《C语言程序设计》课程设计报告

**设计题目** 图书信息管理系统

**班 级**

**学生姓名**

**学 号**

**指导教师**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **《C语言程序设计》课程设计考核评估** | | | | | |
| 序号 | 项目名称 | 权重 | 得分 | | |
|  | | |
|  |  |  |
| 1 | 课设报告（规范性、完整性、合理性） | 30% |  |  |  |
| 2 | 程序实现（数据结构、编程规范、算法合理） | 30% |  |  |  |
| 3 | 演示与验收（界面友好、功能正确、极限测试） | 20% |  |  |  |
| 4 | 陈述答辩（思路清晰度、熟练度，回答正确度） | 20% |  |  |  |
|  | 课设总分 |  |  |  |  |

**验收教师签名：**

年 月

**目录**

[1 课题内容及要求 3](#_Toc30495)

[1.1 总体描述 3](#_Toc16949)

[1.2 功能要求 3](#_Toc6590)

[2 总体设计 3](#_Toc12983)

[3 详细设计 4](#_Toc27457)

[3.1 链表结点结构 4](#_Toc20653)

[3.2 链表 4](#_Toc26367)

[3.3 函数列表 5](#_Toc13626)

[3.4 主要功能模块设计（流程图） 6](#_Toc10275)

[3.4.1 Book \*read\_from\_file() 6](#_Toc17167)

[3.4.3 void input\_book() 6](#_Toc12806)

[3.4.5 void revise\_book() 7](#_Toc5598)

[3.4.7 void find\_name\_book() void find\_author\_book() 7](#_Toc18509)

[void find\_ISBN\_book() void find\_publish\_book() 7](#_Toc1680)

[3.5 算法实现 8](#_Toc15555)

[3.5.1 插入法实现链表排序 8](#_Toc12833)

[3.5.2 其余算法实现 8](#_Toc2388)

[4 课程设计小结 9](#_Toc8258)

[5 参考资料 9](#_Toc32404)

[6 附录 9](#_Toc20536)

1 课题内容及要求

1.1 总体描述

设计一个简单的图书信息管理系统，可用于个人、家庭等小型书库管理。本程序在Visual Studio 2017开发环境下运行。其主要功能是实现书籍信息的读取、添加、删除、排序、存储等。这是一个简单、大众的管理系统，功能简要，操作简单，占用资源少，方便了小型书库管理。

1.2 功能要求

设计和实现一个“图书信息管理系统”，满足以下要求：

1.系统以菜单方式工作；

2.使用链表或结构数组对图书信息进行管理和维护；

3.使用二进制文件在磁盘上保存图书记录信息；**（可选）**

4.链表中各结点或结构数组中各元素包括“图书编号、书名、作者、出版社、出版时间、价格”基本字段；

5.实现如下基本功能：

(1)图书信息录入功能

(2)图书信息浏览功能 （输出所有的图书信息）

(3)图书信息排序 （按图书编号由大到小的顺序排列）

(4)图书信息查询 （按图书编号和作者两种方式查询图书信息）

(5)图书信息统计 （按出版社统计满足条件的图书数目，并显示图书信息）

(6)图书删除 （删除指定编号的图书）

(7)图书修改 （修改指定编号的图书的基本信息）

(8)将图书信息保存到文件存盘 （**可选，**将链表或结构数组的数据以文件的形式存盘）

(9)从文件中读入图书信息 （**可选，**将原来已经存盘的文件读入内存，进行管理）

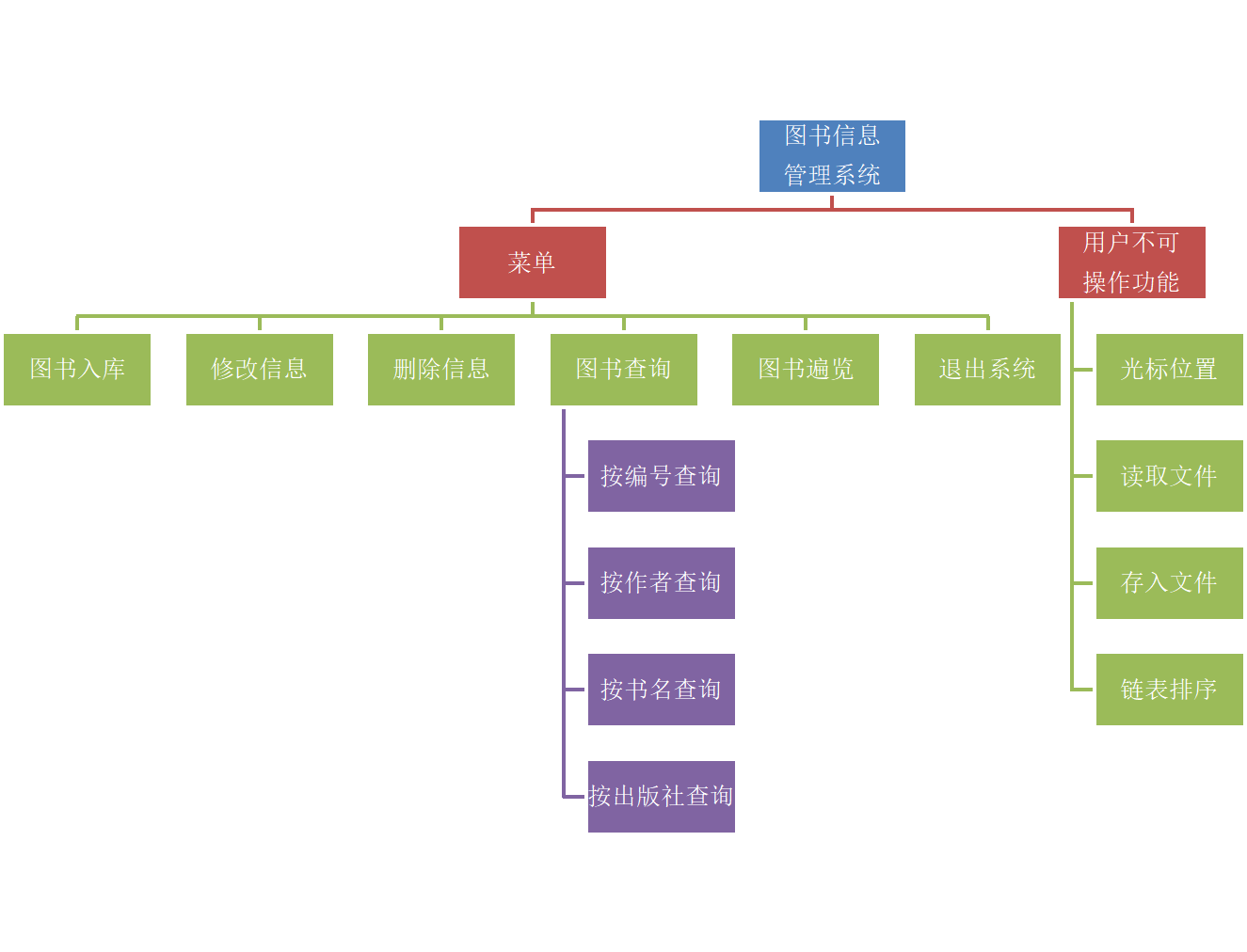
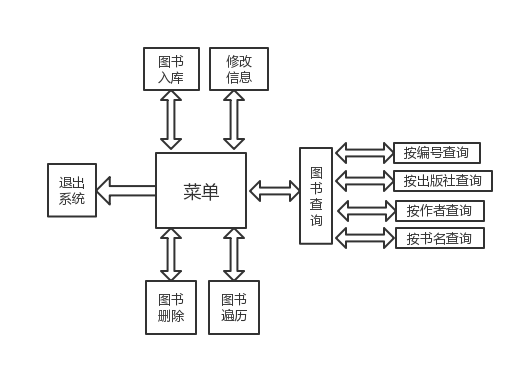
2 总体设计图 1 功能模块图

图 2 结构框图



3 详细设计

3.1 链表结点结构

struct Book {

char ISBN[11]; //图书编号

char name[12]; //书名

char author[12]; //作者

char type[10]; //书籍分类

char publish[12]; //出版社

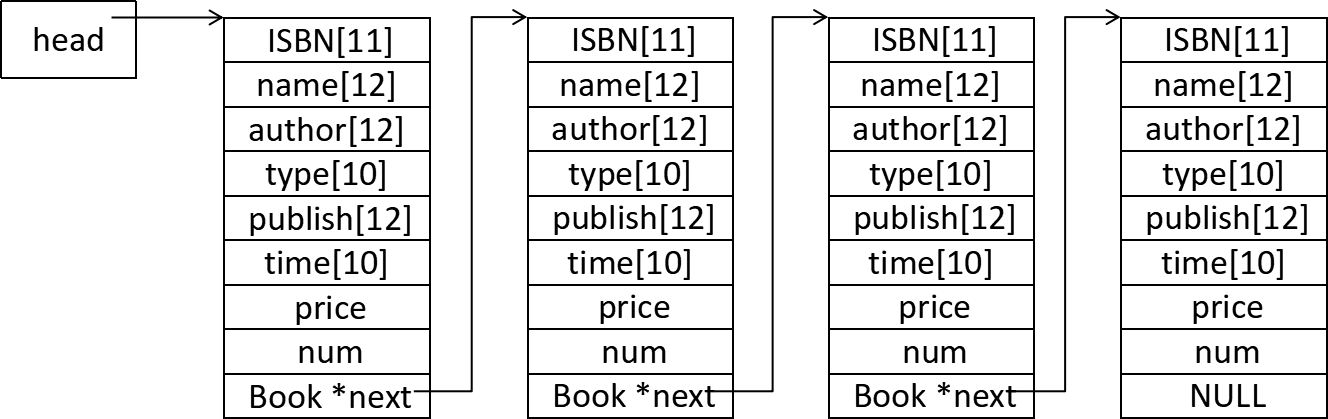
char time[10]; //出版时间

float price; //价格

int num; //数量

struct Book \*next; //指针域

};

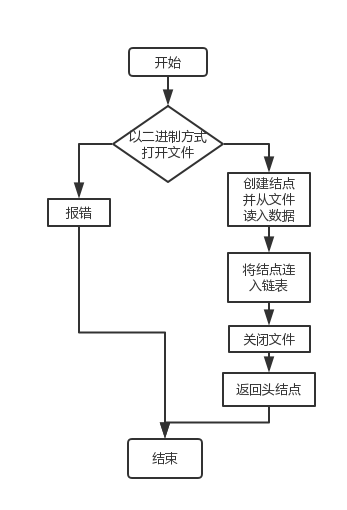
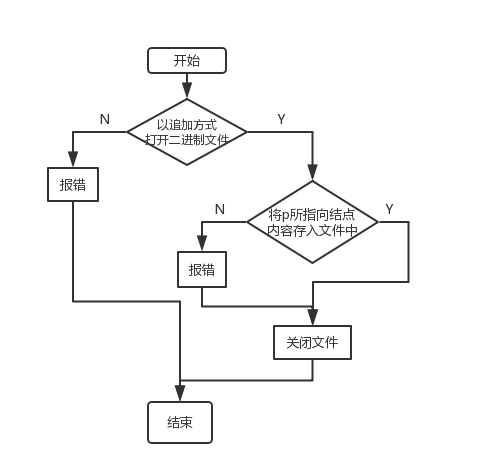
3.2 链表

3.3 函数列表

|  |  |  |
| --- | --- | --- |
| 序号 | 函数原型 | 函数简要说明 |
| 1 | int main() | 主函数，只包含一个menu()菜单 |
| 2 | void menu() | 菜单函数，主体是一个switch函数，可选择功能并跳转到各功能函数 |
| 3 | void loca(int x, int y) | 光标定位 |
| 4 | Book \*read\_from\_file() | 从目标文件中读取数据，形成完整链表，返回表头 |
| 5 | void save\_to\_file(Book \*p) | 将链表存入目标文件。读一条记录存一条，顺序写入。只存入数据区内容，空出指针区以便读取形成链表时赋予指针 |
| 6 | void find\_book() | 选择查找方式（编号、作者、书名、出版社），主体是switch函数 |
| 7 | void input\_book() | 输入数据。动态分配内存，向其中输入一条记录，然后调用save\_to\_file(Book \*p)存储入文件，添加在文件末尾 |
| 8 | void print\_book() | 遍历链表，输出全部数据区数据 |
| 9 | void del\_book() | 删除数据，本质是链表删除结点。读取文件形成无序链表，删除符合条件的结点 |
| 10 | void revise\_book() | 修改数据，本质是链表修改结点数据。读取文件形成无序链表，修改符合条件的结点的数据。 |
| 11 | void find\_name\_book() | 书名查找，读取文件后，遍历完整链表，符合条件则输出，并计算符合条件的图书总数 |
| 12 | void find\_author\_book() | 作者查找，读取文件后，遍历完整链表，符合条件则输出，并计算符合条件的图书总数 |
| 13 | void find\_ISBN\_book() | 图书编号查找，读取文件后，遍历完整链表，符合条件则输出，并计算符合条件的图书总数 |
| 14 | void find\_publish\_book() | 出版社查找，读取文件后，遍历完整链表，符合条件则输出，并计算符合条件的图书总数 |
| 15 | Book \*sort\_book(Book \*head) | 链表排序函数（插入法），以表头为形参，返回排序后的链表的表头。不改变文件中数据的存储顺序，所以必须在读取文件形成链表后使用 |
| 16 | void over() | 退出图书信息管理系统，本质是exit(0) |

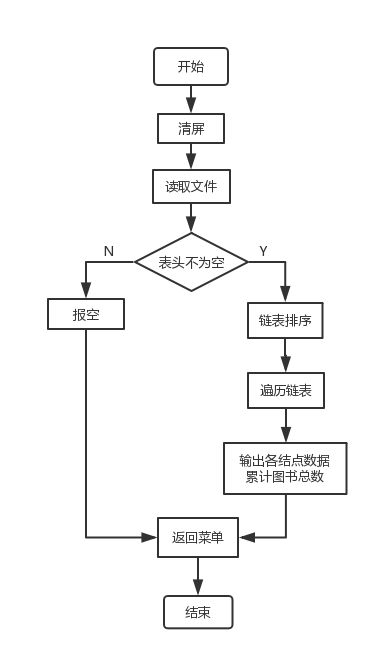
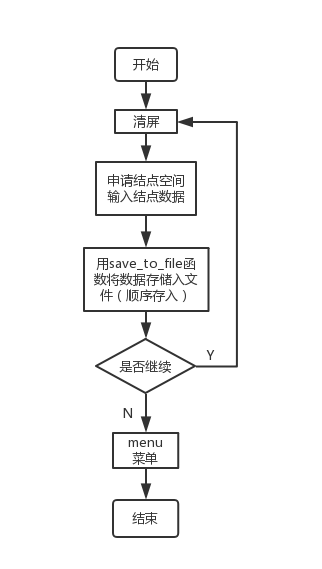
3.4 主要功能模块设计（流程图）

3.4.1 Book \*read\_from\_file()3.4.2 void save\_to\_file(Book \*p)

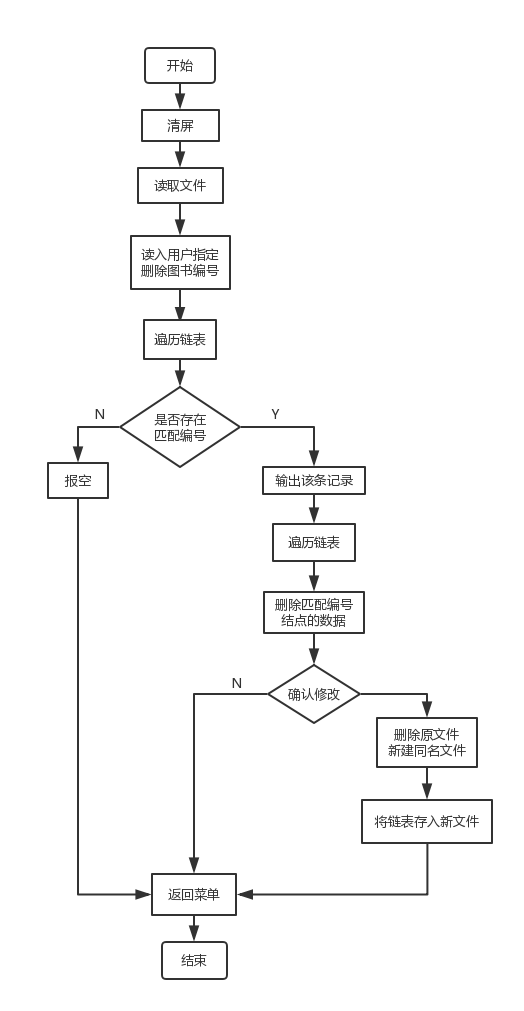
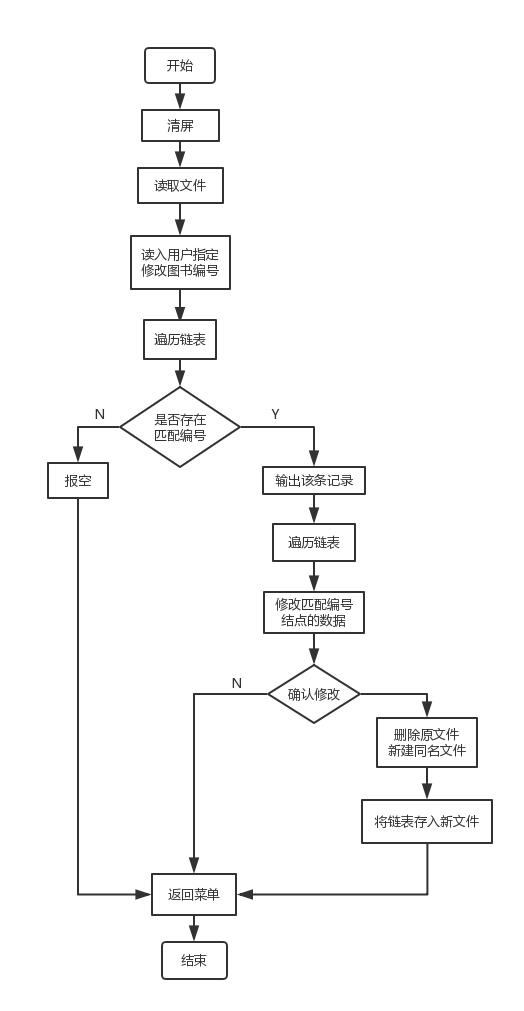
****

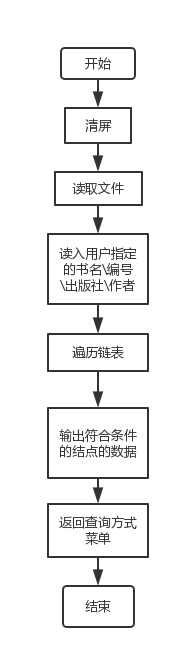
Y

N

3.4.3 void input\_book() 3.4.4 void print\_book()

3.4.5 void revise\_book()3.4.6 void del\_book()

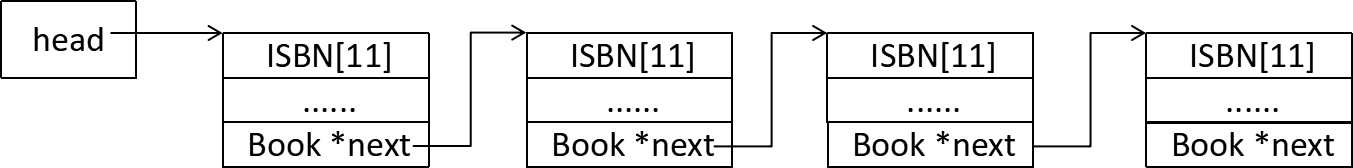
3.4.7 void find\_name\_book() void find\_author\_book()

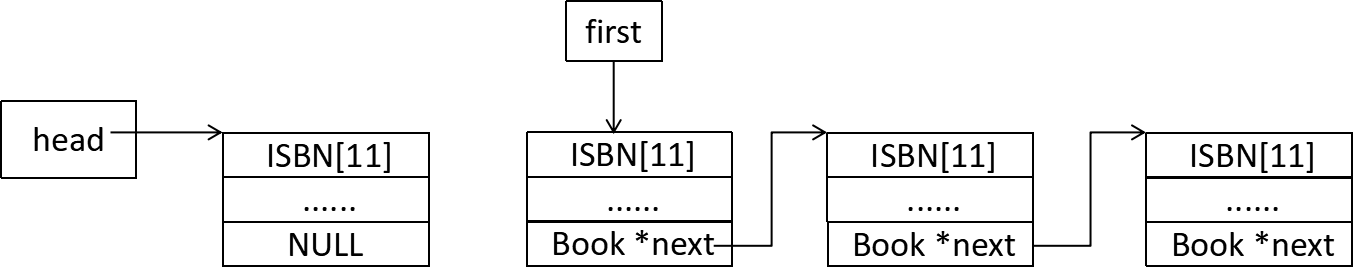
void find\_ISBN\_book() void find\_publish\_book()

3.5 算法实现

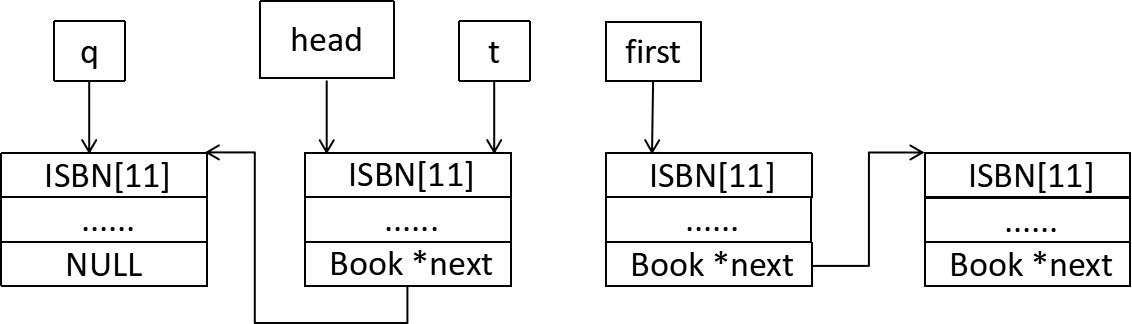
3.5.1 插入法实现链表排序

1 原链表：

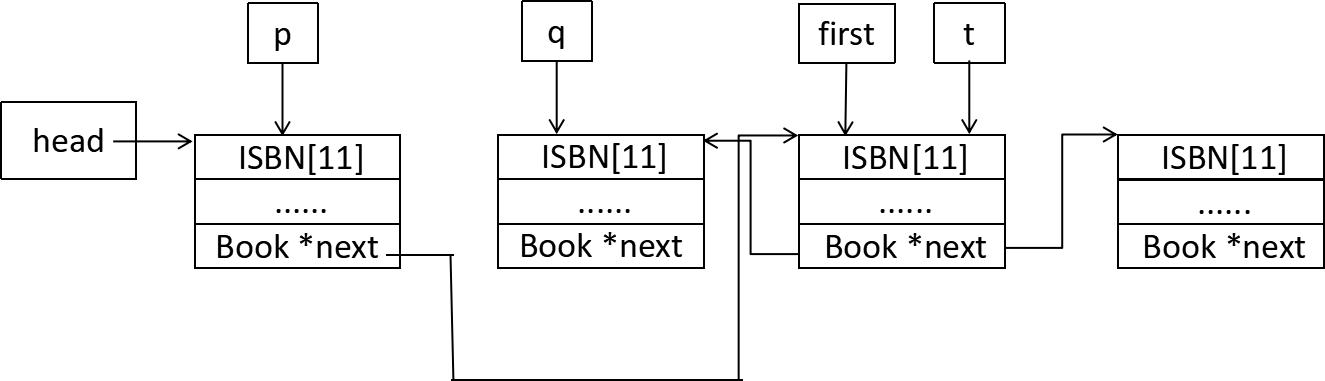
****

2 分成只含一个节点的目标链表和一段无序链表：

3.1 如果插入节点t需要插在目标链表表头前：



3.2 如果插入节点t插在目标链表中：



4.直至first指向最后一个节点的next域（即first==NULL）停止

3.5.2 其余算法实现

本系统中大量使用链表遍历或其变形等较为简单的算法，不画图具体解释。

遍历算法核心是工作指针从头结点开始，如所指结点符合条件则进行操作，然后工作指针指向后一个结点，重复流程，直至末尾结点。

4 课程设计小结

很感谢有这样的机会独立完成一个C语言小型系统。

通过此次课程设计的练习，我对C语言有了更深的理解，也对软件设计有了最初步的了解。其一是将书本知识用于实践中，体会了结构化程序设计方法、模块化编程思想。其二是尝试使用需求分析、总体设计、详细设计、编码、单元测试、综合测试的软件设计过程。其三是在反复使用函数、链表、指针、文件过程中对它们有了更深的了解。其四是提高了问题解决的能力，借鉴一些参考资料，学会遇到问题在网络上寻找解决方法或思路，而不单单是闭门造车。

在设计和编码的过程中，我也发现了自己的诸多不足。其一是前期对各功能模块没有清晰的构想，导致后期在编码和整合过程中出现了不少问题，且导致图书排序只实现了输出时排序，并没有做到存储顺序改变。其二是对文件读取、存储不熟悉，尤其是读取文件时末尾判断误差，导致了严重的读取错误，而且排查BUG用了很长时间。其三是编码能力较弱，空有想法而难以实现功能。

经过这一次课程设计的锻炼，我收获了很多，更发现了很多不足。课程设计是对综合能力的检验。我们总是在边学习边进步，希望借此次课设的机会，明确以后继续学习C语言或者学习其他计算机语言的目标和要求。

5 参考资料

1.谭浩强.C程序设计（第四版）[M].北京：清华大学出版社，2010：311-316

2.谭浩强.C程序设计（第四版）学习辅导[M].北京：清华大学出版社，2010：148-149

3.C语言课程设计-图书管理系统

https://blog.csdn.net/chaokudeztt/article/details/79720201?utm\_source=blogxgwz1，2018

4.C语言实现图书管理系统

https://blog.csdn.net/qq\_41226029/article/details/79438131，2018

5.C链表和文件操作实战--图书管理系统

https://blog.csdn.net/yueguanghaidao/article/details/7605283?utm\_source=blogxgwz2，2012

6.c语言程序设计项目设计报告（学生成绩管理系统）

https://wenku.baidu.com/view/82620b03c950ad02de80d4d8d15abe23492f035d.html?from=search

6 附录

源程序清单：

(程序运行环境为Visual Studio 2017）

#include "stdafx.h"

#include "windows.h"

#include "string.h"

#include "stdlib.h"

#define LEN sizeof(struct Book)

struct Book {

char ISBN[11]; //图书编号

char name[12]; //书名

char author[12]; //作者

char type[10]; //书籍分类

char publish[12]; //出版社

char time[10]; //出版时间

float price; //价格

int num; //数量

struct Book \*next; //指针域

};

//公共声明

void loca(int x, int y); //光标位置

void menu(); //菜单

void input\_book(); //图书入库

Book \*read\_from\_file(); //将文件中的内容读出到链表中

void save\_to\_file(Book \*p); //将图书信息存入文件

void find\_book(); //查询

void print\_book(); //图书总览

void del\_book(); //删除图书

void revise\_book(); //修改信息

void find\_name\_book(); //按书名查询

void find\_author\_book(); //按作者查询

void find\_ISBN\_book(); //按图书编号查询

void find\_publish\_book();//按出版社查询

Book \*sort\_book(Book \*head); //按图书编号由大到小排序

void over(); //退出

int main()

{

menu(); //菜单

return 0;

}

void loca(int x, int y) //将光标移动到x,y坐标处

{

COORD pos = { x,y }; //COORD是Windows API中定义的一种结构，表示一个字符在控制台屏幕上的坐标

HANDLE Out = GetStdHandle(STD\_OUTPUT\_HANDLE); //GetStdHandle（）返回标准的输入、输出或错误的设备的句柄，也就是获得输入、输出/错误的屏幕缓冲区的句柄

SetConsoleCursorPosition(Out, pos); //API中定位光标位置的函数

}

void menu()//菜单

{

do

{

//菜单界面模仿了其他系统

system("cls"); //清屏

loca(50, 5);

printf(" 图书信息管理系统");

loca(48, 8);

printf("| 1.图书入库 |");

loca(48, 10);

printf("| 2.图书遍览 |");

loca(48, 12);

printf("| 3.图书查询 |");

loca(48, 14);

printf("| 4.图书删除 |");

loca(48, 16);

printf("| 5.修改信息 |");

loca(48, 18);

printf("| 6.退出书库 |");

loca(80, 26);

printf("designed by 周绍峰");

loca(48, 20);

printf("------------------------");

loca(54, 22);

printf("请选择操作：");

int t; //选择功能

scanf("%d", &t);

switch (t)

{

case 1:input\_book(); break;

case 2:print\_book(); break;

case 3:find\_book(); break;

case 4:del\_book(); break;

case 5:revise\_book(); break;

case 6:over(); break;

default:break;

}

} while (1);

}

Book \*read\_from\_file() //从文件中读取，得到一个链表，用指针返回

{

FILE \*fp; //文件指针

int n = 0;

Book \*head = NULL, \*p2 = NULL, \*p1, \*pr=NULL;

fp = fopen("mybook.dat", "rb"); //以只读的方式打开文件（二进制）

if (fp == NULL)

{

printf("cannot open file\n");

}

while (!feof(fp)) //判断文件位置标志是否移动到文件末尾

{

n++;

p1 = (Book\*)malloc(LEN); //向内存申请一段空间

fread(p1, LEN, 1, fp); //将fp所指向的文件中的内容赋给p1

if (n == 1) //头指针、头结点

{

head = p1;

p2 = p1;

}

else

{

pr = p2;

p2->next = p1;

p2 = p1;

}

}

if (pr != NULL) //处理末尾节点

pr->next = NULL;

fclose(fp); //关闭文件

return(head); //返回头指针

}

void save\_to\_file(Book \*p) //将p中内容写入文件

{

FILE \*fp; //文件指针

if ((fp = fopen("mybook.dat", "ab"))== NULL) //以追加的方式打开名为mybook的二进制文件

{

printf("cannot open file"); //文件找不到

exit(0);

}

if (fwrite(p, LEN, 1, fp) != 1) //将p所指向的一段大小为N的内容存入fp所指向的文件中

{

printf("write error"); //没有存储成功则不返回1

}

fclose(fp); //关闭文件

}

void input\_book() //图书录入，写一个记录存一个记录

{

do

{

system("cls");

char t;

Book \*p;

p = (struct Book \*)malloc(LEN); //申请空间,用p记录

//输入图书信息

loca(48, 8);

printf("请输入图书编号：");

scanf("%s", p->ISBN); getchar();

loca(48, 10);

printf("请输入书名：");

scanf("%s", p->name); getchar();

loca(48, 12);

printf("请输入作者名：");

scanf("%s", p->author); getchar();

loca(48, 14);

printf("请输入图书类别：");

scanf("%s", p->type); getchar();

loca(48, 16);

printf("请输入图书出版单位：");

scanf("%s", p->publish); getchar();

loca(48, 18);

printf("请输入图书出版时间：");

scanf("%s", p->time); getchar();

loca(48, 20);

printf("请输入图书价格：");

scanf("%f", &p->price); getchar();

loca(48, 22);

printf("请输入图书数量：");

scanf("%d", &p->num);

save\_to\_file(p); //存储入文件

system("cls");

loca(46, 8);

printf("-------------------------");

loca(46, 9);

printf("| |");

loca(46, 10);

printf("| 保存成功！是否继续？ |");

loca(46, 12);

printf("| 1.是 2.否 |");

loca(46, 13);

printf("| |");

loca(46, 14);

printf("-------------------------");

loca(46, 15);

printf("=");

while (1) //利用死循环防止其他按键干扰

{

t = getchar();

if (t == '1')

break;

else if (t == '2')

menu();

}

} while (1);

}

void print\_book() //图书遍历

{

system("cls");

struct Book\* head, \*p;

int i = 11;

int sum = 0;

head = read\_from\_file(); //读取文件

loca(5, 5);

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*图书遍览\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

loca(5, 8);

printf("-------------------------------------------------------------------------------------------------------------");

loca(5, 9);

printf("图书编号 书名 作者名 图书类别 出版单位 出版时间 价格 数量");

loca(5, 10);

printf("-------------------------------------------------------------------------------------------------------------");

if (head == NULL)

{

loca(45, 11);

printf("书库暂时没有书(按任意键返回)");

getchar();

menu();

}

head = sort\_book(head); //链表排序

p = head; //开始遍历

do

{

loca(5, i);

printf("%-10s%-12s%-12s%-16s%-16s%-18s%-14.2f%-10d\n", p->ISBN, p->name, p->author, p->type, p->publish, p->time, p->price, p->num);

i+=2;

sum += p->num; //计算图书总量

p = p->next;

}while (p != NULL);

loca(48, 7);

printf("图书总数: %d 本", sum);

loca(48, i+1);

printf("按任意键返回...");

getchar(); getchar();

menu();

}

void revise\_book() //修改图书信息

{

do

{

system("cls");

Book \*head, \*p;

int j = 0;

char ch, t;

FILE \*fp; //文件指针

char \_ISBN[11]; //图书编号

char name[12]; //书名

char author[12]; //作者

char type[10]; //书籍分类

char publish[12]; //出版社

char time[10]; //出版时间

float price; //价格

int num; //数量

head = read\_from\_file(); //读取文件

p = head;

loca(48, 10);

printf("请输入你要修改的图书的编号：");

scanf("%s",\_ISBN);

loca(5, 5);

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*图书信息\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

loca(5, 8);

printf("-------------------------------------------------------------------------------------------------------------");

loca(5, 9);

printf("图书编号 书名 作者名 图书类别 出版社 出版时间 价格 数量");

loca(5, 10);

printf("-------------------------------------------------------------------------------------------------------------");

while (p != NULL)

{

if (strcmp(p->ISBN, \_ISBN) == 0) //如果检索到ISBN，则输出此条记录

{

j++;

loca(5, 12);

printf("%-10s%-12s%-16s%-12s%-14s%-18s%-14.2f%-10d\n", p->ISBN, p->name, p->author, p->type, p->publish, p->time, p->price, p->num);

Sleep(500);

}

p = p->next;

}

if (j == 0) //如果j=0，没有找到相应的信息

{

loca(40,12);

printf("没有找到相应的书籍(按0返回)");

while (1) //死循环防止其他按键干扰

{

ch = getchar();

if (ch == '0')

menu();

}

}

p = head; //让p重新指向表头

while (p != NULL) {

if (strcmp(p->ISBN, \_ISBN) == 0) //如果p不为空

{

//输入要修改的信息

loca(48, 14);

printf("请输入图书编号：");

scanf("%s", \_ISBN); getchar(); strcpy(p->ISBN, \_ISBN);

loca(48, 16);

printf("请输入书名：");

scanf("%s", name); getchar(); strcpy(p->name, name);

loca(48, 18);

printf("请输入作者名：");

scanf("%s", author); getchar(); strcpy(p->author, author);

loca(48, 20);

printf("请输入图书类别：");

scanf("%s", type); getchar(); strcpy(p->type, type);

loca(48, 22);

printf("请输入图书出版单位：");

scanf("%s", publish); getchar(); strcpy(p->publish, publish);

loca(48, 24);

printf("请输入图书出版时间：");

scanf("%s", time); getchar(); strcpy(p->time, time);

loca(48, 26);

printf("请输入图书价格：");

scanf("%f", &price); getchar(); p->price = price;

loca(48, 28);

printf("请输入图书数量：");

scanf("%d", &num); getchar(); p->num = num;

break;

}

p = p->next;

}

system("cls");

loca(46, 8);

printf("-------------------------");

loca(46, 9);

printf("| |");

loca(46, 10);

printf("| 是否确认修改？ |");

loca(46, 12);

printf("| 1.是 2.否 |");

loca(46, 13);

printf("| |");

loca(46, 14);

printf("-------------------------");

loca(46, 15);

while (1) //利用死循环防止其他按键干扰

{

t = getchar();

if (t == '1')

{

break;

}

else if (t == '2')

menu();

}

system("cls");

loca(46, 10);

printf("正在修改，请稍后....");

Sleep(1000);

fp = fopen("mybook.dat", "wb"); //以只写的方式打开名为mybook的二进制文件，删除之前的文件并重建文件

if (fp == NULL)

{

printf("cannot open file");

exit(0);

}

if (fwrite(head, LEN, 1, fp) != 1) //将头结点写入fp所指向的文件中

{

printf("write error!");

}

fclose(fp); //关闭文件

if (head != NULL) //如果head不为空

{

p = head->next; //让p指向第二个结点

fp = fopen("mybook.dat", "ab+"); //以追加的方式打开文件

while (p != NULL)

{

if (fwrite(p, LEN, 1, fp) != 1)//将p写入fp所指向的文件中

{

printf("write error!");

}

p = p->next;

}

fclose(fp); //关闭文件

}

system("cls");

loca(46, 10);

printf("修改成功！即将返回主菜单...");

Sleep(1000);

break;

} while (1);

}

void del\_book() //删除图书信息

{

do

{

system("cls");

FILE \*fp;

Book \*head, \*p, \*p2 = NULL;

int j = 0;

char \_ISBN[11];

char t, c, ch;

head = read\_from\_file(); //读取文件，返回表头地址

p = head;

loca(48, 10);

printf("请输入你要删除的图书的编号：");

scanf("%s", \_ISBN);

loca(5, 5);

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*图书信息\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

loca(5, 8);

printf("-------------------------------------------------------------------------------------------------------------");

loca(5, 9);

printf("图书编号 书名 作者名 图书类别 出版单位 出版时间 价格 数量");

loca(5, 10);

printf("-------------------------------------------------------------------------------------------------------------");

while (p != NULL)

{

if (strcmp(p->ISBN, \_ISBN) == 0) //如果检索到ISBN，则输出此条记录以供查阅

{

j++;

loca(5, 12);

printf("%-10s%-12s%-14s%-14s%-14s%-18s%-14.2f%-10d\n", p->ISBN, p->name, p->author, p->type, p->publish, p->time, p->price, p->num);

Sleep(500);

}

p = p->next;

}

if (j == 0) //如果j=0，没有找到相应的信息

{

loca(40, 12);

printf("没有找到相应的书籍(按0返回)");

while (1) //死循环防止其他按键干扰

{

ch = getchar();

if (ch == '0')

menu();

}

}

loca(46, 14);

printf("-------------------------");

loca(46, 15);

printf("| |");

loca(46, 16);

printf("| 是否确认删除？ |");

loca(46, 17);

printf("| 1.是 2.否 |");

loca(46, 18);

printf("| |");

loca(46, 19);

printf("-------------------------");

loca(46, 20);

printf("| = |");

loca(46, 21);

printf("-------------------------");

loca(54, 20);

while (1)

{

t = getchar();

if (t == '1')

break;

else if (t == '2')

menu();

}

p = head; //删除节点

while (\_ISBN != p->ISBN && p->next != NULL)

{

p2 = p;

p = p->next;

}

if (p!=NULL)

{

if (p == head) head = p->next;

else p2->next = p->next;

}

fp = fopen("mybook.dat", "wb"); //以只写的方式打开名为mybook的二进制文件，删除之前的文件并重建文件

if (fp == NULL)

{

loca(50, 10);

printf("cannot open file");

exit(0);

}

if (fwrite(head, LEN, 1, fp) != 1) //将头结点写入fp所指向的文件中

{

printf("write error!");

}

fclose(fp); //关闭文件

if (head != NULL) //如果head不为空

{

p = head->next; //让p指向第二个结点

fp = fopen("mybook.dat", "ab+"); //以追加的方式打开文件

while (p != NULL)

{

if (fwrite(p, LEN, 1, fp) != 1)//将p写入fp所指向的文件中

{

printf("write error!");

}

p = p->next;

}

fclose(fp); //关闭文件

}

system("cls");

loca(46, 10);

printf("正在删除，请稍后....");

Sleep(500);

system("cls");

loca(46, 8);

printf("-------------------------");

loca(46, 9);

printf("| |");

loca(46, 10);

printf("| 删除成功，是否继续？ |");

loca(46, 12);

printf("| 1.是 2.否 |");

loca(46, 13);

printf("| |");

loca(46, 14);

printf("-------------------------");

while (1)

{

c = getchar();

if (c == '1')

{

break;

head = read\_from\_file();

}

else if (c == '2')

{

menu();

}

}

} while (1);

}

void find\_book() //查询图书

{

do

{

system("cls"); //清屏

char t;

loca(56,5);

printf("图书查询");

loca(48,8);

printf("| 1.编号 查询 |");

loca(48,10);

printf("| 2.书名 查询 |");

loca(48,12);

printf("| 3.作者 查询 |");

loca(48, 14);

printf("| 4.出版社查询 |");

loca(48,16);

printf("| 0.返回主菜单 |");

loca(48, 18);

printf("-------------------------");

loca(54, 20);

printf("选择查询方式:");

t = getchar();

switch (t) //简单的选择语句

{

case '0':menu(); break;

case '1':find\_ISBN\_book(); break;

case '2':find\_name\_book(); break;

case '3':find\_author\_book(); break;

case '4':find\_publish\_book(); break;

default :break;

}

}while (1);

}

void find\_ISBN\_book() //按图书编号查询

{

system("cls");

Book \*head, \*p;

int i = 11, sum = 0, j = 0;

head = read\_from\_file();

char ch;

char number[10];

loca(48, 8);

printf("请输入您要查询图书的编号：");

getchar();

gets\_s(number);

loca(48, 10);

printf("正在查询...");

Sleep(500);

p = head;

loca(5, 5);

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*图书总览\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

loca(5, 8);

printf("-------------------------------------------------------------------------------------------------------------");

loca(5, 9);

printf("图书编号 书名 作者名 图书类别 出版社 出版时间 价格 数量");

loca(5, 10);

printf("-------------------------------------------------------------------------------------------------------------");

while (p != NULL)

{

if (p != NULL && strcmp(p->ISBN, number) == 0) //如果查询到相应的编号，则输出此条记录

{

j++;

loca(5, i);

printf("%-10s%-12s%-16s%-12s%-14s%-18s%-14.2f%-10d\n", p->ISBN, p->name, p->author, p->type, p->publish, p->time, p->price, p->num);

i+=2;

sum += p->num; //记录数量

}

p = p->next;

}

if (j == 0) //如果j=0，没有找到相应的信息

{

loca(40, 12);

printf("没有找到相应的书籍(按0返回查询菜单)");

while (1) //死循环防止其他按键干扰

{

ch = getchar();

if (ch == '0')

find\_book();

}

}

loca(52, 7);

printf("总量为%d本", sum);

loca(50, i+2);

printf("按任意键返回...");

getchar();

find\_book();

}

void find\_author\_book() //按作者名查询

{

system("cls");

Book \*head, \*p;

int i = 11, sum = 0, j = 0;

head = read\_from\_file();

char author[10];

char ch;

loca(48, 8);

printf("请输入您要查询图书的作者名：");

getchar();

gets\_s(author);

loca(48, 10);

printf("正在查询....");

Sleep(500);

p = head;

loca(5, 5);

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*图书总览\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

loca(5, 8);

printf("-------------------------------------------------------------------------------------------------------------");

loca(5, 9);

printf("图书编号 书名 作者名 图书类别 出版社 出版时间 价格 数量");

loca(5, 10);

printf("-------------------------------------------------------------------------------------------------------------");

while (p != NULL)

{

if (p != NULL && strcmp(p->author, author) == 0)

{

j++;

loca(5, i);

printf("%-10s%-12s%-16s%-12s%-14s%-18s%-14.2f%-10d\n", p->ISBN, p->name, p->author, p->type, p->publish, p->time, p->price, p->num);

sum += p->num;

i+=2;

}

p = p->next;

}

if (j == 0) //如果j=0，没有找到相应的信息

{

loca(40, 12);

printf("没有找到相应的书籍(按0返回查询菜单)");

while (1) //死循环防止其他按键干扰

{

ch = getchar();

if (ch == '0')

find\_book();

}

}

loca(46, 7);

printf("该作者图书数量为%d本",sum);

loca(50, i+2);

printf("按任意键返回...");

getchar();

find\_book();

}

void find\_name\_book() //按书名查询

{

system("cls");

Book\* head, \*p;

int i = 11, sum = 0, j = 0;

head = read\_from\_file();

char name[10];

char ch;

loca(48, 8);

printf("请输入您要查询图书的书名：");

getchar();

gets\_s(name);

loca(48, 10);

printf("正在查询....");

Sleep(1500);

p = head;

loca(5, 5);

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*图书总览\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

loca(5, 8);

printf("-------------------------------------------------------------------------------------------------------------");

loca(5, 9);

printf("图书编号 书名 作者名 图书类别 出版社 出版时间 价格 数量");

loca(5, 10);

printf("-------------------------------------------------------------------------------------------------------------");

while (p != NULL)

{

if (p != NULL && strcmp(p->name, name) == 0)

{

j++;

loca(5, i);

printf("%-10s%-12s%-16s%-12s%-14s%-18s%-14.2f%-10d\n", p->ISBN, p->name, p->author, p->type, p->publish, p->time, p->price, p->num);

i+=2;

sum += p->num;

}

p = p->next;

}

if (j == 0) //如果j=0，没有找到相应的信息

{

loca(40, 12);

printf("没有找到相应的书籍(按0返回查询菜单)");

while (1) //死循环防止其他按键干扰

{

ch = getchar();

if (ch == '0')

find\_book();

}

}

loca(48, 7);

printf("该图书总量为%d本", sum);

loca(50, i+1);

printf("按任意键返回...");

getchar();

find\_book();

}

void find\_publish\_book() //按出版社查询

{

system("cls");

Book\* head, \*p;

int i = 11, sum = 0, j = 0;

head = read\_from\_file();

char \_publish[10];

char ch;

loca(48, 8);

printf("请输入您要查询的出版社：");

getchar();

gets\_s(\_publish);

loca(48, 10);

printf("正在查询....");

Sleep(1500);

p = head;

loca(5, 5);

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*图书遍览\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

loca(5, 8);

printf("-------------------------------------------------------------------------------------------------------------");

loca(5, 9);

printf("图书编号 书名 作者名 图书类别 出版社 出版时间 价格 数量");

loca(5, 10);

printf("-------------------------------------------------------------------------------------------------------------");

while (p != NULL)

{

if (p != NULL && strcmp(p->publish, \_publish) == 0)

{

loca(5, i);

printf("%-10s%-12s%-16s%-12s%-14s%-18s%-14.2f%-10d\n", p->ISBN, p->name, p->author, p->type, p->publish, p->time, p->price, p->num);

i += 2;

sum += p->num;

}

p = p->next;

}

if (j == 0) //如果j=0，没有找到相应的信息

{

loca(40, 12);

printf("没有找到相应的书籍(按0返回查询菜单)");

while (1) //死循环防止其他按键干扰

{

ch = getchar();

if (ch == '0')

find\_book();

}

}

loca(46, 7);

printf("该出版社图书总量为%d本", sum);

loca(50, i + 1);

printf("按任意键返回...");

getchar();

find\_book();

}

Book \*sort\_book(Book \*head)

{

Book \*first; //为原链表剩下用于直接插入排序的节点头指针

Book \*t; //临时指针变量：插入节点

Book \*q, \*p=NULL; //临时指针变量

first = head->next; //原链表剩下用于直接插入排序的节点链表

head->next = NULL; //只含有一个节点的链表

while (first != NULL) //遍历剩下无序的链表

{

for (t = first, q = head; ((q!= NULL) && (strcmp(q->ISBN,t->ISBN)>0)); p = q, q = q->next); //无序节点在有序链表中找插入的位置

//退出for循环，就是找到了插入的位置

first = first->next; //无序链表中的节点离开，以便它插入到有序链表中

if (q == head) //插在第一个节点之前

{

head = t;

}

else //p是q的前驱

{

p->next = t;

}

t->next = q;

}

return head;

}

void over() //退出系统

{

char t;

system("cls");

loca(48, 11);

printf("-----------------------");

loca(48, 12);

printf("| 您确定要退出吗? |");

loca(48, 14);

printf("| 1.确定 2.取消 |");

loca(48, 15);

printf("-----------------------");

loca(48, 16);

while (1)

{

t = getchar();

switch (t)

{

case '1':

system("cls");

loca(48, 10);

printf("正在退出....");

Sleep(1500);

system("cls");

loca(48, 10);

printf("已退出软件");

loca(48, 12);

printf("谢谢使用！");

Sleep(3000);

loca(48, 14);

exit(0); break; //终止程序

case '2':

menu(); break; //调用函数，进入菜单

default :break;

}

}

}