

# 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 19 级（3+2）

小组组长： 姜滢

小组成员： 黄璐瑶、周梦佳、任婷婷

指导教师： 张万里 职称： 讲师

2022 年 1 月 6 日

## 目录

一、设计目的 .....	- 2 -
二、设计内容 .....	- 2 -
三、需求分析 .....	- 2 -
四、系统设计 .....	- 3 -
五、概要设计 .....	- 5 -
1、工作分配 .....	- 5 -
2、流程图 .....	- 5 -
3、程序模块 .....	- 5 -
六、主程序 .....	- 10 -
七、运行截图 .....	- 10 -
八、总结 .....	- 15 -
九、参考文献 .....	- 16 -
十、附录：源代码（ <b>main.app</b> ） .....	- 16 -

# 学生信息管理系统

## 一、设计目的

1.数据结构课程设计是综合运用数据结构课程中学到的几种典型数据结构，以及程序设计语言(C 语言)，自行实现一个较为完整的应用系统的设计与开发。

2.课程设计，自己通过系统分析、系统设计、编程调试，写实验报告等环节，进一步掌握应用系统设计的方法和步骤，灵活运用并深刻理解典型数据结构在软件开发中的应用。

3.将知识应用于实际的方法，提高分析和解决问题的能力，增加综合能力。学生信息管理系统:

(1)熟练掌握链表存储结构及其建立过程和常用操作:

(2)学会自己调试程序的方法并掌握一定的技巧

## 二、设计内容

现需要编写一个学生信息管理系统，使用人群是辅导员和学生，辅导员可以对学生基本信息进行删除、修改和查看成绩排序，学生可以通过学号和姓名进行查找学生的信息。

## 三、需求分析

1.每位学生的信息有：学号、姓名、年龄、性别、地址、电话、专业、成绩等，用链表对学生的信息进行存储

2.全部数据可以只存放在内存中

3.系统中能实现的操作和功能如下：

1) 添加学生信息：

根据不同的学生分别输入下列信息：学号、姓名、年龄、性别、地址、电话、专业、成绩

2) 修改学生信息：

可以对不同的学生信息进行修改：学号、姓名、年龄、性别、地址、电话、专业、成绩

3) 删除学生信息:

删除某个学生的所有信息

4) 查询学生信息:

可以根据该学生的学号或者姓名具体查询该学生的学生信息

5) 插入学生信息:

根据学生的学号、姓名、年龄、性别、地址、电话、专业、成绩进行插入

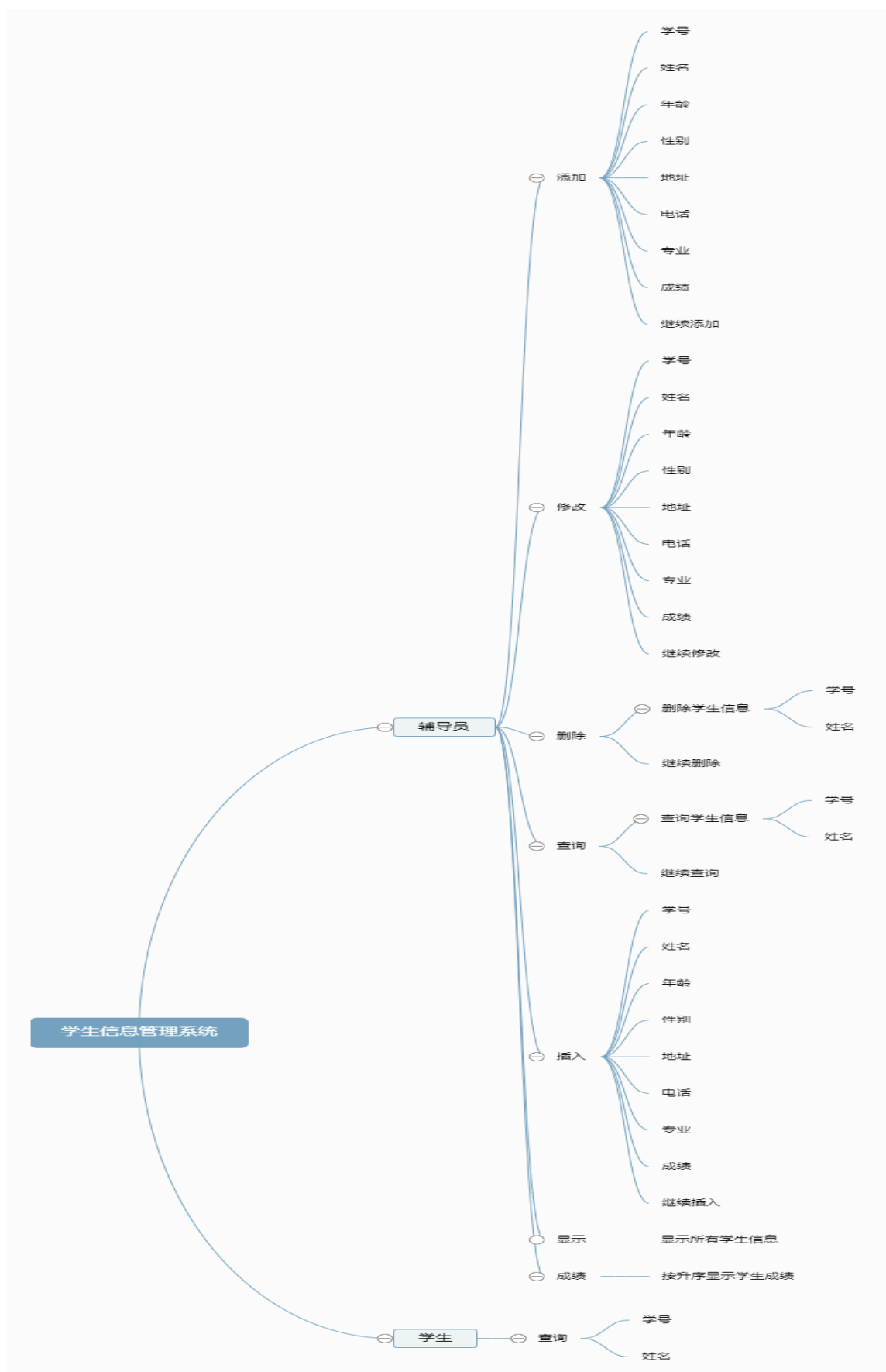
6) 显示学生信息:

显示所有学生的具体信息

7) 成绩升序排序:

学生成绩按升序排序显示出来

## 四、系统设计

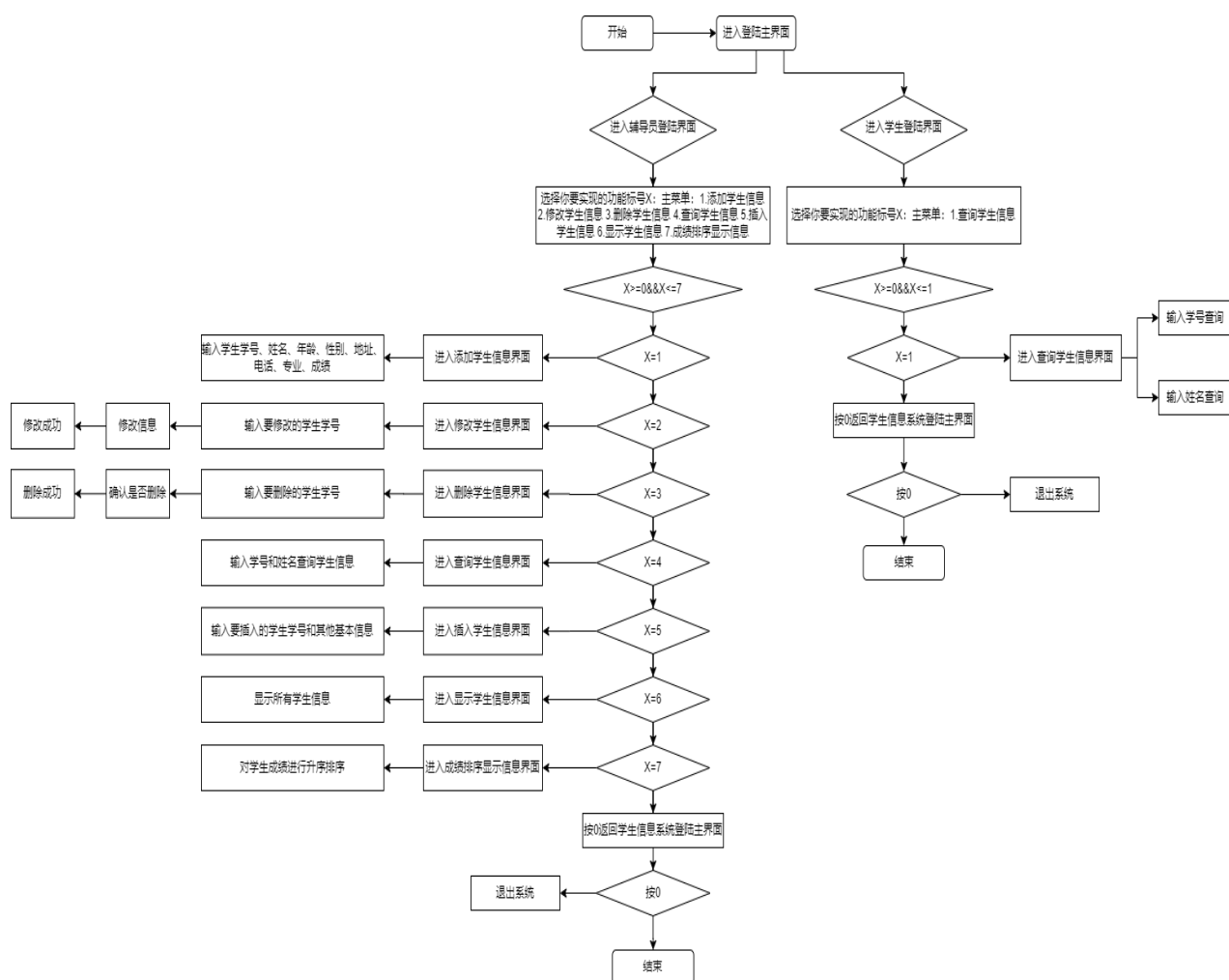


## 五、概要设计

### 1、工作分配

- (1) 框架设计: 由姜滢进行构思, 同时参考其他人的意见。
- (2) 系统设计: 由任婷婷进行框架的编写。
- (3) 程序设计: 由姜滢、任婷婷、黄璐瑶、周梦佳进行各项功能的编辑。
- (4) 程序调试: 由黄璐瑶进行程序的调试与调整。
- (5) 文档制作, 由任婷婷、姜滢完成文档的编辑工作。

### 2、流程图



### 3、程序模块

程序的模块为

学生信息管理系统主界面选择登录人群:

```

int MainMenu()
{
    int a;
    system("cls"); //清屏
    printf("=====\\n");
    printf("==                  欢迎来到学生信息管理系统主菜单!!!                ==\\n");
    printf("==\\n");
    printf("==                  1. 指导员                    ==\\n");
    printf("==                  2. 学生                        ==\\n");
    printf("==                  0. 退出                        ==\\n");
    printf("=====\\n");
    scanf("%d", &a);
    return a;
}

```

辅导员添加信息:

```

    if (strcmp(p1->number, p3->number) == 0) //判断p1和p3是否重复
    {
        printf("重复!\\n");
        scanf("%d", &a);
        while (a != 1) //输入1时才可以重新返回系统主界面
        {
            printf("你输入的数字有误\\n");
            scanf("%d", &a);
        }
        return x; //返回头指针
    }
    p3 = p3->next; //指向下一个结点
}
p2 = x;
while (p2)
{
    if (p2->next == NULL)
    {
        p2->next = p1;
        p1->next = NULL;
    }
    p2 = p2->next;
}
length++; //链表长度加1
printf("添加成功!\\n");
printf("\\n是否继续?(y/n):");
choice1 = getch(); //不回显函数 当用户按下某个字符时 函数自动读取 无需按回车
printf("\\n");
}
return x; //返回头指针
}

stu *creatlink(stu *x)
{
    stu *p1, *p2, *p3;
    int a;
    int i = 1;
    char choice1;
    x = p2 = (stu*)malloc(N); //创建头结点 p2始终指向尾结点, 开始时指向头结点
    strcpy(p2->number, "-99999"); //拷贝字符串
    x->next = NULL;
    choice1 = 'y';
    for (i = 1; choice1 == 'y'; i++)
    {
        p1 = (stu*)malloc(N);
        printf("=====\\n");
        printf("请输入第%d个学生的信息:\\n", i);
        printf("学号: ");
        scanf("%s", p1->number);
        printf("姓名: ");
        scanf("%s", p1->name);
        printf("年龄: ");
        scanf("%s", p1->age);
        printf("性别: ");
        scanf("%s", p1->sex);
        printf("地址: ");
        scanf("%s", p1->add);
        printf("电话: ");
        scanf("%s", p1->Tel);
        printf("专业: ");
        scanf("%s", p1->special);
        printf("成绩: ");
        scanf("%d", &p1->score);
        if (x == NULL)
        {
            x = p1;
            x->next = NULL; //尾结点的next域值为NULL
        }
        else
        {
            p3 = x;
            while (p3)
            {

```

辅导员修改信息:

```

        scanf("%s", ch);
        strcpy(p->number, ch);
        break;
    case 2:
        scanf("%s", ch);
        strcpy(p->name, ch);
        break;
    case 3:
        scanf("%s", ch);
        strcpy(p->age, ch);
        break;
    case 4:
        scanf("%s", ch);
        strcpy(p->sex, ch);
        break;
    case 5:
        scanf("%s", ch);
        strcpy(p->add, ch);
        break;
    case 6:
        scanf("%s", ch);
        strcpy(p->Tel, ch);
        break;
    case 7:
        scanf("%s", ch);
        strcpy(p->special, ch);
        break;
    case 8:
        scanf("%d", &cp);
        p->score = cp;
        break;
    default:
        break;
    }
    printf("修改后学生信息如下:\n");
    printf("\n学号:%s 姓名:%s 年龄:%s 性别:%s 地址:%s 电话:%s 专业:%s 成绩:%d\n\n", p->number, p->name, p->age, p->sex, p->add, p->Tel, p->special, p->score);
    printf("修改成功!\n");
    printf("\n是否继续修改? \n是y否n:   ");
    choice = getch();
    if (choice == 'n')
        break;
}
return y;
}

stu *modify(stu *y)
{
    int choi;
    char choice;
    int cp;
    char num[20], ch[20];
    stu *p;
    if (y == NULL)
    {
        printf("\n请输入要修改学生的学号: ");
        scanf("%s", num);
        printf("\n该生不存在!按任意键继续..");
        getch();
        return y; // 返回头指针 (y为形参变量)
    }
    printf("\n请输入要修改学生的学号: ");
    scanf("%s", num);
    while (p)
    {
        if (strcmp(p->number, num) != 0)
            p = p->next;
        else
            break;
    }
    if (!p)
    {
        printf("\n该生不存在!按任意键继续..");
        getch();
        return y;
    }
    printf("学生信息如下:\n");
    printf("\n学号:%s 姓名:%s 年龄:%s 性别:%s 地址:%s 电话:%s 专业:%s 成绩:%d\n\n", p->number, p->name, p->age, p->sex, p->add, p->Tel, p->special, p->score);

    while (1)
    {
        printf("\n请选择要修改的项: \n");
        printf("\n学号 2 姓名 3 年龄 4 性别 5 地址 6 电话 7 专业 8 成绩\n");
        scanf("%d", &choi);
        printf("请输入修改后的内容: \n");
        switch (choi)
        {
            case 1:

```

## 辅导员删除信息:

```

stu* delet(stu* x)
{
    char num[20];
    stu* p, * pre;
    char choice2;
    if (x == NULL)
    {
        printf("\n错误执行!");
        getch();
        return x;
    }
    printf("\n请输入要删除学生的学号: ");
    scanf("%s", num);
    pre = x;
    int judge = 0;
    p = pre->next;
    while (p)
    {
        if (strcmp(p->number, num) == 0)
        {
            judge = 1;
            printf("你要删除的信息如下,请确认是否删除:\n");
            printf("\n学号:%s 姓名:%s 年龄:%s 性别:%s 地址:%s 电话:%s 专业:%s 成绩:%d\n\n", p->number, p->name, p->age, p->sex, p->add, p->Tel, p->special, p->score);
            printf("\n是y否n: "); // 显示要删除的学生的信息
            choice2 = getch();
            if (choice2 == 'y')
            {
                pre->next = p->next;
                free(p); // 释放空间
                return x;
                printf("删除学生成功!\n");
                break;
            }
            else
            {
                printf("没有删除该生!\n");
            }
        }
        pre = p;
        p = p->next;
    }
    if (judge == 0)
    {
        printf("\n该生不存在!按任意键继续..\n");
        getch();
        return x;
    }
    return x;
}

```



辅导员查询信息：

1) 通过学生姓名查询：

```
void findName(stu* y)
{
    char name[20];
    stu* p;
    if (y == NULL)
    {
        printf("\n请输入要查找学生的姓名: ");
        scanf("%s", name);
        printf("\n该生不存在! 按任意键继续..");
        getch();
        return;
    }
    printf("\n请输入要查找学生的姓名: ");
    scanf("%s", name);
    p = y;
    while (p)
    {
        if (strcmp(p->name, name) != 0)
            p = p->next;
        else
            break;
    }
    if (!p)
    {
        printf("\n该生不存在! 按任意键继续..");
        getch();
        return;
    }
    printf("\n你查找的学生信息如下:\n");
    printf("\n学号:%s 姓名:%s 年龄:%s 性别:%s 地址:%s 电话:%s 专业:%s 成绩:%d\n\n", p->number, p->name, p->age, p->sex, p->add, p->Tel, p->special, p->score);
    printf("\n按任意键继续..");
    getch();
}
```

2) 通过学生学号查询：

```
void findNum(stu* y)
{
    char num[20];
    stu* p;
    if (y == NULL)
    {
        printf("\n请输入要查找学生的学号: ");
        scanf("%s", num);
        printf("\n该生不存在! 按任意键继续..");
        getch();
        return;
    }
    printf("\n请输入要查找学生的学号: ");
    scanf("%s", num);
    p = y;
    while (p)
    {
        if (strcmp(p->number, num) != 0)
            p = p->next;
        else
            break;
    }
    if (!p)
    {
        printf("\n该生不存在! 按任意键继续..");
        getch();
        return;
    }
    printf("\n你查找的学生信息如下:\n");
    printf("\n学号:%s 姓名:%s 年龄:%s 性别:%s 地址:%s 电话:%s 专业:%s 成绩:%d\n\n", p->number, p->name, p->age, p->sex, p->add, p->Tel, p->special, p->score);
    printf("\n按任意键继续..");
    getch();
}
```

辅导员插入信息：

```
void insert(stu* x)
{
    int num, i;
    student* p, * q;
    p = x;

    printf("请输入你要插入位置: ");
    scanf("%d", &num);
    if (num > length) //大于链表长度时
    {
        printf("没有找到该位置. 按任意键继续..\n");
    }
    else
    {
        printf("请输入你要插入的学生的学号、姓名、年龄、性别、地址、电话、专业、成绩: \n");
        q = (stu*)malloc(N);
        scanf("%s %s %s %s %s %s %d", q->number, q->name, q->age, q->sex, q->add, q->Tel, q->special, &q->score);
        for (i = 0; i < num; i++)
            p = p->next;
        q->next = p->next;
        p->next = q;
        length++;
        print(x);
        printf("插入学生信息成功! \n");
        return;
    }
    getch();
}
```

辅导员显示信息：

```

void print(stu* p)
{
    if (p == NULL)
    {
        printf("\n该生不存在!按任意键继续...");
        getch();
        return;
    }
    p = p->next;
    while (p)
    {
        printf("\n学号:%s 姓名:%s 年龄:%s 性别:%s 地址:%s 电话:%s 专业:%s 成绩:%d\n\n", p->number, p->name, p->age, p->sex, p->add, p->Tel, p->special, p->score);
        p = p->next;
    }
    printf("\n按任意键继续...");
    getch();
}

```

辅导员成绩排序显示:

```

stu* sort(stu* x)
{
    stu* p1;
    stu* p2;
    char number[20]; // 学号
    char name[20];   // 姓名
    char age[20];    // 年龄
    char sex[20];    // 性别
    char add[20];    // 地址
    char Tel[20];    // 电话
    char special[20]; // 专业
    int score; // 成绩
    p1 = x;
    while (p1 != NULL)
    {
        p2 = p1->next;
        while (p2 != NULL)
        {
            if (p2->score < p1->score) // 完成数值交换
            {
                strcpy(number, p2->number);
                strcpy(name, p2->name);
                strcpy(age, p2->age);
                strcpy(sex, p2->sex);
                strcpy(add, p2->add);
                strcpy(Tel, p2->Tel);
                strcpy(special, p2->special);

                score = p2->score;

                strcpy(p2->number, p1->number);
                strcpy(p2->name, p1->name);
                strcpy(p2->age, p1->age);
                strcpy(p2->sex, p1->sex);
                strcpy(p2->add, p1->add);
                strcpy(p2->Tel, p1->Tel);
                strcpy(p2->special, p1->special);

                p2->score = p1->score;

                strcpy(p1->number, number);
                strcpy(p1->name, name);
                strcpy(p1->age, age);
                strcpy(p1->sex, sex);
                strcpy(p1->add, add);
                strcpy(p1->Tel, Tel);
                strcpy(p1->special, special);
                p1->score = score;
            }
            p2 = p2->next;
        }
        p1 = p1->next;
    }
    print(x);
    printf("\n学生信息排序结束!按任意键继续...\n");
    getch();
    return x;
}

```

学生查询信息:

```

int FindMenu()
{
    int a;
    system("cls"); // 清屏
    printf("===== \n");
    printf("1. 按学号 = \n");
    printf("2. 按姓名 = \n");
    printf("0. 退出 = \n");
    printf("===== \n");
    scanf("%d", &a);
    return a;
}

```

## 六、主程序

程序 1. 尾插法创建链表

程序 2. 冒泡排序

## 七、运行截图

学生信息管理系统主界面：

```
=====
==                               ==
==          欢迎来到学生信息管理系统主界面!!!          ==
==                               ==
==                               ==
==          1. 辅导员          ==
==          2. 学生            ==
==          0. 退出            ==
==                               ==
=====
```

选择不同的人群就会有不同的权限进入不同的界面，先进入辅导员界面对学生进行添加等操作。这里需要输入辅导员的账号和密码才能进入系统：

```
=====
==                               ==
==          欢迎来到学生信息管理系统主界面!!!          ==
==                               ==
==                               ==
==          1. 辅导员          ==
==          2. 学生            ==
==          0. 退出            ==
==                               ==
=====
1
请输入辅导员账号：
admin
请输入密码：
12345
```

辅导员界面：

```
=====
1. 添加学生信息
2. 修改学生信息
3. 删除学生信息
4. 查询学生信息
5. 插入学生信息
6. 显示学生信息
7. 成绩排序显示信息
0. 返回
=====
请输入您要执行的操作代码:
```

现在对学生信息进行添加:

```
请输入第1个学生的信息:
学号: 001
姓名:姜滢
年龄:21
性别:女
地址:合川
电话:177
专业:计算机科学与技术
成绩:98
添加成功!
是否继续?(y/n):
=====
请输入第2个学生的信息:
学号: 002
姓名:黄璐瑶
年龄:22
性别:女
地址:渝北
```

修改学生信息:

```
E:\zy\第一学期\数据结构\实验\期末\s1\Debug\s1.exe
1. 添加学生信息
2. 修改学生信息
3. 删除学生信息
4. 查询学生信息
5. 插入学生信息
6. 显示学生信息
7. 成绩排序显示信息
0. 返回
=====
请输入您要执行的操作代码: 2
请输入要修改学生的学号: 003
学生信息如下:
学号:003 姓名:周梦佳 年龄:y 性别:女 地址:綦江 电话:178 专业:计算机科学与技术 成绩:97

请选择要修改的项:
1学号 2 姓名 3 年龄 4 性别 5 地址 6 电话 7 专业 8 成绩
3
请输入修改后的内容:
22
修改后学生信息如下:
学号:003 姓名:周梦佳 年龄:22 性别:女 地址:綦江 电话:178 专业:计算机科学与技术 成绩:97
修改成功!
是否要继续修改?
是y 否n:
```

删除学生信息:

```
=====
1. 添加学生信息
2. 修改学生信息
3. 删除学生信息
4. 查询学生信息
5. 插入学生信息
6. 显示学生信息
7. 成绩排序显示信息
0. 返回
=====
请输入您要执行的操作代码: 3
请输入要删除学生的学号: 006
你要删除的信息如下, 请确认是否删除:
学号:006 姓名:李四 年龄:24 性别:男 地址:西安 电话:145 专业:金融 成绩:85

是y, 否n:
```

查询被删除的学生信息:

```
=====
=                               1. 按学号                               =
=                               2. 按姓名                               =
=                               0. 退出                                =
=====
1

请输入要查找学生的学号：006

该生不存在!按任意键继续..
```

查询未被删除的学生信息：

```
=====
=                               1. 按学号                               =
=                               2. 按姓名                               =
=                               0. 退出                                =
=====
1

请输入要查找学生的学号：001
您查找的学生信息如下：

学号:001 姓名:姜滢 年龄:21 性别:女 地址:合川 电话:177 专业:计算机科学与技术 成绩:98

按任意键继续..
```

显示所有的学生信息：

```
=====
1. 添加学生信息
2. 修改学生信息
3. 删除学生信息
4. 查询学生信息
5. 插入学生信息
6. 显示学生信息
7. 成绩排序显示信息
0. 返回
=====
请输入您要执行的操作代码: 6

学号:001 姓名:姜滢 年龄:21 性别:女 地址:合川 电话:177 专业:计算机科学与技术 成绩:98

学号:002 姓名:黄璐瑶 年龄:22 性别:女 地址:渝北 电话:155 专业:计算机科学与技术 成绩:97

学号:003 姓名:周梦佳 年龄:22 性别:女 地址:綦江 电话:178 专业:计算机科学与技术 成绩:97

学号:004 姓名:任婷婷 年龄:19 性别:女 地址:九龙坡 电话:156 专业:计算机科学与技术 成绩:87

按任意键继续..
```

对学生成绩进行就升序排序:

```
=====
1. 添加学生信息
2. 修改学生信息
3. 删除学生信息
4. 查询学生信息
5. 插入学生信息
6. 显示学生信息
7. 成绩排序显示信息
0. 返回
=====
请输入您要执行的操作代码: 7

学号:004 姓名:任婷婷 年龄:19 性别:女 地址:九龙坡 电话:156 专业:计算机科学与技术 成绩:87

学号:003 姓名:周梦佳 年龄:22 性别:女 地址:綦江 电话:178 专业:计算机科学与技术 成绩:97

学号:002 姓名:黄璐瑶 年龄:22 性别:女 地址:渝北 电话:155 专业:计算机科学与技术 成绩:97

学号:001 姓名:姜滢 年龄:21 性别:女 地址:合川 电话:177 专业:计算机科学与技术 成绩:98

按任意键继续..
```

进入学生界面:

```
=====
1. 查询学生信息
0. 返回
=====
请输入您要执行的操作代码:
```

对学生信息进行查询:

```
=====
1. 按学号
2. 按姓名
0. 退出
=====
1
请输入要查找学生的学号: 002
您查找的学生信息如下:
学号:002 姓名:黄璐瑶 年龄:22 性别:女 地址:渝北 电话:155 专业:计算机科学与技术 成绩:97
按任意键继续..
```

```
=====
1. 按学号
2. 按姓名
0. 退出
=====
2
请输入要查找学生的姓名: 任婷婷
你查找的学生信息如下:
学号:004 姓名:任婷婷 年龄:19 性别:女 地址:九龙坡 电话:156 专业:计算利科学与技术 成绩:87
按任意键继续..
```

## 八、总结

这次课程设计的心得体会通过实习我的收获如下:

- 1、巩固和加深了对数据结构的理解,提高综合运用本课程所学知识的能力。
- 2、培养了我选用参考书,查阅手册及文献资料的能力。培养独立思考,深入研究,分析问题、解决问题的能力。
- 3、通过实际编译系统的分析设计、编程调试,掌握应用软件的分析方法和工程设计方法。



4、通过课程设计，培养了我严肃认真的工作作风，逐步建立正确的生产观念、经济观念和全局观念。从刚开始得觉得很难，到最后把这个做出来，付出了很多，也得到了很多，以前总以为自己对编程的地方还不行，现在，才发现只要认真做，没有什么不可能。

编程时要认真仔细，出现错误要及时找出并改正，（其中对英语的要求也体现出来了，因为它说明错误的时候都是英语）遇到问题要去查相关的资料。反复的调试程序，最好是多找几个同学来对你的程序进行调试并听其对你的程序的建议，在他们不知道程序怎么写的时候完全以一个用户的身份来用对你的用户界面做一些建议，正所谓当局者迷旁观者清，把各个注意的问题要想到；同时要形成自己的编写程序与调试程序的风格，从每个细节出发，不放过每个知识点，注意与理论的联系和理论与实践的差别。另外，要注意符号的使用，注意对字符处理，特别是对指针的使用很容易出错且调试过程是不会报错的，那么我们要始终注意指针的初始化不管它怎么用以免不必要麻烦。

## 九、参考文献

C 语言程序设计-----科学出版社

数据结构（C 语言描述）-----清华大学出版社

数据结构（使用 C 语言）-----电子科技大学出版社

## 十、附录：源代码（main.app）

```
#include "stu.h"

int main()
{
    int nChoose1;

    int nChoose2;

    int nChoose3;

    int nChoose4;

    int nEnd = 1;

    int nEnd2 = 1;
```

```

int nEnd3 = 1;

char sName[20];

char sPassword[20];

//choice 用于选择对数据执行操作, i 循环, a 主界面选择//

stu* x;

x = NULL;

while (nEnd)
{

    nChoose1 = MainMenu();

    switch (nChoose1)
    {

    case 0:

        nEnd = 0;

        sort(x);                //排序

        outfile(x);              //将链表内容写入文件

        system("cls");          //清屏

        printf("          \n          谢      谢      使      用      !!!\n");

        break;

    case 1:

        printf("请输入指导员账号: \n");

        scanf("%s", sName);

        printf("请输入密码: \n");

```

```

scanf("%s", sPassword);

if (strcmp(sName, "admin") != 0)
{
    printf("账号错误! \n");
}

else
{
    if (strcmp(sPassword, "12345") != 0)
    {
        printf("密码错误! \n");
    }

    else
    {
        nEnd2 = 1;

        while (nEnd2)
        {
            nChoose2 = ManageMenu();

            switch (nChoose2)
            {

            case 1:

                x = creatlink(x);

                break;

            case 2:

                x = modify(x);

```

```

        break;

case 3:

    x = delet(x);

    break;


case 4:

    nChoose4 = FindMenu();

    switch (nChoose4)
    {

case 1:

        findNum(x);

        break;

case 2:

        findName(x);

        break;

default:

        printf(" 选项输入不正确, 无法执行.\n");

        break;

    }

    break;

case 5:

    insert(x);

    break;

case 6:

```

```

        print(x);

        break;

    case 7:

        x = sort(x);

        break;

    case 0:

        nEnd2 = 0;

        break;

    default:

        printf(" 选项输入不正确，无法执行。\\n");

        break;

    }

}

}

break;

case 2:

    nEnd3 = 1;

    while (nEnd3)

    {

        nChoose3 = StudentMenu();

        switch (nChoose3)

        {

            case 0:

```

```

        nEnd3 = 0;

        break;

case 1:

    nChoose4 = FindMenu();

    switch (nChoose4)
    {

case 1:

        findNum(x);

        break;

case 2:

        findName(x);

        break;

default:

        printf(" 选项输入不正确，无法执行。 \n");

        break;

    }

    break;

default:

    printf("选项输入不正确，无法执行。 \n");

    break;

    }

    }

    break;

default:

```

```
        printf("选项输入不正确，无法执行。\\n");  
        break;  
    }  
}  
  
return 0;  
}
```