图

2、实现学生成绩信息的插入：输入选项 2，插入一条学生信息。



图 3 成绩信息插入数据图

检查插入的信息是否已经录入，输入选项 7，输出每个学生的信息：

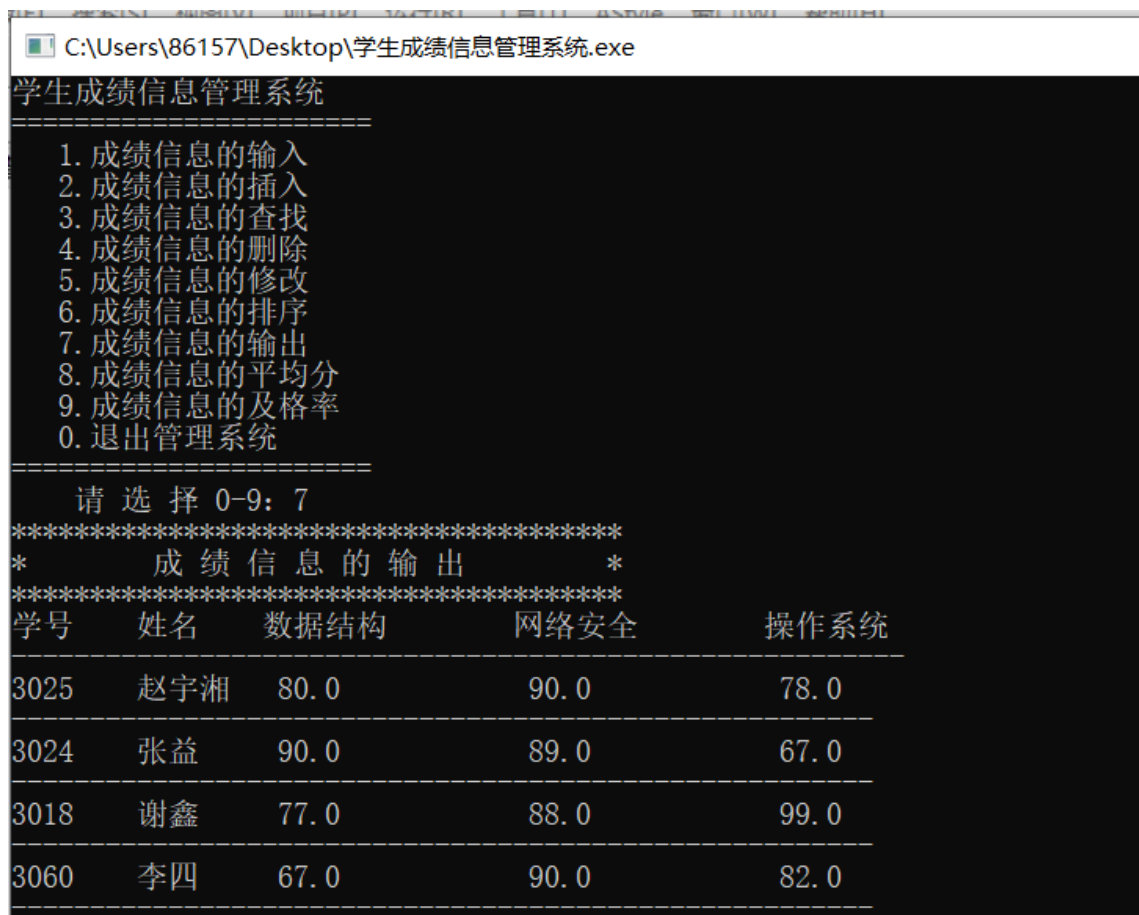


图 4 查看成绩信息插入情况图

3、实现学生成绩信息的查找：输入选项 3，有两种方式查找学生信息，输入不存在的学号或姓名，提示没找到；输入已存在的学号或姓名，输出查找结果。

(1) 按学号查询：



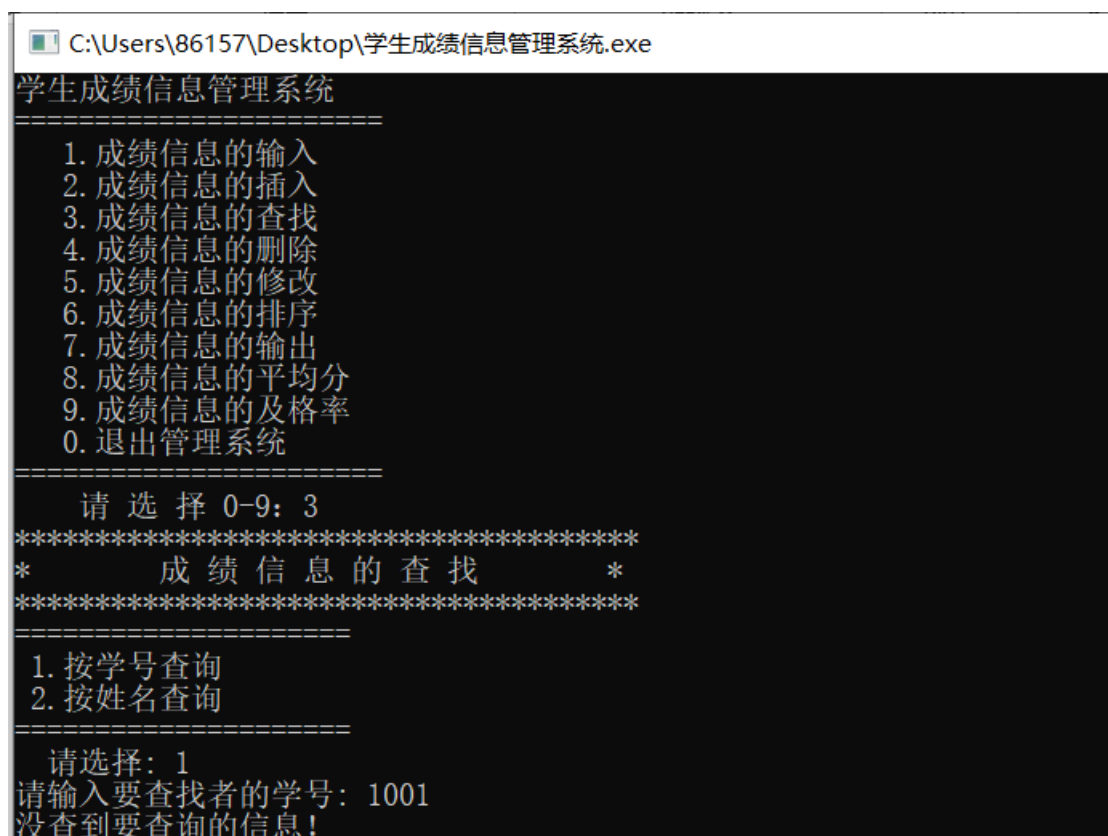


图 5 未查找到成绩信息提示图



图 6 按学号查找图

(2) 按姓名查询:



图 7 按姓名查找图

4、实现学生成绩信息的删除：输入选项 4，有按学号和按姓名两种方式找到要删除的学生信息，若找到所要查找的信息，则提示“真的要删除该节点吗？”，且删除成功后，提示“该学生的信息已被删除！”。若没找到要查询的信息，则提示“没有查到要删除的信息!”。

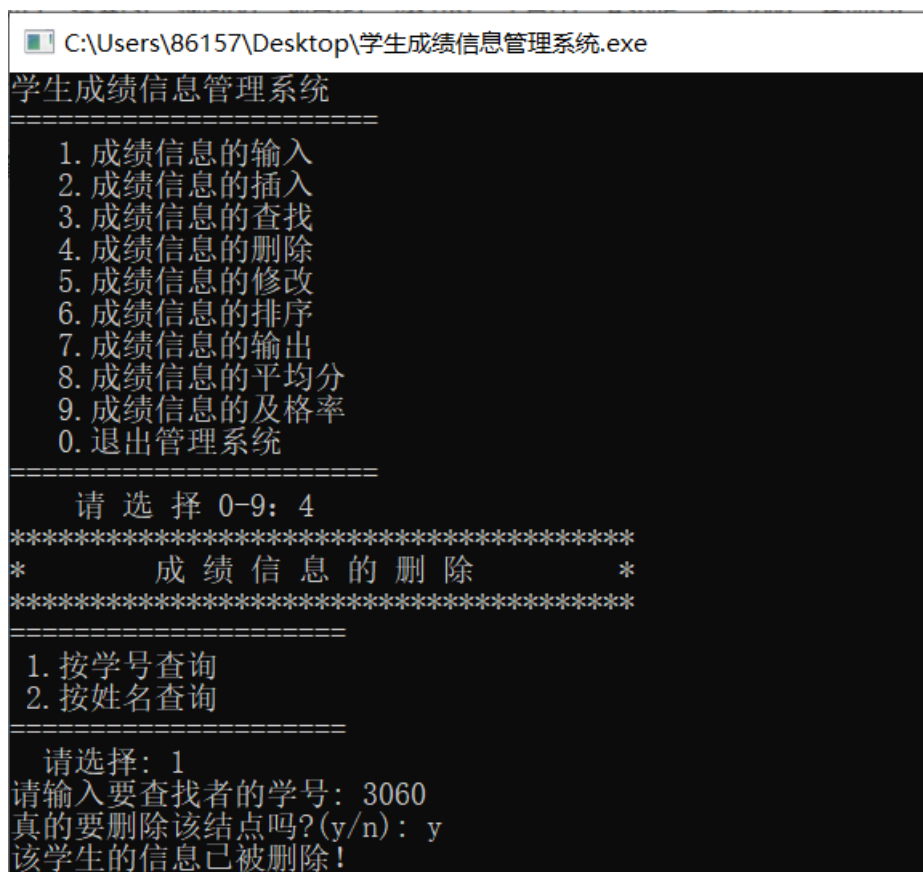


图 8 删除学生信息图



图 9 未找到删除信息提示图

检查信息是否已经删除，输入选项 7，输出每个学生的信息：

```

=====
学生成绩信息管理系统
=====
1. 成绩信息的输入
2. 成绩信息的插入
3. 成绩信息的查找
4. 成绩信息的删除
5. 成绩信息的修改
6. 成绩信息的排序
7. 成绩信息的输出
8. 成绩信息的平均分
9. 成绩信息的及格率
0. 退出管理系统
=====
请 选 择 0-9: 7
*****
*          成  绩  信  息  的  输  出          *
*****
学号    姓名    数据结构    网络安全    操作系统
-----
3025    赵宇湘    80.0        90.0        78.0
-----
3024    张益        90.0        89.0        67.0
-----
3018    谢鑫        77.0        88.0        99.0
-----

```

图 10 查看删除情况图

5、实现学生成绩信息的修改：输入选项 5，按两种方式其中一种找到要修改的学生成绩信息，依次输入修改后的成绩。



图 11 修改学生信息图

检查成绩是否已经修改成功，输入选项 7，查看修改后的成绩。

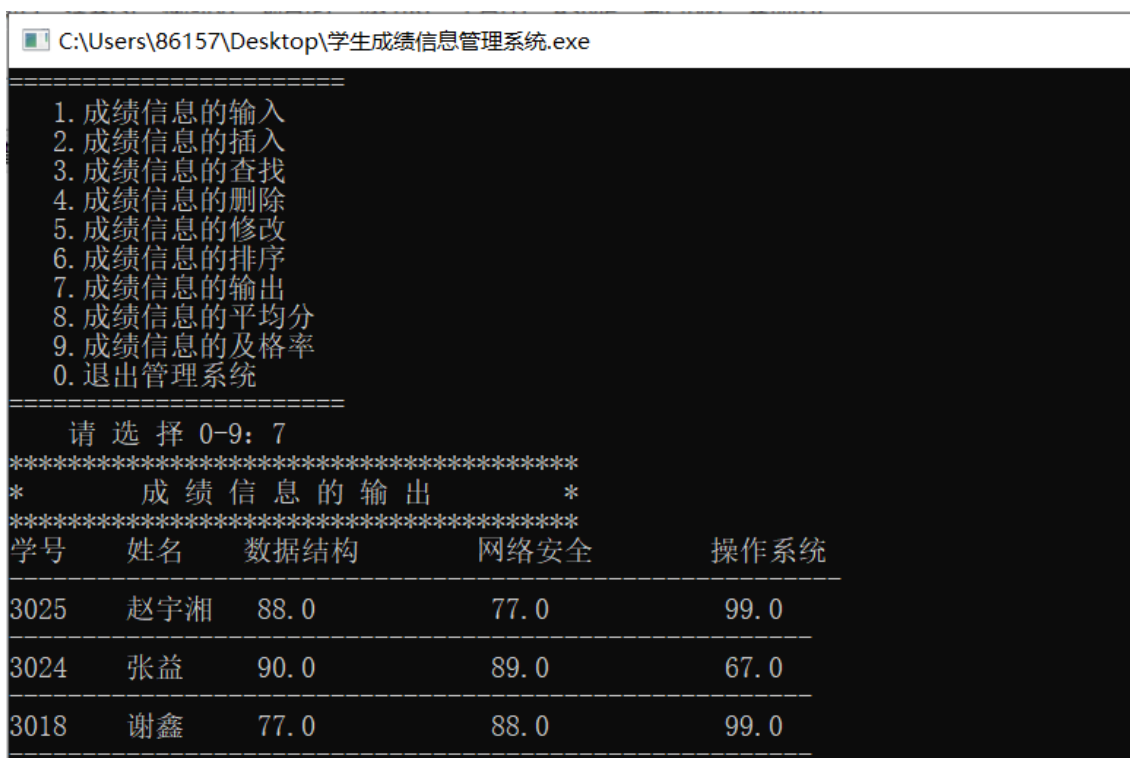


图 12 查看修改情况图

6、实现学生成绩信息的排序：输入选项 6，有两种排序方式。

(1) 按学号排序：

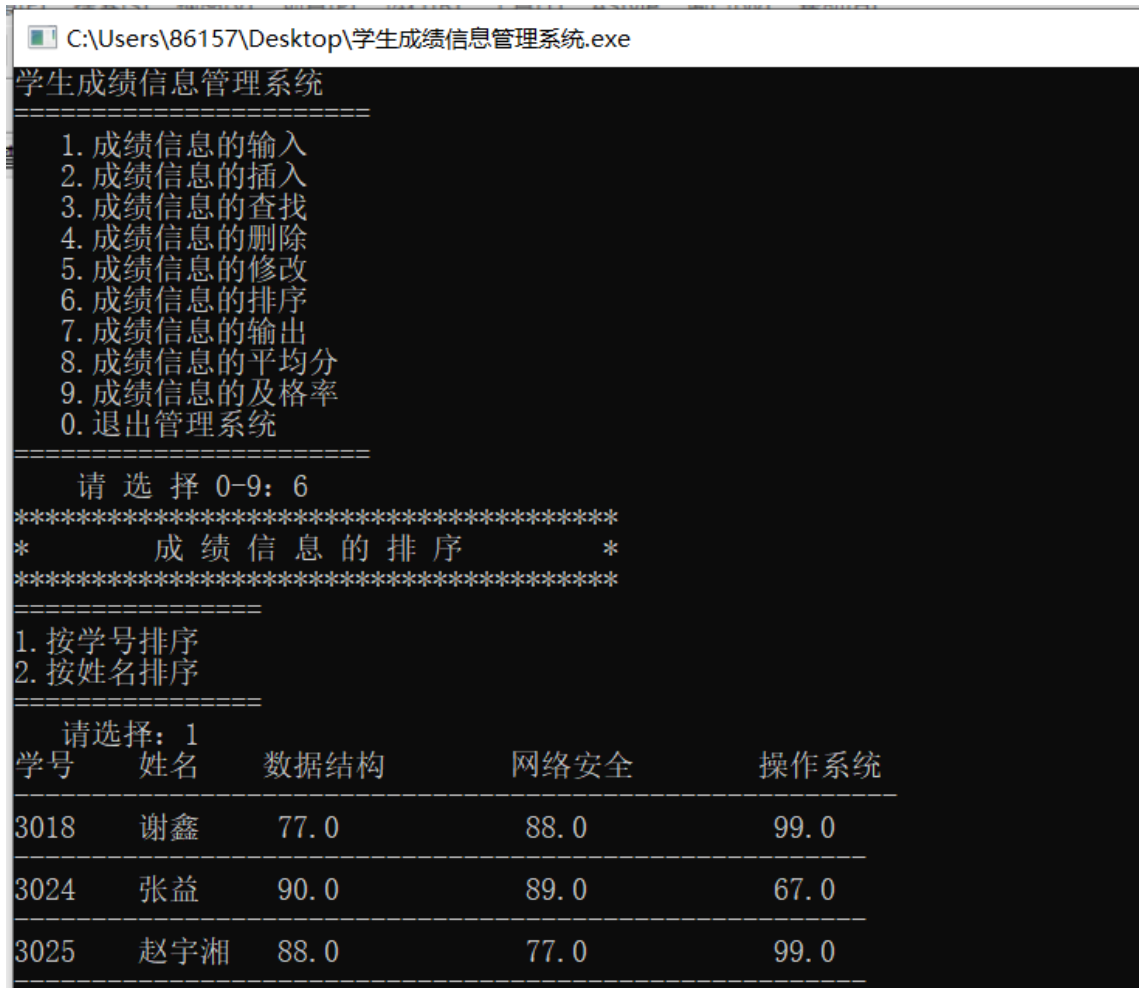


图 13 按学号排序图

(2) 按姓名排序：

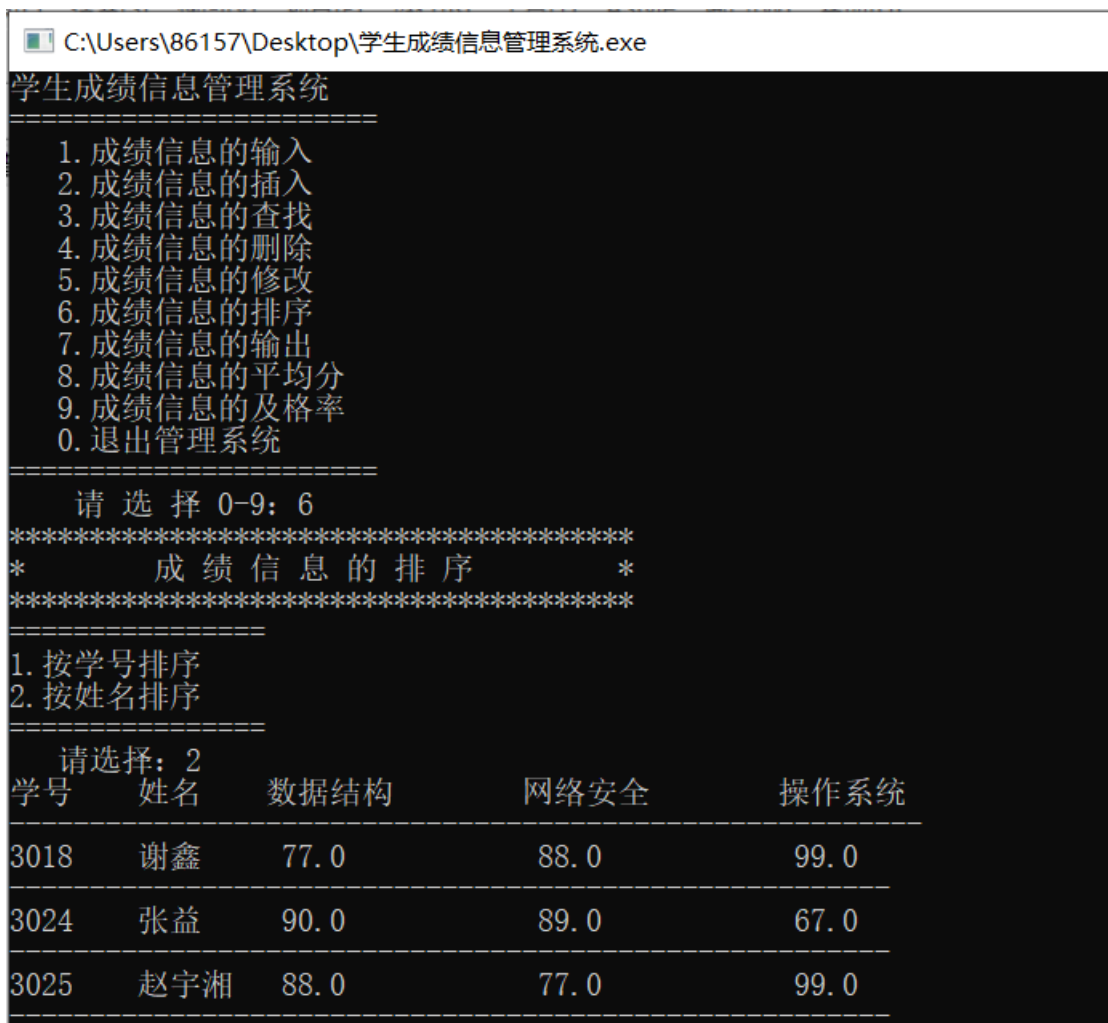


图 14 按姓名排序图

7、实现学生成绩信息的输出：输入选项 7，输出每个学生的信息。

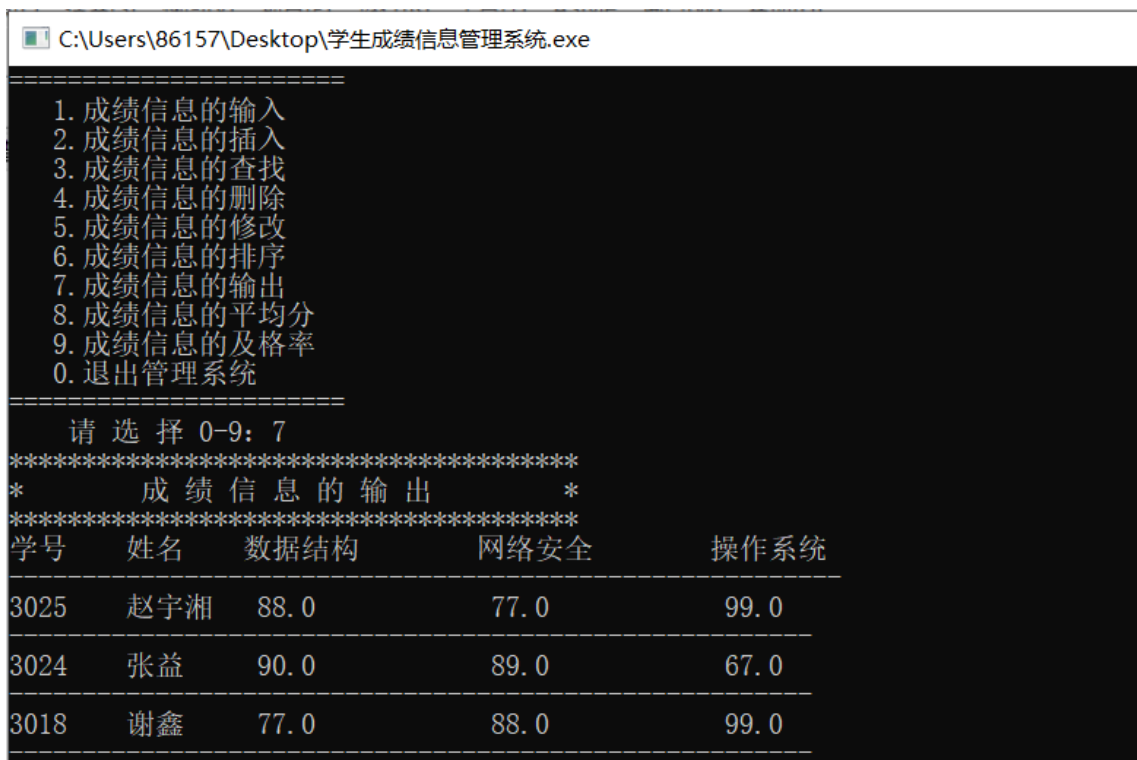


图 15 成绩信息输出图

8、实现求学生成绩的平均分：输入选项 8，查看各科成绩的平均分。

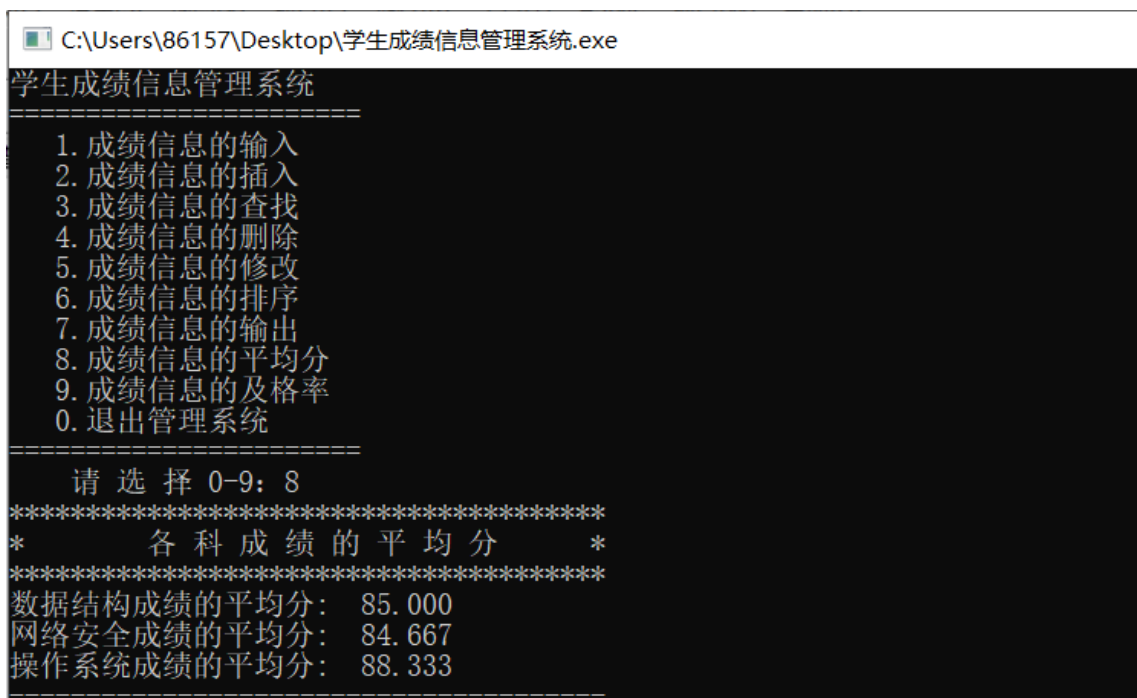


图 16 计算平均分图

9、实现求学生成绩的及格率：输入选项 9，查看各科成绩的及格率。





图 17 计算及格率图

10、退出系统：输入选项 0，退出系统。

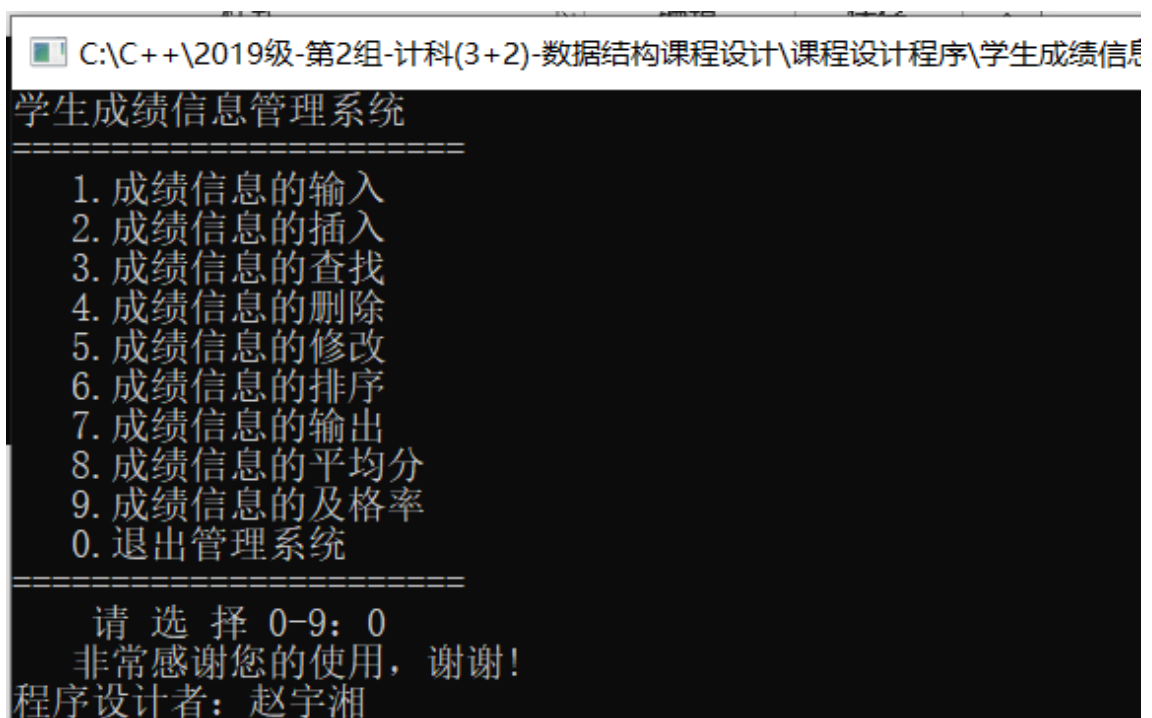


图 18 退出系统图

## 七、总结

本程序在刚开始调试时有许多错误，但在我们的努力及同学的帮助下都被一

一克服，现在操作本程序时可根据提示进行相关操作，能正确输出结果。在刚开始的几次调试中曾经出现过不能运行、不能产生十以内随机数字、不能随机出现加减，不会正确输出结果、不能进行循环练习等等问题。经过我的努力及同学的帮助，这些问题得到克服，并且使程序的功能也得到了进一步的完善。现在它能对出错的题目发出报警声，并且给出正确答案。最后还能分别输出对错的题数及所得分数。

在这次设计过程中，不仅复习课本上所学知识，还通过查资料、问同学学到了课本上没有的知识。从而启发我，要想写好程序，在写好课本知识的同时还需要多读和专业有关的一些书籍，同时还需要多动脑子，尽量把所学的知识综合起来应用，力争写出完美的程序。除此之外，我还得到了一些有用的教训：写程序时必须细心，不能输错一个字符标点，就连全角半角也得注意。在修改时要有耐心，编译出错后必须逐个错误去改正，绝不能心急浮躁，否则修改之后还会有新的错误。