

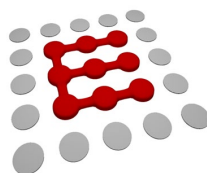


北京理工大学  
Beijing Institute of Technology

# 本科实验报告

实验名称： 学生成绩管理（链表）

课程名称：	数据结构与算法设计实验	实验时间：	2017/3/31
任课教师：	李岩	实验地点：	良乡机房 401
实验教师：	苏京霞	实验类型：	<input type="checkbox"/> 原理验证 <input checked="" type="checkbox"/> 综合设计 <input type="checkbox"/> 自主创新
学生姓名：	施念		
学号/班级：	1120161302/05011609	组 号：	72
学 院：	信息与电子学院	同组搭档：	
专 业：	电子信息类	成 绩：	



信息与电子学院

SCHOOL OF INFORMATION AND ELECTRONICS

## 1. 实验目的

设计一个“学生成绩管理系统”。主要实现学生信息的录入、添加、修改、删除、排序和查看等基本功能。

## 2. 实验题目

编写一个学生成绩管理程序。学生成绩以一个学生一条记录的形式存储，每个学生记录包含的信息有序号、学号、姓名及 5 门功课的成绩。具体功能如下：

- a. 获取学生成绩。可以从文件中读取成绩，也可直接录入。
- 查询学生成绩，输入学生学号或姓名等信息后，显示学生在成绩单中的位置及所有信息。
- b. 添加学生成绩，在学生成绩单的指定位置添加学生成绩信息。
- c. 修改学生成绩，在学生成绩单中，修改指定学生的成绩信息。
- d. 删除学生成绩，在学生成绩单中，删除指定学生的成绩信息。
- e. 保存文件。当学生成绩单发生添加、修改、删除等变化后，都要对最终结果进行保存。

## 3. 实验基础知识

利用单链表储存学生成绩，一个节点表示一个学生。通过对单链表节点的增添、删除、信息、冒泡排序以及与文件结合的读取保存等操作实现对学生成绩的管理。

## 4. 概要分析

### 1) 构造学生信息结构体储存学生信息

代码如下：

```
struct student { //学生信息结构体定义
    char num[20]; //学号
    char name[20]; //姓名
    float score[6]; //五门功课成绩以及平均分成绩
    int no; //序号
    struct student * next; //指向下一个节点
};
typedef struct student STU;
```

### 2) 全局变量

全局变量只有一个：

STU \*head; //头指针为全局变量，便于访问

### 3) 链表选择

选择没有头结点的单链表，单链表的头指针为全局变量。

### 4) 函数模块

主函数：登录前要输入账号和密码（初始都是六个 0），验证成功后进入功能菜单选项，三次验证机会用完则强制退出。每次调用完子

函数都返回菜单，直到用户选择功能选项的退出才结束。（结束前询问是否保存文件）

子函数：实现各个功能的操作，子函数之前可相互调用

void Pass();//登录身份验证，三次输入机会

void Login\_MFace();//登录界面

void Input\_Data();//手动输入数据

void Insert\_Data();//插入函数

void Save\_Data();//保存信息到文件中

void Up\_Data();//从文件中读取信息并建立链表

void Search\_Infor();//查找函数

void S\_key(int i);//按学号、姓名查找

void display(STU \* p);//将 p 指针所指学生信息显示出来

void Sort\_Data();//排序函数

void Sort\_num();//按照学号排序

void Sort\_mark(int choice);//按照第 choice-2 门成绩排序（包括平均分）

void Modify\_Data();//修改信息函数

void Modify\_key(int j);//按照关键字（学号或姓名）检索修改

void Calculate\_Average(STU \* p);//计算 p 指针所指学生的平均分

void Delete();//删除某个学生的信息

void PRINT(STU \* p);//输入 p 指针所指学生的所有信息

void No\_i();//为学生插上序号。

#### 5) 功能菜单：

```
void Login_MFace() {
    system("cls");
    printf("请根据提示选择相应的操作：\n\t");
    printf("1-----显示全部学生信息\n\t");
    printf("2-----查询信息\n\t");
    printf("3-----排序\n\t");
    printf("4-----修改信息\n\t");
    printf("5-----删除信息\n\t");
    printf("6-----添加信息\n\t");
    printf("7-----保存信息\n\t");
    printf("0-----退出\n\t");
    return;
}
```

## 5. 调试分析

- 1) 涉及到文件的操作，可以在直接录入，文件默认保存在 D 盘子目录下。
- 2) 在每一个选择输入前的时候，都要用 fflush (stdin) 对输入缓存区

进行清空，防止影响后面数据的输入。在用户选择输入后要对所输入的选择进行判断，正确后才执行下一条语句。

- 3) 对单链表进行冒泡排序时，要进行节点之前的交换，所以要在第一个节点之前加一个新节点以便交换。
- 4) 在结构定义里，从最一开始的定义一个数组和一个变量储存五科成绩，到后来改为用一个数组储存五科成绩和平均分，在进行排序的时候，将平均分和五科成绩等同看待，减少了一个子函数。
- 5) 将学生数据进行保存写入文件里的时候，要避免文件结尾换行符的产生（换行符对文件的读取有影响），所以采用最后一个学生数据和之前的学生数据区别写入文件。

## 6. 测试结果

### 1) 验证身份

欢迎使用成绩管理系统：

用户名：000000

密 码：000000

登陆成功！

请按任意键继续. . .

### 2) 录入信息

### 3) 读取文件

请选择数据输入方式：

1->手动输入

2->文件导入

2

读取成功！

请按任意键继续. . .

### 4) 功能菜单

请根据提示选择相应的操作：

1-----显示全部学生信息

2-----查询信息

3-----排序

4-----修改信息

5-----删除信息

6-----添加信息

7-----保存信息

0-----退出

### 5) 显示全部学生信息

全部学生信息为：

序号	姓名	学号	高数	语文	英语	C语言	数据结构	平均分
1	李雪铮	1120161298	99.00	88.00	77.00	66.00	55.00	77.00
2	厉运达	1120161299	99.00	100.00	98.00	79.00	99.00	95.00
3	马腾	1120161300	99.00	88.00	99.00	77.00	69.00	86.40
4	施念	1120161302	99.00	78.00	74.00	98.00	99.00	89.60

请按任意键继续. . .

### 6) 查询信息（按学号查找）

请输入学号:  
1120161302

序号	姓名	学号	高数	语文	英语	C语言	数据结构	平均分
4	施念	1120161302	99.00	78.00	74.00	98.00	99.00	89.60

是否继续查找:  
1. 通过学号查找  
2. 通过姓名查找  
3. 退出查找

## 7) 查询信息（按姓名查找）

请输入姓名:  
李雪铮

序号	姓名	学号	高数	语文	英语	C语言	数据结构	平均分
1	李雪铮	1120161298	99.00	88.00	77.00	66.00	55.00	77.00

是否继续查找:  
1. 通过学号查找  
2. 通过姓名查找  
3. 退出查找

## 8) 排序菜单

排序方式为:

1. 按照学号排序
2. 按照高数成绩排序
3. 按照语文成绩排序
4. 按照英语成绩排序
5. 按照C语言成绩排序
6. 按照数据结构成绩排序
7. 按照平均分排序

## 9) 排序结果（以学号排序为例）

全部学生信息为:

序号	姓名	学号	高数	语文	英语	C语言	数据结构	平均分
1	李雪铮	1120161298	99.00	88.00	77.00	66.00	55.00	77.00
2	厉运达	1120161299	99.00	100.00	98.00	79.00	99.00	95.00
3	马腾	1120161300	99.00	88.00	99.00	77.00	69.00	86.40
4	施念	1120161302	99.00	78.00	74.00	98.00	99.00	89.60

请按任意键继续. . .

## 10) 修改信息菜单

请提供您要修改的学生的一项信息:  
1. 学号（推荐）  
2. 姓名

## 11) 修改信息

您要修改的学生信息为:

序号	姓名	学号	高数	语文	英语	C语言	数据结构	平均分
4	施念	1120161302	99.00	78.00	74.00	98.00	99.00	89.60

可修改的项目有:  
1. 姓名  
2. 学号  
3. 高数  
4. 语文  
5. 英语  
6. C语言  
7. 数据结构  
8. 以上全部

请输入修改项目:

## 12) 删除信息

```

请输入要删除的学生的学号:
1120161302
您要删除的学生数据为:
学号      姓名      学号      高数      语文      英语      C语言      数据结构      平均分
1         施念      1120161302      99.00      78.00      74.00      98.00      99.00      89.60

删除成功!

是否继续删除:
1. 是
2. 否

```

### 13) 增加学生信息

```

请按提示输入插入学生的信息:
姓名: 王亚静
学号: 1120161314
高数: 98
语文: 99
英语: 100
C语言: 78
数据结构: 98

插入成功!

是否继续插入? (Y/N)

```

### 14) 保存信息

```

请根据提示选择相应的操作:
1-----显示全部学生信息
2-----查询信息
3-----排序
4-----修改信息
5-----删除信息
6-----添加信息
7-----保存信息
0-----退出
7

保存成功!

请按任意键继续. . .

```

### 15) 退出时提示是否要保存

```

请根据提示选择相应的操作:
1-----显示全部学生信息
2-----查询信息
3-----排序
4-----修改信息
5-----删除信息
6-----添加信息
7-----保存信息
0-----退出
0

是否保存操作后的数据?
1. 是
2. 否
1

保存成功!

请按任意键继续. . .

```

## 7. 附录（源代码）

```

#include"stdio.h"
#include"stdlib.h"
#include"string.h"

```

```

struct student { //学生信息结构体定义

```

```

    char num[20];
    char name[20];
    float score[6]; // 五门功课成绩以及平均分成绩
    int no;
    struct student * next;
};

typedef struct student STU;

STU *head; // 头指针为全局变量，便于访问

void Pass(); // 登录身份验证
void Login_MFace(); // 登录界面
void Input_Data(); // 手动输入数据
void Insert_Data(); // 插入函数
void Save_Data(); // 保存信息到文件中
void Up_Data(); // 从文件中读取信息并建立链表
void Search_Infor(); // 查找函数
void S_key(int i); // 按学号、姓名查找
void display(STU * p); // 将 p 指针所指学生信息显示出来
void Sort_Data(); // 排序函数
void Sort_num(); // 按照学号排序
void Sort_mark(int choice); // 按照第 choice-2 门成绩排序（包括平均分）
void Modify_Data(); // 修改信息函数
void Modify_key(int j); // 按照关键字（学号或姓名）检索修改
void Calculate_Average(STU * p); // 计算 p 指针所指学生的平均分
void Delete(); // 删除某个学生的信息
void PRINT(STU * p); // 输入 p 指针所指学生的所有信息
void No_i(); // 为学生插上序号

int main() {
    printf("\n\n\t\t\t 写在使用前： \n");
    printf("\t\t\t 登录的用户名和密码均为 000000 （六个零） \n\n\n");
    system("pause");
    system("cls");
    printf("\n 欢迎使用成绩管理系统： \n");
    Pass();
    system("cls");
    STU * j, * p;
    int n, m;
    int i;
    int flag = 1;

```

```

int choice;
printf("\n\n 请选择数据输入方式： \n1->手动输入\n2->文件导入\n");
scanf("%d", &n);
if (n == 1) Input_Data();
if (n == 2) Up_Data();
for (p = head;p->next != NULL;p = p->next) {
    Calculate_Average(p);
}
Calculate_Average(p);
No_i();
while (flag) {
    Login_MFace();
    scanf("%d", &m);
    switch (m) {
        case 0:
            printf("\n 是否保存操作后的数据？ \n");
            printf("1. 是\n");
            printf("2. 否\n");
            scanf("%d", &choice);
            while (choice != 1 && choice != 2) {
                printf("\n 输入有误，请重新输入（1-2）： ");
                scanf("%d", &choice);
            }
            switch (choice) {
                case 1: Save_Data(); break;
                case 2: break;
            }

            flag = 0; break;
        case 1: printf("\n 全部学生信息为： \n");
            printf("序号          姓名          学号\n");
            高数      语文      英语      C 语言      数据结构      平均分\n");
            for (j = head;j != NULL;j = j->next)

                printf("%-15d%-15s%-15s%-10.2f%-10.2f%-10.2f%-10.2f%-10.2f%-10.2f%-10.2f\n", j->no,j->name, j->num, j->score[0], j->score[1], j->score[2], j->score[3], j->score[4], j->score[5]);
                break;
        case 2: Search_Infor(); break;
        case 3: Sort_Data(); break;
        case 4: Modify_Data(); break;
        case 5: Delete(); break;
    }
}

```



```

        case 6: Insert_Data();break;
        case 7: Save_Data(); break;
        default:printf("\n\n 选择错误，请重新输入
");Login_MFace();break;
    };
    system("pause");
}
return 0;
}

void Pass() {
    int i = 0, j = 0;
    char ch;
    char name[10], password[10];
    while (i < 3) {
        printf("\n\t\t\t 用户名: ");
        gets(name);
        printf("\n\t\t\t 密 码: ");
        gets(password);
        if (!strcmp(name, "000000") && !strcmp(password, "000000")) {
            printf("\n\t\t\t 登陆成功! \n\n");
            system("pause");
            return;
        }
        else {
            printf("\n\t\t\t 登陆失败，你还有%d 次机会\n", 2 - i);
            i++;
        }
    }
    printf("\n\t\t\t 尝试次数已用完，您已被强制退出");
    exit(0);
}

void Login_MFace() {
    system("cls");
    printf("请根据提示选择相应的操作: \n\t\t");
    printf("1-----显示全部学生信息\n\t\t");
    printf("2-----查询信息\n\t\t");
    printf("3-----排序\n\t\t");
    printf("4-----修改信息\n\t\t");
    printf("5-----删除信息\n\t\t");
    printf("6-----添加信息\n\t\t");

```

```

printf("7-----保存信息\n\t\t");
printf("0-----退出\n\t\t");
return;
}

void Input_Data() {
    int n = 1; //判断输入的是第几个学生
    int flag = 1; //循环条件
    char ch;
    STU * p1;
    p1 = NULL;
    head = NULL;

    head = (STU *)malloc(sizeof(STU)); //开辟一个新节点
    head->next = NULL;

    if (head == NULL) //节点开辟不成功
    {
        printf("\n 链表创建失败，请稍后再试！ \n");
        return;
    }

    while (flag)
    { //循环输入
        if (n == 1) {
            printf("\t\t输入学生信息\n\t\t请输入第%d个学生的信息:\n",
n);
            PRINT(head);

        }
        else {
            p1 = (STU *)malloc(sizeof(STU));
            printf("\t\t输入学生信息\n\t\t请输入第%d个学生的信息:\n",
n);
            PRINT(p1);
            p1->next = head;
            head = p1;
        }
        n++;
        printf("是否继续录入? (Y/N)\n");
        fflush(stdin); //清除缓存区
        scanf("%s", &ch);
    }
}

```

```

        while (ch != 'Y' && ch != 'N') {
            printf("\n 输入错误，请重新输入： ");
            scanf("%s", &ch);
        }
        switch (ch) {
            case 'Y': break;
            case 'N': flag = 0; break;
        }
        system("cls");
    }
}

void Insert_Data() {
    system("cls");
    char ch;
    int flag = 1;
    STU *p;
    p = head;
    printf("\n 请按提示输入插入学生的信息： \n");
    while (flag) {
        p = (STU *)malloc(sizeof(STU));
        if (p == NULL) //节点开辟不成功
        {
            printf("\n 链表创建失败，请稍后再试！ \n");
            return;
        }
        PRINT(p); //录入数据
        Calculate_Average(p);
        p->next = head;
        head = p;
        if (p != NULL)
            printf("\n 插入成功！ \n");
        printf("\n 是否继续插入？ (Y/N)\n");
        scanf("%s", &ch); //需要用 %s 而不可以用 %c，缓存区残留换
行符
        while (ch != 'Y' && ch != 'N') {
            printf("输入错误，请重新输入： ");
            fflush(stdin);
            scanf("%s", &ch);
        }
        switch (ch) {
            case 'Y': break;

```

```

        case 'N': flag = 0; break;
    }
}
return;
}

void Save_Data() {
    FILE * fp;
    STU * p;
    if ((fp = fopen("D:\\StudentData.txt", "w")) == NULL) {
        printf("无法打开文件\n");
        exit(1);
    }
    p = head;
    do {
        fprintf(fp,
            "%3d    %10s    %10s    %8.2f %8.2f %8.2f %8.2f %8.2f %8.2f\n",
            p->no, p->name, p->num, p->score[0], p->score[1], p->score[2],
            p->score[3], p->score[4], p->score[5]);
        p = p->next;
    } while (p->next != NULL); // 保证文件最后一个字符不是换行符，解决了读取数据时换行符的问题
    fprintf(fp,
        "%3d    %10s    %10s    %8.2f %8.2f %8.2f %8.2f %8.2f %8.2f",
        p->no, p->name, p->num, p->score[0], p->score[1], p->score[2], p->score[3],
        p->score[4], p->score[5]);
    printf("保存成功! \n\n");
    fclose(fp);
}

void No_i() {
    STU * p;
    p = head;
    int i = 1;
    while (p->next != NULL) {
        p->no = i;
        i++;
        p = p->next;
    }
    p->no = i;
    return;
}

```

```

}

void Up_Data() {
    int n = 1;
    char ch;
    FILE * fp;
    STU * p, *q;
    if ((fp = fopen("D:\\StudentData.txt", "r")) == NULL) {
        printf("无法打开文件\n");
        exit(1);
    }
    p = (STU*)malloc(sizeof(STU));
    p->next = NULL;
    while (!feof(fp)) {
        if (n == 1) {
            fscanf(fp, "%d%s%s%f%f%f%f%f", &p->no, p->name,
p->num, &p->score[0], &p->score[1], &p->score[2], &p->score[3],
&p->score[4], &p->score[5]);
        }
        else {
            q = (STU*)malloc(sizeof(STU));
            fscanf(fp, "%d%s%s%f%f%f%f%f", &q->no, q->name,
q->num, &q->score[0], &q->score[1], &q->score[2], &q->score[3],
&q->score[4], &p->score[5]);
            q->next = p;
            p = q;
        }
        n += 1;
    }
    head = p;
    fclose(fp);
    printf("读取成功! \n");
    system("pause");
    system("cls");
}

```

```

void Search_Infor() {
    int choice;
    system("cls");
    printf("查找方式为: \n");
    printf("\t\t1.通过学号查找（推荐）\n");
}

```

```

printf("\t\t2.通过姓名查找\n");
scanf("%d", &choice);
while (choice != 1 && choice != 2) {
    fflush(stdin);//清除缓存区
    printf("输入有误，请重新输入（1-2）：");
    scanf("%d", &choice);
}
S_key(choice);
return;
}

void S_key(int i) {
    system("cls");
    STU *p;
    p = NULL;
    char key[20];
    int choice;
    if (i == 1) {
        printf("请输入学号：\n");
    }
    else {
        printf("请输入姓名：\n");
    }

    scanf("%s", key);
    p = head;//无表头节点的链表
    if (p == NULL) {
        printf("\n 空表，无学生数据！");
    }
    if (i == 1) {
        while (p != NULL && strcmp(p->num, key)) { //两个判断条件不能颠倒（最后一个节点时 p 已空，无 p->next）
            p = p->next;
        }
    }
    else {
        while (p != NULL && strcmp(p->name, key)) { //两个判断条件不能颠倒（最后一个节点时 p 已空，无 p->next）
            p = p->next;
        }
    }
}

```

```

if (p == NULL) {
    printf("找不到此学生！ ");
}
else
{
    display(p);
}
printf("是否继续查找： \n");
printf("1. 通过学号查找\n");
printf("2. 通过姓名查找\n");
printf("3. 退出查找\n");
scanf("%d", &choice);
while (choice != 1 && choice != 2 && choice != 3) {
    printf("输入有误，请重新输入（1-3）： ");
    scanf("%d", &choice);
}
switch (choice) {
    case 1:
    case 2: S_key(choice); break;
    case 3: return;
}
return;
}

void display(STU * p) {
    printf("序号      姓名      学号      高数\n");
    printf("语文      英语      C语言      数据结构      平均分\n");
    printf("%-15d%-15s%-15s%-10.2f%-10.2f%-10.2f%-10.2f%-10.2f%-10.2f%-10.2f\n", p->no, p->name, p->num, p->score[0], p->score[1], p->score[2], p->score[3], p->score[4], p->score[5]);
    return;
}

void Sort_Data() {
    int choice;
    system("cls");
    printf("排序方式为： \n");
    printf("\t\t1.按照学号排序\n");
    printf("\t\t2.按照高数成绩排序\n");
    printf("\t\t3.按照语文成绩排序\n");
    printf("\t\t4.按照英语成绩排序\n");

```

```

printf("\t\t5.按照 C 语言成绩排序\n");
printf("\t\t6.按照数据结构成绩排序\n");
printf("\t\t7.按照平均分排序\n");
scanf("%d", &choice);
while (choice < 1 || choice > 7) {
    printf("输入有误，请重新输入（1-7）： ");
    scanf("%d", &choice);
}
switch (choice) {
case 1: Sort_num(); break;
default: Sort_mark(choice - 2); break;
}
No_i();
printf("\n 排序成功\n");
}

void Sort_num() {
    STU *endpt; //控制循环比较
    STU *p;     //临时指针变量
    STU *p1, *p2;
    p1 = (STU *)malloc(sizeof(STU));
    p1->next = head; //增加一个节点，放在第一个节点的前面(因为第一个节点没有前驱，不能交换地址)
    head = p1;       //让 head 指向 p1 节点，排序完成后，p1 节点释放掉
    for (endpt = NULL; endpt != head; endpt = p)
    {
        for (p = p1 = head; p1->next->next != endpt; p1 = p1->next)
        {
            if (strcmp(p1->next->num, p1->next->next->num) > 0) {
                //如果前面的节点比后面节点的大，则交换
                p2 = p1->next->next;
                p1->next->next = p2->next;
                p2->next = p1->next;
                p1->next = p2;
                p = p1->next->next;
            }
        }
    }
    p1 = head;
    head = head->next; //让 head 指向排序后的第一个节点
    free(p1);         //释放 p1
}

```



```

    p1 = NULL;
    return;
}

void Sort_mark1(int choice) {
    STU *endpt; //控制循环比较
    STU *p;      //临时指针变量
    STU *p1, *p2;

    p1 = (STU *)malloc(sizeof(STU));
    p1->next = head; //增加一个节点, 放在第一个节点的前面(第一个节点没有前驱, 不能交换地址)
    head = p1;       //让 head 指向 p1 节点, 排序完成后, 把 p1 节点释放掉

    for (endpt = NULL; endpt != head; endpt = p)
    {
        for (p = p1 = head; p1->next->next != endpt; p1 = p1->next)
        {
            if (p1->next->score[choice] < p1->next->next->score[choice])
            //如果前面的节点键值比后面节点的键值小, 则交换
            {
                p2 = p1->next->next;
                p1->next->next = p2->next;
                p2->next = p1->next;
                p1->next = p2;
                p = p1->next->next;
            }
        }
    }
    p1 = head;
    head = head->next; //让 head 指向排序后的第一个节点
    free(p1);         //释放 p1
    p1 = NULL;
    return;
}

void Modify_Data() {
    int choice;
    system("cls"); //清屏
    printf("请提供您要修改的学生的一项信息: \n");
    printf("\t\t1.学号 (推荐) \n");

```

```

printf("\t\t2.姓名\n");
scanf("%d", &choice);
printf("\n");
while (choice < 1 || choice>2) {
    fflush(stdin);
    printf("输入有误，请重新输入（1-2）： ");
    scanf("%d", &choice);
}
switch (choice) {
case 1:Modify_key(1);break;
case 2:Modify_key(2);break;
}
}

void Modify_key(int j) {
    STU *p;
    char key[20];
    int choice;
    int i;
    if (j == 1) {
        printf("请输入学号： \n");
    }
    else {
        printf("请输入姓名： \n");
    }

    scanf("%s", key);
    p = head;
    if (p == NULL) {
        printf("空表，无学生数据！");
    }
    if (j == 1) {
        while (p != NULL&&strcmp(p->num, key)) { //不考虑重名的情况
            p = p->next;
        }
    }
    else {
        while (p != NULL&&strcmp(p->name, key)) { //不考虑重名的情况
            p = p->next;
        }
    }
}

```

```

}

if (p == NULL) {
    printf("\n 找不到此学生！ ");
}
else
{
    system("cls");
    printf("您要修改的学生信息为： \n");
    display(p);
    printf("\n 可修改的项目有： \n");
    printf("1.姓名\t\t5.英语\t\n");
    printf("2.学号\t\t6.C 语言\t\n");
    printf("3.高数\t\t7.数据结构\t\n");
    printf("4.语文\t\t8.以上全部\n");
    printf("请输入修改项目： ");
    scanf("%d", &i);
    while (i < 1 || i>8) {
        printf("\n 输入有误，请重新输入（1-8）： ");
        scanf("%d", &i);
    }
    printf("\n 请输入要修改的学生的新的数据： \n");
    switch (i) {
        case 1:printf("请输入新的姓名： ");
            scanf("%s", p->name);
            break;
        case 2:printf("请输入新的学号： ");
            scanf("%s", p->num);
            break;
        case 3:printf("请输入新的高数分数： ");
            scanf("%f", &p->score[0]);
            break;
        case 4:printf("请输入新的语文分数： ");
            scanf("%f", &p->score[1]);
            break;
        case 5:printf("请输入新的英语分数： ");
            scanf("%f", &p->score[2]);
            break;
        case 6:printf("请输入新的 C 语言分数： ");
            scanf("%f", &p->score[3]);
            break;
        case 7:printf("请输入新的数据结构分数： ");

```

```

        scanf("%f", &p->score[4]);
        break;
    case 8: PRINT(p);
        break;
    }
    Calculate_Average(p);
}
printf("\n 是否继续修改: ");
printf("\n1. 继续修改");
printf("\n2. 退出修改\n 您的选择是: ");
scanf("%d", &choice);
while (choice < 1 || choice > 2) {
    fflush(stdin); //清除缓冲区
    printf("\n 输入有误, 请重新输入 (1-2): ");
    scanf("%d", &choice);
}
switch (choice) {
    case 1: Modify_Data(); break;
    case 2: return;
}
}

```

```

void Calculate_Average(STU * p) {
    float sum = 0.0;
    for (int i = 0; i < 5; i++) { //求和
        sum += p->score[i];
    }
    p->score[5] = sum / 5.0; //求平均分
}

```

```

void Delete() {
    system("cls");
    STU *p, *q;
    char num[20];
    int choice;
    printf("请输入要删除的学生的学号: \n");
    scanf("%s", num);
    p = head; //无表头节点的链表
    if (p == NULL) {
        printf("\n 空表, 无学生数据! ");
    }
    else if (!strcmp(p->num, num)) {

```

```

        printf("您要删除的学生数据为: \n");
        display(p);
        head = p->next;//第一个节点即为要删除的节点
        free(p);
        printf("\n 删除成功! \n");
    }
    else {
        while (p->next != NULL&& strcmp(p->next->num, num)) { //两个
判断条件不能颠倒（最后一个节点时 p 已空，无 p->next）
            p = p->next;
        }

        if (p->next == NULL) {
            printf("找不到此学生! ");
        }
        else
        {
            printf("您要删除的学生数据为: \n");
            display(p->next);
            q = p->next;//q 为要删除的节点
            p->next = q->next;
            free(q);
            printf("\n 删除成功! \n");
        }
    }

    printf("\n 是否继续删除: \n");
    printf("1. 是\n");
    printf("2. 否\n");
    scanf("%d", &choice);
    while (choice != 1 && choice != 2) {
        printf("\n 输入有误，请重新输入（1-2）: ");
        scanf("%d", &choice);
    }
    switch (choice) {
        case 1:Delete();break;
        case 2:return;
    }
    return;
}

void PRINT(STU * p) {

```

```
printf("姓名: ");
scanf("%s", p->name);
printf("学号: ");
scanf("%s", p->num);
printf("高数: ");
scanf("%f", &p->score[0]);
printf("语文: ");
scanf("%f", &p->score[1]);
printf("英语: ");
scanf("%f", &p->score[2]);
printf("C 语言: ");
scanf("%f", &p->score[3]);
printf("数据结构: ");
scanf("%f", &p->score[4]);
```

```
}
```