

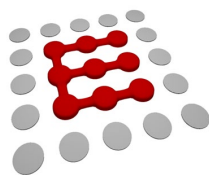


北京理工大学  
Beijing Institute of Technology

# 本科实验报告

实验名称： 图书管理

课程名称：	数据结构与算法设计实验	实验时间：	2017/4/21
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

有一个小型书库保管了大量图书，关于图书有大量信息需要处理，这些信息包括图书的分类、书名、作者名、购买日期、价格等。现要求编写一个程序以便于对图书的管理。

## 2. 实验题目

- a. 建立图书信息。
- b. 提供想找功能，按照多种关键字查找需要的书籍。例如按书名查找，输入书名后，将显示出该图书的所有信息，或显示指定信息。
- c. 提供排序功能，按照多种关键字对所有的书籍进行排序，例如按出版日期进行排序。
- d. 提供维护功能，可以对图书信息进行添加、修改、删除等功能。

## 3. 实验基础知识

单链表的建立，strchr 等字符处理函数的应用，冒泡法排序与单链表的结合，链表与文件的结合。

## 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 book { //图书信息结构体定义
    char cate[10]; //类别
    char book[25]; //书名
    char author[25]; //作者名
    char num[25]; //图书编号
    int date[3]; //出版日期（年 月 日）
    float price; //书的价格
    struct book * next;
};
typedef struct book BOOK; 全局变量
```

- 2) 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
```

```

void Login_MFace();//登录界面
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(BOOK *q);//按（作者，书名，编号）所含关键字查找，可在
    结果中继续查找
void S_Date();//按日期查找
void display(BOOK * p);//将 p 指针所指图书信息显示出来
void Sort_Data();//排序函数
void Sort_num();//按照图书编号排序
void Sort_Cate();//按照图书类别排序
void Sort_Date();//按照图书日期排序
void Sort_Book();//按照书名排序
void Modify_Data();//修改信息函数
void Modify_key(int j);//按照关键字（书名或作者名）检索修改
void Delete();//删除某本图书的信息
void PRINT(BOOK * p);//输入 p 指针所指图书的所有信息

```

#### 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\t");
    printf("1-----显示全部图书信息\n\t\t");
    printf("2-----查询图书信息\n\t\t");
    printf("3-----排序\n\t\t");
    printf("4-----修改图书信息\n\t\t");
    printf("5-----删除图书信息\n\t\t");
    printf("6-----添加图书信息\n\t\t");
    printf("7-----保存信息\n\t\t");
    printf("0-----退出\n\t\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) 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。

- 5) 在进行关键字查找的时候，把查询到的结果储存到一个新的链表中，如果继续查找，则通过对此链表进行查找，一步一步缩小查找范围。

## 6. 测试结果

- 1) 验证身份

```
欢迎使用图书管理系统：

          用户名：000000

          密  码：000000

          登陆成功！

请按任意键继续. . .
```

- 2) 录入信息

```
请选择数据输入方式：
1->手动输入
2->文件导入
1

          输入图书信息
          请输入第1本图书的信息：

编号：9867234
书名：远去的故乡
作者：王亚静
价格：13.14
日期（年月日以空格隔开）：2016 3 20
图书类别：爱情
是否继续录入？(Y/N)
Y
```

- 3) 文件中读取信息

```
请选择数据输入方式：
1->手动输入
2->文件导入
2

文件读取成功！

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-----显示全部图书信息
2-----查询图书信息
3-----排序
4-----修改图书信息
5-----删除图书信息
6-----添加图书信息
7-----保存信息
0-----退出
```

- 5) 显示全部图书信息

4-----修改图书信息  
5-----删除图书信息  
6-----添加图书信息  
7-----保存信息  
0-----退出  
1

全部图书信息为:  

编号	书名	作者	价格	出版日期	图书类别
8887721	数学和英语的联系	无妄之灾	66.20	2014年5月13日	英语
7877234	数学的奥秘	田桂芳	66.20	1999年12月8日	数学
1414876	高等数学	马建忠	77.00	2014年3月18日	数学
9278652	以往的故乡	十年	46.20	2011年3月3日	文学
26723322	守望的稻草人	七秒钟的记忆	77.20	2005年3月23日	文学
9867234	远去的故乡	王亚静	13.14	2016年3月20日	爱情

请按任意键继续. . .

6) 查询信息（按关键字查找）

请输入关键字:  
数学

编号	书名	作者	价格	出版日期	图书类别
8887721	数学和英语的联系	无妄之灾	66.20	2014年5月13日	英语
7877234	数学的奥秘	田桂芳	66.20	1999年12月8日	数学
1414876	高等数学	马建忠	77.00	2014年3月18日	数学

是否在结果中继续查找:  
1. 是  
2. 否

7) 在查询结果中继续查找

请输入关键字:  
英语

编号	书名	作者	价格	出版日期	图书类别
8887721	数学和英语的联系	无妄之灾	66.20	2014年5月13日	英语

是否在结果中继续查找:  
1. 是  
2. 否

8) 查询信息（按日期查找）

请输入日期（年月日以空格隔开）: 2014 5 13

编号	书名	作者	价格	出版日期	图书类别
8887721	数学和英语的联系	无妄之灾	66.20	2014年5月13日	英语

是否继续查找:  
1. 是  
2. 否

9) 排序菜单

排序方式为:

- 按照书名排序
- 按照图书类别排序
- 按照编号进行排序
- 按照出版日期排序

10) 排序结果（书名排序）

全部图书信息为:

编号	书名	作者	价格	出版日期	图书类别
1414876	高等数学	马建忠	77.00	2014年3月18日	数学
26723322	守望的稻草人	七秒钟的记忆	77.20	2005年3月23日	文学
7877234	数学的奥秘	田桂芳	66.20	1999年12月8日	数学
8887721	数学和英语的联系	无妄之灾	66.20	2014年5月13日	英语
9278652	以往的故乡	十年	46.20	2011年3月3日	文学
9867234	远去的故乡	王亚静	13.14	2016年3月20日	爱情

请按任意键继续. . .

#### 11) 排序结果（类别排序）

全部图书信息为:

编号	书名	作者	价格	出版日期	图书类别
9867234	远去的故乡	王亚静	13.14	2016年3月20日	爱情
1414876	高等数学	马建忠	77.00	2014年3月18日	数学
7877234	数学的奥秘	田桂芳	66.20	1999年12月8日	数学
26723322	守望的稻草人	七秒钟的记忆	77.20	2005年3月23日	文学
9278652	以往的故乡	十年	46.20	2011年3月3日	文学
8887721	数学和英语的联系	无妄之灾	66.20	2014年5月13日	英语

请按任意键继续. . .

#### 12) 排序结果（编号排序）

全部图书信息为:

编号	书名	作者	价格	出版日期	图书类别
1414876	高等数学	马建忠	77.00	2014年3月18日	数学
26723322	守望的稻草人	七秒钟的记忆	77.20	2005年3月23日	文学
7877234	数学的奥秘	田桂芳	66.20	1999年12月8日	数学
8887721	数学和英语的联系	无妄之灾	66.20	2014年5月13日	英语
9278652	以往的故乡	十年	46.20	2011年3月3日	文学
9867234	远去的故乡	王亚静	13.14	2016年3月20日	爱情

请按任意键继续. . .

#### 13) 排序结果（日期排序）

全部图书信息为:

编号	书名	作者	价格	出版日期	图书类别
9867234	远去的故乡	王亚静	13.14	2016年3月20日	爱情
8887721	数学和英语的联系	无妄之灾	66.20	2014年5月13日	英语
1414876	高等数学	马建忠	77.00	2014年3月18日	数学
9278652	以往的故乡	十年	46.20	2011年3月3日	文学
26723322	守望的稻草人	七秒钟的记忆	77.20	2005年3月23日	文学
7877234	数学的奥秘	田桂芳	66.20	1999年12月8日	数学

请按任意键继续. . .

#### 14) 修改信息菜单

请提供您要修改的图书的信息:

1. 书名
2. 作者名
3. 编号

#### 15) 修改信息（作者名为例）

```

您要修改的图书信息为：
编号      书名      作者      价格      出版日期      图书类别
9867234    远去的故乡    王亚静    13.14    2016年3 月20日    爱情

可修改的项目有：
1. 编号      5. 日期
2. 书名      6. 类别
3. 作者      7. 以上全部
4. 价格

请输入修改项目： 1

请输入要修改的图书的新的数据：
请输入新的编号： 59211314

是否继续修改：
1. 继续修改
2. 退出修改
您的选择是： 2
请按任意键继续. . .

```

#### 16) 删除信息

```

请输入要删除的图书的编号：
26723322
您要删除的图书数据为：
编号      书名      作者      价格      出版日期      图书类别
26723322    守望的稻草人    七秒钟的记忆    77.20    2005年3 月23日    文学

删除成功！

是否继续删除：
1. 是
2. 否
-

```

#### 17) 增加图书信息

```

编号： 5981332
书名： 老师你好漂亮
作者： 十年之后
价格： 99.99
日期（年月日以空格隔开）： 2017 5 1
图书类别： 搞怪

插入成功！

是否继续插入？（Y/N）

```

#### 18) 保存信息

```

请根据提示选择相应的操作：
1-----显示全部图书信息
2-----查询图书信息
3-----排序
4-----修改图书信息
5-----删除图书信息
6-----添加图书信息
7-----保存信息
0-----退出
7

保存成功！
请按任意键继续. . .

```

#### 19) 退出时提示是否要保存

```

请根据提示选择相应的操作：
1-----显示全部图书信息
2-----查询图书信息
3-----排序
4-----修改图书信息
5-----删除图书信息
6-----添加图书信息
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. . .

-----
Process exited after 271 seconds with return value 0
请按任意键继续. . .

```

#### 7. 附录（源代码）

```

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

struct book { //图书信息结构体定义
    char cate[10]; //类别
    char book[25]; //书名
    char author[25]; //作者名
    char num[25]; //图书编号
    int date[3]; //出版日期（年 月 日）
    float price; //书的价格
    struct book * next;
};

```



```
typedef struct book BOOK;
```

```
BOOK *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 Insert_Data();//插入函数
```

```
void Input_Data();//录入图书信息
```

```
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(BOOK *q);//按（作者，书名，编号）所含关键字查找，可在  
结果中继续查找
```

```
void S_Date();//按日期查找
```

```
void display(BOOK * p);//将 p 指针所指图书信息显示出来
```

```
void Sort_Data();//排序函数
```

```
void Sort_num();//按照图书编号排序
```

```
void Sort_Cate();//按照图书类别排序
```

```
void Sort_Date();//按照图书日期排序
```

```
void Sort_Book();//按照书名排序
```

```
void Modify_Data();//修改信息函数
```

```
void Modify_key(int j);//按照关键字（书名或作者名）检索修改
```

```
void Delete();//删除某本图书的信息
```

```
void PRINT(BOOK * p);//输入 p 指针所指图书的所有信息
```

```
int main() {
```

```
    BOOK * j;
```

```
    int m,n;
```

```
    int flag = 1;
```

```
    int choice;
```

```
    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 欢迎使用图书管理系统： \n");
```

```
    Pass();
```

```
    head = NULL;
```

```
    system("cls");
```

```
    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);
```

```
    while (n != 1 && n != 2) {
```

```
        printf("\n 输入有误，请重新输入（1-2）： ");
```

```

scanf("%d", &n);
}
if (n == 1) Input_Data();
if (n == 2) Up_Data();

fflush(stdin);
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 全部图书信息为: \n");
        printf("编号          书名          作者\n");
        printf("出版日期          图书类别\n");
        for (j = head; j != NULL; j = j->next)
            display(j);
        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\n"); 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-----显示全部图书信息\n\t\t");
    printf("2-----查询图书信息\n\t\t");
    printf("3-----排序\n\t\t");
    printf("4-----修改图书信息\n\t\t");
    printf("5-----删除图书信息\n\t\t");
    printf("6-----添加图书信息\n\t\t");
    printf("7-----保存信息\n\t\t");
    printf("0-----退出\n\t\t");
    return;
}

void Insert_Data() {
    system("cls");

```

```

char ch;
int flag = 1;
BOOK *p;
p = head;
printf("\n 请按提示输入新入图书的信息: \n");
while (flag) {
    system("cls");
    p = (BOOK *)malloc(sizeof(BOOK));
    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
    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");
    fflush(stdin);
    scanf("%c", &ch);
    while (ch != 'Y' && ch != 'N') {
        printf("\n 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 Input_Data() {
    int n = 1;//判断输入的是第几本图书
    int flag = 1;//循环条件
    char ch;
    BOOK * p1;
    p1 = NULL;
    head = NULL;

```

```

head = (BOOK *)malloc(sizeof(BOOK)); //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输入图书信息\n\t\t请输入第%d本图书的信息:\n",
n);
        PRINT(head);

    }
    else {
        p1 = (BOOK *)malloc(sizeof(BOOK));
        printf("\t\t输入图书信息\n\t\t请输入第%d本图书的信息:\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 Save_Data() {

```

```

FILE * fp;
BOOK * p;
if ((fp = fopen("D: \\BookData.txt", "w")) == NULL) {
    printf("\n 无法打开文件\n");
    exit(1);
}
p = head;
do {
    fprintf(fp, "%12s %20s %12s %10.2f %4d %2d %2d    %10s\n",
p->num, p->book, p->author, p->price, p->date[0], p->date[1], p->date[2],
p->cate);
    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, "%12s %20s %12s %10.2f %4d %2d %2d    %10s",
p->num, p->book, p->author, p->price, p->date[0], p->date[1], p->date[2],
p->cate);
    printf("\n 保存成功! \n");
    fclose(fp);
}

void Up_Data() {
    int n = 1;
    char ch;
    FILE * fp;
    BOOK * p, *q;
    if ((fp = fopen("D: \\BookData.txt", "r")) == NULL) {
        printf("\n 无法打开文件\n");
        exit(1);
    }
    p = (BOOK*)malloc(sizeof(BOOK));
    p->next = NULL;
    while (!feof(fp)) {
        if (n == 1) {
            fscanf(fp, "%s%s%s%f%d%d%d%s", p->num, p->book,
p->author, &p->price, &p->date[0], &p->date[1], &p->date[2], p->cate);
        }
        else {
            q = (BOOK*)malloc(sizeof(BOOK));
            fscanf(fp, "%s%s%s%f%d%d%d%s", q->num, q->book,
q->author, &q->price, &q->date[0], &q->date[1], &q->date[2], q->cate);

```

```

        q->next = p;
        p = q;
    }
    n += 1;
}
head = p;
fclose(fp);
printf("\n\n 文件读取成功! \n\n");
system("pause");
system("cls");
}

void Search_Infor() {
    int choice;
    system("cls");
    printf("请选择查找方式为: \n");
    printf("\t\t1.通过关键字查找\n");
    printf("\t\t2.通过日期查找\n");
    scanf("%d", &choice);
    while (choice != 1 && choice != 2) {
        printf("\n 输入有误, 请重新输入 (1-2): ");
        fflush(stdin); //清除缓存区
        scanf("%d", &choice);
    }
    switch (choice) {
        case 1: S_key(head); break;
        case 2: S_Date(); break;
    }
    return;
}

void S_Date() {
    system("cls");

    BOOK *p;
    p = NULL;
    int y, m, d;
    int n = 0; //记录查找到的图书的本数
    int choice;
    printf("\n 请输入日期 (年月日以空格隔开): ");
    scanf("%d%d%d", &y, &m, &d);

```

```

p = head;//无表头节点的链表
if (p == NULL) {
    printf("\n 空表，无图书数据！");
}
else {

}

while (p != NULL) {
    if (p->date[0] == y&& p->date[1] == m&& p->date[2] == d) {
        if (n == 0) {
            printf("编号          书名          作者
价格      出版日期          图书类别\n");
        }
        display(p);
        n++;
    }
    p = p->next;
}

if (n == 0) {
    printf("\n 无此图书的信息\n");
}

printf("\n 是否继续查找： \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）：");
    fflush(stdin);
    scanf("%d", &choice);
}
switch (choice) {
case 1: Search_Infor(); break;
case 2: return;
}
return;
}

void S_key(BOOK * q) {
    system("cls");
    BOOK *p, *r, *head1;
    head1 = NULL;//初始化操作
    p = NULL;

```



```

char key[20];
int choice;
int n = 0; //记录查找到的图书的数目
printf("请输入关键字: \n");
scanf("%s", key);
p = q; //无表头节点的链表
if (p == NULL) {
    printf("\n 空表, 无图书数据! ");
}
while (p != NULL) {
    if (strstr(p->num, key) != NULL || strstr(p->book, key) != NULL ||
    strstr(p->author, key) != NULL) {
        r = (BOOK *)malloc(sizeof(BOOK));
        *r = *p;
        if (n == 0) {
            printf("编号          书名          作者\n");
            printf("出版日期          图书类别\n");
            r->next == NULL;
            head1 = r;
        }
        else {
            r->next = head1;
            head1 = r;
        }
        n += 1;
        display(p);
    }
    p = p->next;
}

if (n == 0) {
    printf("\n 无此图书的信息\n");
}
printf("是否在结果中继续查找: \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): ");
    fflush(stdin);
    scanf("%d", &choice);
}

```

```

        switch (choice) {
            case 1:S_key(head1);break;
            case 2:return;
        }
        return;
    }

void display(BOOK * p) {
    printf("%-12s%-20s%-12s%-10.2f%-4d 年%-2d 月%-2d
    日    %-10s\n", p->num, p->book, p->author, p->price, p->date[0],
    p->date[1], p->date[2], p->cate);
    return;
}

void Sort_Data() {
    int choice;
    system("cls");
    printf("排序方式为: \n");
    printf("\t\t1.按照书名排序\n");
    printf("\t\t2.按照图书类别排序\n");
    printf("\t\t3.按照编号进行排序\n");
    printf("\t\t4.按照出版日期排序\n");
    scanf("%d", &choice);
    while (choice < 1 || choice>4) {
        printf("\n 输入有误, 请重新输入 (1-4): ");
        scanf("%d", &choice);
    }
    switch (choice) {
        case 1:Sort_Book();break;
        case 2:Sort_Cate();break;
        case 3:Sort_num();break;
        case 4:Sort_Date();break;
    }
    printf("\n 排序成功\n\n");
}

void Sort_Book() {
    BOOK *endpt;    //控制循环比较
    BOOK *p;        //临时指针变量
    BOOK *p1, *p2;
    p1 = (BOOK *)malloc(sizeof(BOOK));
    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->book, p1->next->next->book) > 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_num() {
    BOOK *endpt;    //控制循环比较
    BOOK *p;        //临时指针变量
    BOOK *p1, *p2;
    p1 = (BOOK *)malloc(sizeof(BOOK));
    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_Cate() {
    BOOK *endpt;      //控制循环比较
    BOOK *p;          //临时指针变量
    BOOK *p1, *p2;
    p1 = (BOOK *)malloc(sizeof(BOOK));
    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->cate, p1->next->next->cate) > 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_Date() {
    BOOK *endpt;    //控制循环比较
    BOOK *p;        //临时指针变量
    BOOK *p1, *p2;

    p1 = (BOOK *)malloc(sizeof(BOOK));
    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->date[0] < p1->next->next->date[0]
            ||(p1->next->date[0] == p1->next->next->date[0]&& p1->next->date[1] <
            p1->next->next->date[1]) ||(p1->next->date[0] == p1->next->next->date[0]
            && p1->next->date[1] == p1->next->next->date[1]&& p1->next->date[2]
            <p1->next->next->date[2]))
            { //如果前面的节点键值比后面节点的时间小,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("请提供您要修改的图书的信息: \n");
    printf("\t\t1.书名\n");
    printf("\t\t2.作者名\n");
}

```

```

printf("\t\t3.编号\n");
scanf("%d", &choice);
printf("\n");
while (choice < 1 || choice>3) {
    fflush(stdin);
    printf("输入有误，请重新输入（1-3）： ");
    scanf("%d", &choice);
}
switch (choice) {
case 1:Modify_key(1);break;
case 2:Modify_key(2);break;
case 3:Modify_key(3);break;
}
}

void Modify_key(int j) {
    BOOK *p;
    char key[20];
    int choice;
    int i;
    if (j == 1) {
        printf("请输入书名： \n");
    }
    else if (j == 2) {
        printf("请输入作者名： \n");
    }
    else {
        printf("请输入编号： \n");
    }

    scanf("%s", key);
    p = head;
    if (p == NULL) {
        printf("空表，无图书数据！ ");
    }
    if (j == 1) {
        while (p != NULL&&strcmp(p->book, key)) {//不考虑重复的情
况
            p = p->next;
        }
    }
    else if (j == 2) {

```

```

        while (p != NULL&&strcmp(p->author, key)) {//不考虑重复的情况
            p = p->next;
        }
    }
    else {
        while (p != NULL&&strcmp(p->num, key)) {//不考虑重复的情况
            p = p->next;
        }
    }

    if (p == NULL) {
        printf("\n 找不到此图书! ");
    }
    else
    {
        system("cls");
        printf("您要修改的图书信息为: \n");
        printf("编号          书名          作者          价格\n");
        printf("出版日期          图书类别\n");
        display(p);
        printf("\n 可修改的项目有: \n");
        printf("1.编号\t\t5.日期\t\n");
        printf("2.书名\t\t6.类别\t\n");
        printf("3.作者\t\t7.以上全部\t\n");
        printf("4.价格\t\t\n");
        printf("请输入修改项目: ");
        scanf("%d", &i);
        while (i < 1 || i > 7) {
            printf("\n 输入有误, 请重新输入 (1-7): ");
            scanf("%d", &i);
        }
        printf("\n 请输入要修改的图书的新的数据: \n");
        switch (i) {
            case 1:printf("请输入新的编号: ");
                    scanf("%s", p->num);
                    break;
            case 2:printf("请输入新的书名: ");
                    scanf("%s", p->book);
                    break;
            case 3:printf("请输入新的作者名: ");
                    scanf("%f", p->author);

```

```

        break;
    case 4:printf("请输入新的价格: ");
        scanf("%f", &p->price);
        break;
    case 5:printf("请输入新的日期(年月日以空格隔开): ");
        scanf("%d%d%d", &p->date[0], &p->date[1], &p->date[2]);
        break;
    case 6:printf("请输入新的类别: ");
        scanf("%s", p->cate);
        break;
    case 7: PRINT(p);
        break;
    }
}
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;

}
return;
}

void Delete() {
    system("cls");
    BOOK *p, *q;
    char num[20];
    int choice;
    printf("请输入要删除的图书的编号: \n");
    scanf("%s", num);
    p = head;//无表头节点的链表
    if (p == NULL) {
        printf("\n 空表, 无图书数据! ");
    }
}

```



```

else if (!strcmp(p->num, num)) {
    printf("您要删除的图书数据为: \n");
    printf("编号      书名      作者      价格\n");
    printf("出版日期      图书类别\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("找不到此图书! ");
    }
    else
    {
        printf("您要删除的图书数据为: \n");
        printf("编号      书名      作者\n");
        printf("价格      出版日期      图书类别\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(BOOK * p) {  
    printf("编号: ");  
    scanf("%s", p->num);  
    printf("书名: ");  
    scanf("%s", p->book);  
    printf("作者: ");  
    scanf("%s", p->author);  
    printf("价格: ");  
    scanf("%f", &p->price);  
    printf("日期（年月日以空格隔开）: ");  
    scanf("%d%d%d", &p->date[0], &p->date[1], &p->date[2]);  
    printf("图书类别: ");  
    scanf("%s", p->cate);  
  
}
```