**西 安 邮 电 大 学**

**（计算机学院）**

**C语言课程设计报告**

**专业名称：网络工程**

**班 级：网工1801**

**学生姓名：李睿**

**学 号：04182023**

**班内序号：20**

**指导教师：**

**实验时间：2019.3.29**

1. **基本要求**

要求完成某个图书馆对图书信息的管理。每种图书的基本信息包括：图书编号、书名、作者、图书种类、出版方、出版日期、图书价格、图书数量，总共八个方面。要求数据以文件方式存储，实现信息的增删改查，并且查阅数据时可以对数据进行排序、统计，实现单项、多项查询。按系统特点设计界面，提高界面友好性。

1. **主要功能**

（1）图书管理员注册和登陆功能。

（2）图书信息的录入及存储：录入基本信息，并且检验录入的信息是否有重复的。

（3）更新：（增添、删除、修改）即对图书信息进行更新的过程。

（4）查询：按照图书编号、书名或作者条件进行查询。

（5）排序：

①按图书编号进行升序排序。

②按图书价格进行升序排序。

（6）输出报表：

图书基本信息

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 编号 | 书名 | 作者 | 分类 | 出版方 | 出版日期 | 价格 | 数量 |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

（6）界面：界面采用菜单选择，层次清楚，界面美观，操作方便。

**图书管理系统**

1. **实 验目的和内容**

本课程设计是为了完成某个图书馆对图书信息的管理，其中包含对图书信息的增加、删除、修改、查询、排序等。主要目的如下：

1. 综合运用C语言知识，掌握软件系统的设计方法和步骤；
2. 熟悉掌握函数的定义及调用；
3. 熟悉掌握结构体变量的定义及使用；
4. 熟悉掌握对文件的读写操作；
5. 加深对指针的理解，提高对指针的运用能力；
6. 上机调试程序，提高分析问题和解决问题的能力，提高程序设计水平。
7. **总体设计**

本系统使用结构体存储图书信息，并且将图书信息以文本形式保存到bookinformation.txt文件中，从文件中读取图书信息，存放至链表中，该系统可分为5大模块：图书信息录入（增添信息）、图书信息查询（多项查询：包括图书编号、书名及作者）、图书信息删除（包含删除作者图书信息、删除指定图书信息：书名或图书编号）、图书信息修改、图书信息总览（包含按图书编号升序排序和按图书价格升序排序）。同时增添了图书管理员的注册及登录功能，更符合实际需求。各模块分块设计，分块调试，然后有机的组合在一起。

1. **存储结构体设计**

图书信息存储结构体：

struct BOOK

{

char bianhao[11]; //图书编号

char bookname[50]; //书名

char writer[50]; //作者

char leibie[20]; //图书类型

char chubanfang[50]; //出版方

char data[11]; //出版日期

float price; //价格

int num; //数量

struct BOOK\* next;

};

图书管理员信息结构体：

struct admin

{

char adname[20];

char mima[20];

char phone[12];

struct admin \* next;

};

**2．功能模块设计**

本图书管理系统共分为七大功能模块，如图：

图书管理系统

管理员注册与登录

图书信息总览

图书信息修改

图书信息删除

录入图书信息

图书信息查询

删除作者图书

按图书作者、类型统计

按图书编号、价格排序

删除条件信息

图书管理系统总体设计图

管理员注册与登录：用户可以根据系统提示，依次填写用户名、登录密码和手机号信息，完成管理员的注册，并可用已注册的用户名和密码登录图书管理系统，对图书信息进行增删改查等操作。

录入图书信息：用户可以根据系统所提示信息输入相应的图书信息并保存，每条信息都应该有唯一确定的图书编号。

图书信息查询：用户可以通过输入想要查询图书的编号、书名或作者来获取详细的图书信息。

图书信息删除：用户可以根据系统提示信息对保存在文件内的图书信息进行删除操作。

图书信息修改：用户可以输入想要修改图书的编号或书名来重新录入这本书的基本信息。

图书信息总览：用户可以选择按图书编号升序排序或按图书价格升序排序来总览库文件中的所有内容。也可以选择按图书作者或图书类型进行统计。

**函数功能描述：**

void gotoxy(int x, int y);

功能：实现光标移动；

输入参数：移动到该位置的坐标；

输出参数：无。

void JM();

功能：输出主菜单的界面；

输入参数： 无；

输出参数：无。

void DelJM();

功能：输出删除菜单的界面；

输入参数： 无；

输出参数：无。

void Next();

功能：输出下一步提示框；

输入参数： 无；

输出参数：无。

void output(struct BOOK\* head);

功能：制表输出；

输入参数： 图书信息结构体指针；

输出参数：无。

void CL();

功能：创建链表，存储结构体中的图书信息；

输入参数：无 ；

输出参数：无。

void Save();

功能：输出是否保存的提示框；

输入参数： 无；

输出参数：无。

void Del();

功能：输出是否删除的提示框；

输入参数： 无；

输出参数：无。

void Succeed();

功能：输出操作成功的提示框；

输入参数： 无；

输出参数：无。

void FileSave();

功能：将链表中保存的的内容写入文件；

输入参数： 无；

输出参数：无。

void PXJM();

功能：输出排序功能界面；

输入参数： 无；

输出参数：无。

void FileRead(struct BOOK\* HEAD);

功能：从文件中读取内容，并将其保存在链表中；

输入参数：图书信息结构体指针；

输出参数：无。

void DelWriter(struct BOOK\* HEAD);

功能：实现删除链表中包含某一作者图书信息的结点；

输入参数：图书信息结构体指针；

输出参数：无。

void DelBook(struct BOOK\* HEAD);

功能：删除链表中和输入条件相同的结点；

输入参数：图书信息结构体指针；

输出参数：无。

void PX\_b(struct BOOK\* HEAD);

功能：对链表中的信息按编号升序排序；

输入参数：图书信息结构体指针；

输出参数：无。

void PX\_p(struct BOOK\* HEAD);

功能：对链表中的信息按价格升序排序；

输入参数：图书信息结构体指针；

输出参数：无。

void Check();

功能：通过图书编号、书名或作者查询文件中的信息；

输入参数：无；

输出参数：无。

void Change();

功能：实现对文件中的内容进行修改操作；

输入参数：无；

输出参数：无。

int FRN();

功能：判断是否文件为空文件；

输入参数：无；

输出参数：0或-1。

void menu();

功能：输出主菜单功能选择界面；

输入参数：无；

输出参数：无。

void menu3();

功能：输出选项3复菜单功能选择界面；

输入参数：无；

输出参数：无。

void menu5();

功能：输出选项5复菜单功能选择界面；

输入参数：无；

输出参数：无。

int ReCheck(char str[]);

功能：检查输入的字符串数组是否与文件中的内容相同；

输入参数：字符数组；

输出参数：0或-1。

void FileReadAdmin(struct admin\*head);

功能：从文件读出用户信息

输入参数：结构体指针

输出参数： 无

void output\_writer();

功能：按作者统计图书数量

输入参数：无

输出参数： 无

void output\_type();

功能：按图书类型统计图书数量

输入参数：无

输出参数： 无

**程序实现：**

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#include <Windows.h>

char ch, c, h;//选择

struct BOOK

{

char bianhao[11];//图书编号

char bookname[50];//书名

char writer[50];//作者

char leibie[20];//图书类型

char chubanfang[50];//出版方

char data[11];//出版日期

float price;//价格

int num;//数量

struct BOOK\* next;

};

struct BOOK\* head;

struct BOOK\* p;

void gotoxy(int x, int y);//光标移动

void JM();//界面

void DelJM();//删除功能界面

void Next();//输入y继续，输入其他键退出

void output(struct BOOK\* head);//输出链表

void CL();//创建保存链表

void Save();//保存提示框

void Del();//删除提示框

void Succeed();//成功

void FileSave();//文件保存

void PXJM();//排序界面

void FileRead(struct BOOK\* HEAD);//文件读取

void DelWriter(struct BOOK\* HEAD);//删除作者

void DelBook(struct BOOK\* HEAD);//删除图书（编号或书名）

void PX\_b(struct BOOK\* HEAD);//编号排序

void PX\_p(struct BOOK\* HEAD);//价格排序

void Check();//图书查询

void Change();//图书信息修改

int FRN();//判断是否文件为空

void menu();//主菜单

void menu5();//选项5复菜单

void menu3();//选项3复菜单

int ReCheck(char str[]);//判断输入编号是否重复

//主函数：

int main(void)

{

menu();

return 0;

}

//自定义函数：

void gotoxy(int x, int y)

{

COORD pos = { x, y };

HANDLE Out = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleCursorPosition(Out, pos);

}

void JM()

{

system("cls");

printf("\*请最大化使用\*");

gotoxy(80, 11); printf("\*\*\*\*\*\*\*\*\*图书管理系统\*\*\*\*\*\*\*\*\*");

gotoxy(80, 13); printf(" | 1.图书信息录入 |");

gotoxy(80, 15); printf(" | 2.图书信息查询 |");

gotoxy(80, 17); printf(" | 3.图书信息删除 |");

gotoxy(80, 19); printf(" | 4.图书信息修改 |");

gotoxy(80, 21); printf(" | 5.图书信息总览 |");

gotoxy(80, 23); printf(" | 6.退出程序 |");

gotoxy(86, 25); printf("输入序号，选择功能");

}

void Next()

{

printf("输入y继续，输入其他键退出");

}

void Continue()

{

system("cls");

gotoxy(75, 20); printf("----------------------------------------\n");

gotoxy(75, 22); printf("| |\n");

gotoxy(75, 24); printf("| Continue to enter information? |\n");

gotoxy(75, 26); printf("| y / n |\n");

gotoxy(75, 28); printf("| |\n");

gotoxy(75, 30); printf("----------------------------------------\n");

gotoxy(75, 32); printf(" 输入y继续录入信息，输入n结束录入！");

}

void Save()

{

system("cls");

gotoxy(80, 20); printf("---------------------------\n");

gotoxy(80, 22); printf("| |\n");

gotoxy(80, 24); printf("| Are you save it ? |\n");

gotoxy(80, 26); printf("| y / n |\n");

gotoxy(80, 28); printf("| |\n");

gotoxy(80, 30); printf("---------------------------\n");

gotoxy(80, 32); printf(" 输入y保存！输入n取消！");

}

void Del()

{

system("cls");

gotoxy(80, 20); printf("---------------------------\n");

gotoxy(80, 22); printf("| |\n");

gotoxy(80, 24); printf("| Are you delete it ? |\n");

gotoxy(80, 26); printf("| y / n |\n");

gotoxy(80, 28); printf("| |\n");

gotoxy(80, 30); printf("---------------------------\n");

gotoxy(80, 32); printf(" 输入y保存！输入n取消！");

}

void DelJM()

{

system("cls");

gotoxy(80, 11); printf("\*\*\*\*\*\*\*\*\*图书管理系统\*\*\*\*\*\*\*\*\*");

gotoxy(80, 13); printf(" | 1.删除某作者图书 |");

gotoxy(80, 15); printf(" | 2.编号或书名删除 |");

gotoxy(80, 17); printf(" | 3.返回系统主界面 |");

gotoxy(86, 25); printf(" 输入序号，选择功能");

//c = getch();

}

void PXJM()

{

system("cls");

gotoxy(80, 11); printf("\*\*\*\*\*\*\*\*\*图书管理系统\*\*\*\*\*\*\*\*\*");

gotoxy(80, 13); printf(" | 1.按图书编号排序 |");

gotoxy(80, 15); printf(" | 2.按图书价格排序 |");

gotoxy(80, 17); printf(" | 3.返回系统主界面 |");

gotoxy(86, 19); printf(" 输入序号，选择功能");

}

void CL()//创建链表

{

head->next = NULL;

p = (struct BOOK\*)malloc(sizeof(struct BOOK));

p->next = NULL;

loop:

{

struct BOOK\* s = (struct BOOK\*)malloc(sizeof(struct BOOK));

gotoxy(80, 11); printf("请输入图书编号（小于10位）：");

loop1:scanf("%s", s->bianhao);

if (ReCheck(s->bianhao) == -1)

{

system("cls");

gotoxy(80, 11); printf("输入的编号重复！请重新输入编号：");

goto loop1;

}

gotoxy(80, 13); printf("请输入书名（小于25字）：");

scanf("%s", s->bookname);

gotoxy(80, 15); printf("请输入作者姓名（小于25字）：");

scanf("%s", s->writer);

gotoxy(80, 17); printf("请输入图书类别（小于10字）：");

scanf("%s", s->leibie);

gotoxy(80, 19); printf("请输入图书出版方（小于25字）：");

scanf("%s", s->chubanfang);

gotoxy(80, 21); printf("请输入图书出版日期（如2018.03.03）：");

scanf("%s", s->data);

gotoxy(80, 23); printf("请输入图书价格：");

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

gotoxy(80, 25); printf("请输入图书数量：");

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

if (head->next == NULL)

{

head->next = s;

s->next = NULL;

}

else

{

p->next = s;

s->next = NULL;

}

p = s;

//文件存储

Save();

char a;

loop2:

a = getch();

if (a == 'y' || a == 'Y')

{

FileSave();

Succeed();

Sleep(1000);

}

else if (a == 'n' || a == 'N')

{

system("cls");

gotoxy(90, 25); printf("修改未保存！");

Sleep(1000);

system("cls");

}

else

{

goto loop2;

}

gotoxy(80, 25); Continue();

ch = getch();

if (ch == 'y' || ch == 'Y')

{

system("cls");

goto loop;

}

}

}

void output(struct BOOK\* s)

{

int t = 0;

system("cls");

struct BOOK\* p = (struct BOOK\*)malloc(sizeof(struct BOOK));

p = s;

gotoxy(33, 6); printf("----------------------------------------------------------------------------------------------------------------------------------\n");

gotoxy(40, 7); printf("图书编号 书名 作者 图书类别 出版方 出版日期 图书价格 图书数量\n");

gotoxy(33, 8); printf("----------------------------------------------------------------------------------------------------------------------------------\n");

while (p->next != NULL)

{

gotoxy(40, 9 + t); printf("%s", p->next->bianhao);

gotoxy(55, 9 + t); printf("%s", p->next->bookname);

gotoxy(69, 9 + t); printf("%s", p->next->writer);

gotoxy(82, 9 + t); printf("%s", p->next->leibie);

gotoxy(99, 9 + t); printf("%s", p->next->chubanfang);

gotoxy(114, 9 + t); printf("%s", p->next->data);

gotoxy(131, 9 + t); printf("%g", p->next->price);

gotoxy(150, 9 + t); printf("%d\n", p->next->num);

t = t + 2;

p = p->next;

}

}

void FileSave()

{

char a;

a = getch();

system("cls");

if (a == 'y' || a == 'Y')

{

FILE\* fp = fopen("bookinformation.txt", "a");

if (fp == NULL)

{

printf("打开文件失败！\n");

return 0;

}

fprintf(fp, "%s %s %s %s %s %s %g %d\n", p->bianhao, p->bookname, p->writer, p->leibie, p->chubanfang, p->data, p->price,p->num);

fclose(fp);

}

}

void FileRead(struct BOOK\* HEAD)

{

FILE \*fp = fopen("bookinformation.txt", "rt");

if (fp == NULL)

{

printf("文件打开失败！\n");

return 0;

}

struct BOOK\* p = (struct BOOK\*)malloc(sizeof(struct BOOK));

p->next = NULL;

while (!feof(fp))//判断是否读到文件尾

{

struct BOOK\* s = (struct BOOK\*)malloc(sizeof(struct BOOK));

fscanf(fp, "%s %s %s %s %s %s %f %d\n", s->bianhao, s->bookname, s->writer, s->leibie, s->chubanfang, s->data, &s->price, &s->num);

if (HEAD->next == NULL)

{

HEAD->next = s;

s->next = NULL;

}

else

{

p->next = s;

s->next = NULL;

}

p = s;

}

}

void DelWriter(struct BOOK\* HEAD)

{

int t = 0;

int a = 0;//删除后链表长度计数器

char ch[50];

struct BOOK\* q1 = (struct BOOK\*)malloc(sizeof(struct BOOK));

q1->next = NULL;

struct BOOK\* q2 = (struct BOOK\*)malloc(sizeof(struct BOOK));

q2->next = NULL;

q1 = HEAD;

gotoxy(75, 20); printf("请输入要删除的图书的作者：");

scanf("%s", ch);

while (q1->next != NULL)

{

a++;

q2 = q1->next;

if (q2->next != NULL && (strcmp(ch, q2->writer) == 0 ))

{

t++;

a--;

q1->next = q2->next;

q2 = q2->next;

continue;

}

else if (q2->next == NULL && (strcmp(ch, q2->writer)==0))

{

t++;

a--;

q1->next = NULL;

break;

}

q1 = q1->next;

}

if (t != 0)

{

loop:

Del();

char c;

c = getch();

if (c == 'y' || c == 'Y')

{

q1 = HEAD->next;

FILE\* fp = fopen("bookinformation.txt", "wt");

if (fp == NULL)

{

printf("打开文件失败！\n");

return 0;

}

for (int i = 0; i < a; i++)

{

fprintf(fp, "%s %s %s %s %s %s %g %d\n", q1->bianhao, q1->bookname, q1->writer, q1->leibie, q1->chubanfang, q1->data, q1->price, q1->num);

q1 = q1->next;

}

fclose(fp);

Succeed();

Sleep(1000);

system("cls");

}

else if (c == 'n' || c == 'N')

{

system("cls");

gotoxy(83, 22); printf("修改未保存！");

gotoxy(83, 25); printf("按任意键返回删除菜单！");

getch();

menu3();

}

else

{

goto loop;

}

}

else if (t==0)

{

gotoxy(80, 22); printf("没有找到此作者！！！");

Sleep(1000);

}

}

void DelBook(struct BOOK\* HEAD)

{

int t = 0;

int a = 0;//删除后链表长度计数器

char ch[50];

struct BOOK\* q1 = (struct BOOK\*)malloc(sizeof(struct BOOK));

q1->next = NULL;

struct BOOK\* q2 = (struct BOOK\*)malloc(sizeof(struct BOOK));

q2->next = NULL;

q1 = HEAD;

gotoxy(75, 20); printf("请输入要删除的图书的编号或书名：");

scanf("%s", ch);

while (q1->next != NULL)

{

a++;

q2 = q1->next;

if (q2->next != NULL && (strcmp(ch, q2->bianhao) == 0 || strcmp(ch, q2->bookname) == 0))

{

t++;

a--;

q1->next = q2->next;

q2 = q2->next;

continue;

}

else if (q2->next == NULL && (strcmp(ch, q2->bianhao) == 0 || strcmp(ch, q2->bookname) == 0))

{

t++;

a--;

q1->next = NULL;

break;

}

q1 = q1->next;

}

if (t != 0)

{

loop:

Del();

char c;

c = getch();

if (c == 'y' || c == 'Y')

{

q1 = HEAD->next;

FILE\* fp = fopen("bookinformation.txt", "wt");

if (fp == NULL)

{

printf("打开文件失败！\n");

return 0;

}

for (int i = 0; i < a; i++)

{

fprintf(fp, "%s %s %s %s %s %s %g %d\n", q1->bianhao, q1->bookname, q1->writer, q1->leibie, q1->chubanfang, q1->data, q1->price, q1->num);

q1 = q1->next;

}

fclose(fp);

Succeed();

Sleep(1000);

system("cls");

}

else if (c == 'n' || c == 'N')

{

system("cls");

gotoxy(83, 22); printf("修改未保存！");

gotoxy(83, 25); printf("按任意键返回删除菜单！");

getch();

menu3();

}

else

{

goto loop;

}

}

else

{

gotoxy(80, 22); printf("这本书还没有入库呢！！！");

}

}

void PX\_b(struct BOOK\* HEAD)

{

struct BOOK\* q = (struct BOOK\*)malloc(sizeof(struct BOOK));

q = HEAD;

int a = 0;//链表长度计数器

while (q->next != NULL)

{

a++;

q = q->next;

}

struct BOOK\* q1 = (struct BOOK\*)malloc(sizeof(struct BOOK));

q1->next = NULL;

struct BOOK\* q2 = (struct BOOK\*)malloc(sizeof(struct BOOK));

q2->next = NULL;

if (HEAD->next->next != NULL)

{

for (int i = 0; i<a - 1; i++)

{

q = HEAD;

q1 = q->next;

q2 = q1->next;

while (q2->next != NULL)

{

if (strcmp(q1->bianhao, q2->bianhao) > 0)

{

q->next = q1->next;

q1->next = q2->next;

q2->next = q1;

q2 = q1->next;

q = q->next;

continue;

}

q = q->next;

q1 = q1->next;

q2 = q2->next;

}

if (strcmp(q1->bianhao, q2->bianhao) > 0)

{

q->next = q2;

q2->next = q1;

q1->next = NULL;

}

}

}

q1 = HEAD->next;

FILE\* fp = fopen("bookinformation.txt", "wt");

if (fp == NULL)

{

printf("打开文件失败！\n");

return 0;

}

for (int i = 0; i < a; i++)

{

fprintf(fp, "%s %s %s %s %s %s %g %d\n", q1->bianhao, q1->bookname, q1->writer, q1->leibie, q1->chubanfang, q1->data, q1->price, q1 ->num);

q1 = q1->next;

}

fclose(fp);

}

void PX\_p(struct BOOK\* HEAD)

{

struct BOOK\* q = (struct BOOK\*)malloc(sizeof(struct BOOK));

q = HEAD;

int a = 0;//链表长度计数器

while (q->next != NULL)

{

a++;

q = q->next;

}

struct BOOK\* q1 = (struct BOOK\*)malloc(sizeof(struct BOOK));

q1->next = NULL;

struct BOOK\* q2 = (struct BOOK\*)malloc(sizeof(struct BOOK));

q2->next = NULL;

if (HEAD->next->next != NULL)

{

for (int i = 0; i< a - 1; i++)

{

q = HEAD;

q1 = q->next;

q2 = q1->next;

while (q2->next != NULL)

{

if (q1->price > q2->price)

{

q->next = q1->next;

q1->next = q2->next;

q2->next = q1;

q2 = q1->next;

q = q->next;

continue;

}

q = q->next;

q1 = q1->next;

q2 = q2->next;

}

if (q1->price > q2->price)

{

q->next = q2;

q2->next = q1;

q1->next = NULL;

}

}

}

q1 = HEAD->next;

FILE\* fp = fopen("bookinformation.txt", "wt");

if (fp == NULL)

{

printf("打开文件失败！\n");

return 0;

}

for (int i = 0; i < a; i++)

{

fprintf(fp, "%s %s %s %s %s %s %g %d\n", q1->bianhao, q1->bookname, q1->writer, q1->leibie, q1->chubanfang, q1->data, q1->price, q1 ->num);

q1 = q1->next;

}

fclose(fp);

}

void Change()

{

int t = 0;

system("cls");

FileRead(head);

struct BOOK\* q = (struct BOOK\*)malloc(sizeof(struct BOOK));

q = head;

char ch[50];

gotoxy(80, 15); printf("请输入需要查询图书的编号或书名：");

scanf("%s", ch);

while (q->next != NULL)

{

if (strcmp(ch, q->next->bianhao) == 0 || strcmp(ch, q->next->bookname) == 0)

{

t++;

struct BOOK\* q1 = (struct BOOK\*)malloc(sizeof(struct BOOK));

q1 = q->next;

system("cls");

struct BOOK\* s = (struct BOOK\*)malloc(sizeof(struct BOOK));

gotoxy(80, 11); printf("请重新输入图书编号（小于10位）：");

scanf("%s", s->bianhao);

gotoxy(80, 13); printf("请重新输入书名（小于25字）：");

scanf("%s", s->bookname);

gotoxy(80, 15); printf("请重新输入作者姓名（小于25字）：");

scanf("%s", s->writer);

gotoxy(80, 17); printf("请重新输入图书类别（小于10字）：");

scanf("%s", s->leibie);

gotoxy(80, 19); printf("请重新输入图书出版方（小于25字）：");

scanf("%s", s->chubanfang);

gotoxy(80, 21); printf("请重新输入图书出版日期（如2018.03.03）：");

scanf("%s", s->data);

gotoxy(80, 23); printf("请重新输入图书价格：");

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

gotoxy(80, 25); printf("请重新输入图书数量：");

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

char a;

Save();

loop:a = getch();

if (a == 'y' || a == 'Y')

{

Succeed();

Sleep(1000);

system("cls");

q->next = s;

s->next = q1->next;

}

else if (a == 'n' || a == 'N')

{

system("cls");

gotoxy(90, 25); printf("修改未保存！");

Sleep(1000);

system("cls");

}

else

{

gotoxy(80, 34); printf("请重新输入y或n！");

goto loop;

}

}

q = q->next;

}

if (t != 0)

{

q = head;

int a = 0;//链表长度计数器

while (q->next != NULL)

{

a++;

q = q->next;

}

q = head->next;

FILE\* fp = fopen("bookinformation.txt", "wt");

if (fp == NULL)

{

printf("打开文件失败！\n");

return 0;

}

for (int i = 0; i < a; i++)

{

fprintf(fp, "%s %s %s %s %s %s %g %d\n", q->bianhao, q->bookname, q->writer, q->leibie, q->chubanfang, q->data, q->price, q ->num);

q = q->next;

}

fclose(fp);

}

else

{

gotoxy(80, 22); printf("这本书还没有入库呢！！！");

}

}

void Check()

{

system("cls");

FileRead(head);

struct BOOK\* q = (struct BOOK\*)malloc(sizeof(struct BOOK));

q = head;

char ch[50];

gotoxy(80, 15); printf("请输入需要查询图书的编号或书名或作者：");

gotoxy(80, 17); scanf("%s", ch);

system("cls");

int t = 0;

while (q->next != NULL)

{

if (strcmp(ch, q->next->bianhao) == 0 || strcmp(ch, q->next->bookname) == 0 || strcmp(ch, q->next->writer) == 0)

{

gotoxy(33, 6); printf("----------------------------------------------------------------------------------------------------------------------------------\n");

gotoxy(40, 7); printf("图书编号 书名 作者 图书类别 出版方 出版日期 图书价格 图书数量\n");

gotoxy(33, 8); printf("----------------------------------------------------------------------------------------------------------------------------------\n");

gotoxy(40, 9 + t); printf("%s", q->next->bianhao);

gotoxy(55, 9 + t); printf("%s", q->next->bookname);

gotoxy(69, 9 + t); printf("%s", q->next->writer);

gotoxy(82, 9 + t); printf("%s", q->next->leibie);

gotoxy(99, 9 + t); printf("%s", q->next->chubanfang);

gotoxy(114, 9 + t); printf("%s", q->next->data);

gotoxy(131, 9 + t); printf("%g", q->next->price);

gotoxy(150, 9 + t); printf("%d\n", q->next->num);

t += 2;

}

q = q->next;

}

if (t == 0)

{

system("cls");

gotoxy(80, 21); printf("这本书还没有添加入库！！！");

}

}

void Succeed()

{

system("cls");

gotoxy(80, 20); printf("---------------------------\n");

gotoxy(80, 22); printf("| 成 功 ! |\n");

gotoxy(80, 24); printf("---------------------------\n");

}

int FRN()

{

FILE \*fp;

if (fp = fopen("bookinformation.txt", "rt") == NULL)

{

return -1;

}

else

{

return 0;

}

}

void menu()

{

head = (struct BOOK\*)malloc(sizeof(struct BOOK));

head->next = NULL;

system("cls");

char str;

JM();

str = getch();

switch (str)

{

case '1':

system("cls");

CL();

system("cls");

gotoxy(80, 25); printf("按任意键返回主界面！");

getch();

menu();

break;

case '2':

if (FRN() == 0)

{

Check();

}

else

{

system("cls");

gotoxy(80, 23); printf("还没添加一本图书信息呢，先去添加一本吧！");

}

gotoxy(80, 25); printf("按任意键返回主界面！");

getch();

menu();

break;

case '3':

menu3();

break;

case '4':

if (FRN() == 0)

{

Change();

}

else

{

system("cls");

gotoxy(80, 23); printf("还没添加一本图书信息呢，先去添加一本吧！");

}

gotoxy(80, 25); printf("按任意键返回主界面！");

getch();

menu();

break;

case '5':

menu5();

break;

case '6':system("cls"); exit(0); break;

default: menu();

}

}

void menu5()

{

system("cls");

if (FRN() == 0)

{

PXJM();

char c;

c = getch();

switch (c)

{

case '1':

FileRead(head);

PX\_b(head);

output(head);

gotoxy(80, 25); printf("按任意键返回浏览界面！");

getch();

menu5();

break;

case '2':

FileRead(head);

PX\_p(head);

output(head);

gotoxy(80, 25); printf("按任意键返回浏览界面！");

getch();

menu5();

break;

case '3':

menu();

break;

default:menu5();

}

}

else

{

system("cls");

gotoxy(80, 23); printf("还没添加一本图书信息呢，先去添加一本吧！");

gotoxy(80, 25); printf("按任意键返回主界面！");

getch();

menu();

}

void menu3()

{

system("cls");

if (FRN() == 0)

{

DelJM();

char c;

c = getch();

switch (c)

{

case '1':

system("cls");

FileRead(head);

DelWriter(head);

system("cls");

gotoxy(80, 23); printf("按任意键返回删除菜单！");

getch();

menu3();

break;

case '2':

system("cls");

FileRead(head);

DelBook(head);

gotoxy(80, 24); printf("按任意键返回删除菜单！");

getch();

menu3();

break;

case '3':

menu();

break;

default:menu3();

}

}

else

{

system("cls");

gotoxy(80, 23); printf("还没添加一本图书信息呢，先去添加一本吧！");

gotoxy(80, 25); printf("按任意键返回主界面！");

getch();

menu();

}

}

int ReCheck(char str[])

{

if (FRN() == 0)

{

struct BOOK\* Head = (struct BOOK\*)malloc(sizeof(struct BOOK));

Head->next = NULL;

FileRead(Head);

struct BOOK\* q = (struct BOOK\*)malloc(sizeof(struct BOOK));

q = Head;

while (q->next != NULL)

{

if (strcmp(str, q->next->bianhao) == 0)

{

return -1;

}

q = q->next;

}

return 0;

}

else

{

return 0;

}

}

1. **运行结果截图**
2. 主界面。

运行程序进入主界面，如图1.1所示，共有6个选项。



1.1 图书管理系统主菜单

（2）录入信息界面。

选择选项1进入信息录入界面，用户按提示输入信息，如图2.1所示。当输入完毕时，提示是否保存信息，如图2.2所示，若输入y则将信息保存到文件，输入n取消保存。然后提示是否继续录入信息，如图2.3所示，选择y继续录入信息，输入n提示按任意键返回主菜单。

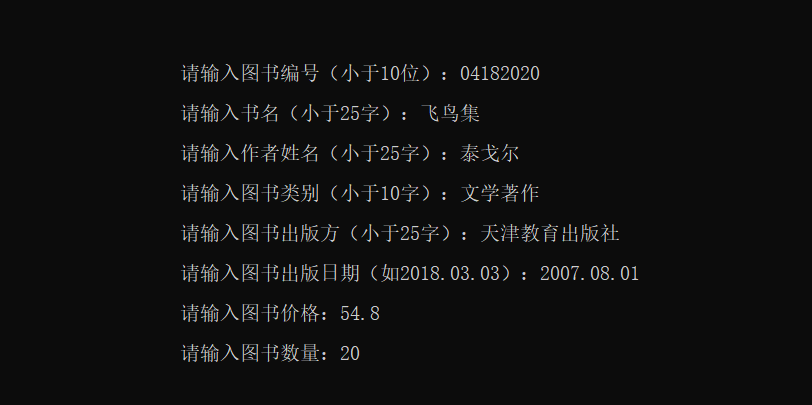


图2.1图书信息录入界面

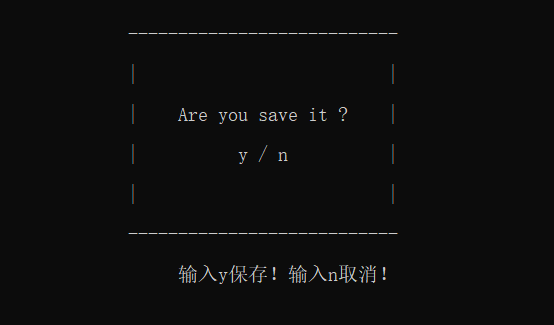


图2.2 保存提示框

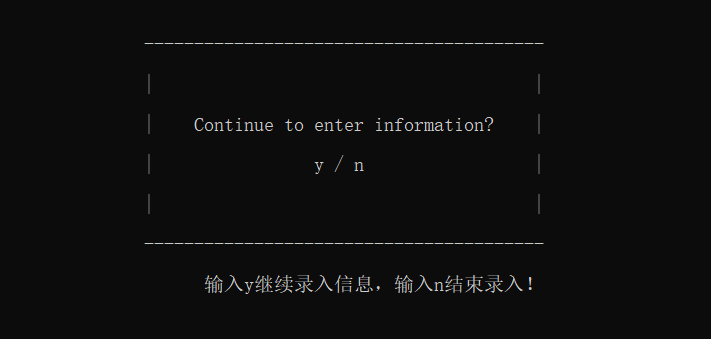


图2.3 是否继续录入信息提示框

（3）图书信息查询界面。

选择选项2进入图书信息查询界面，根据提示信息，可以输入图书编号、书名或作者进行查询，如图3.1、3.2所示。并且如果初次使用没有文件时，提示“还没添加一本图书信息呢，先去添加一本吧！”，如图3.3所示。若输入的信息在文件中不存在则会输出“这本书还没有添加入库！！！”如图3.4所示。

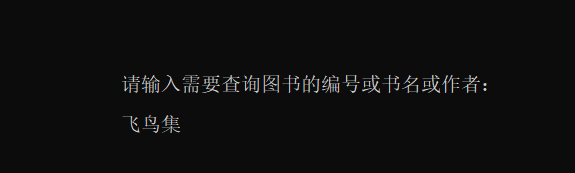


图3.1 查询条件输入

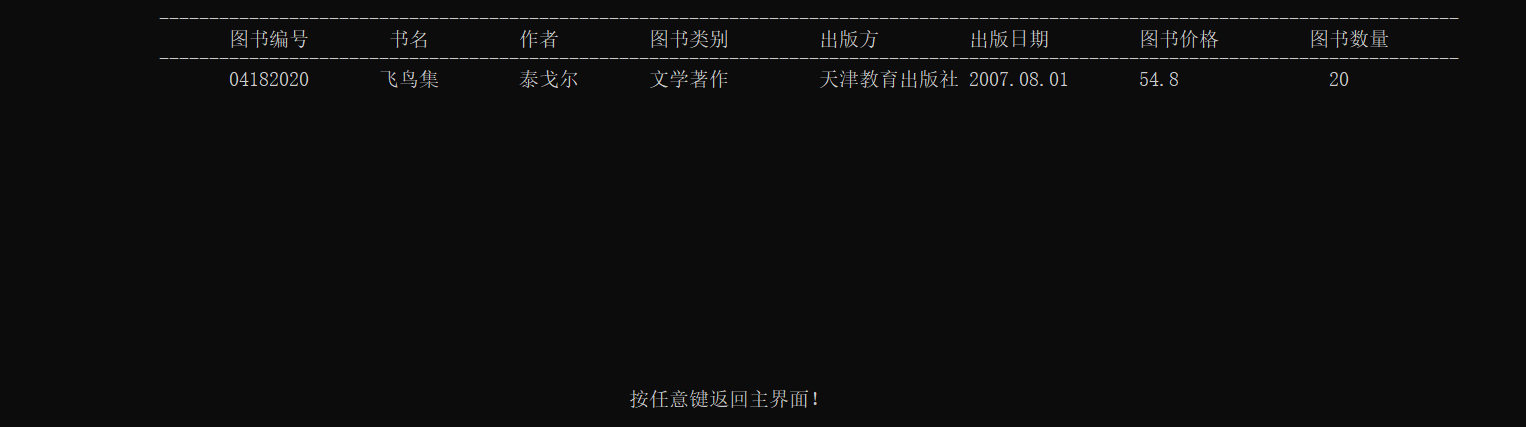


图3.2 查询结果输出

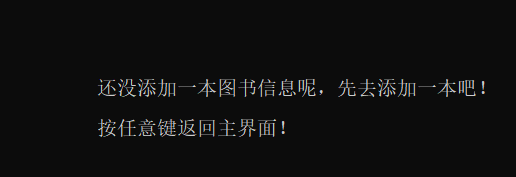


图3.3 初次使用时提示信息

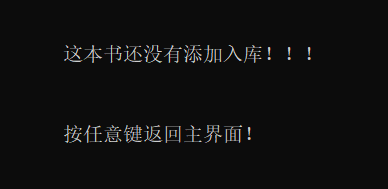
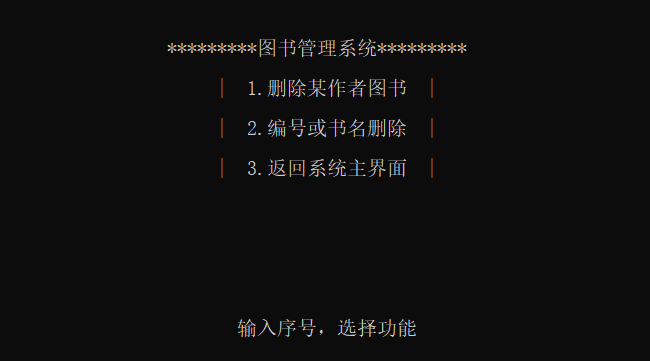


图3.4 输入信息在文件中不存在

（4）图书信息删除界面。

选择选项3进入删除功能选择菜单，如图4.1，选择选项1删除文件中某一作者的图书信息，如图4.2、4.3、4.4、4.5所示。选择选项2删除条件图书信息，如图4.6、4.7所示。



4.1 删除界面



图4.2 删除前文件中的内容

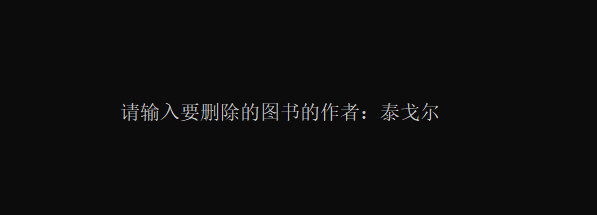


图4.3 删除作者

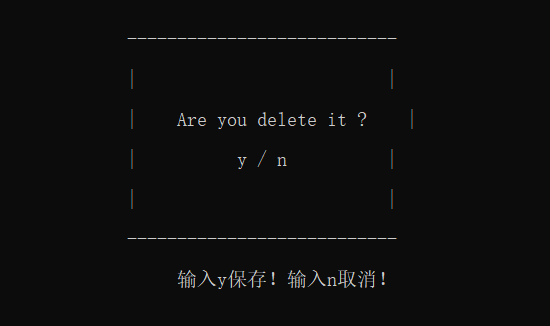


图4.4 确认删除提示框



图4.5 删除作者后文件中的内容

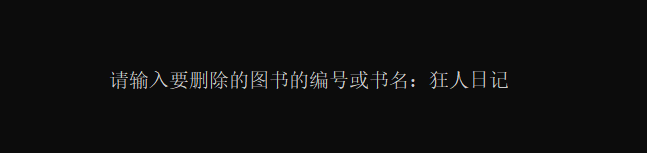


图4.6 删除文件中图书编号或书名信息



图4.7 删除书名后文件的内容

（5）图书信息修改。

选择选项4，进行图书修改界面，若初次使用会“还没添加一本图书信息呢，先去添加一本吧！”，如图3.3所示。输入已存在的图书编号或书名进如修改界面，将数量修改为76，如图5.1。回车后显示是否保存提示框，输入y保存，如图2.2所示。保存成功后按任意键返回主菜单。

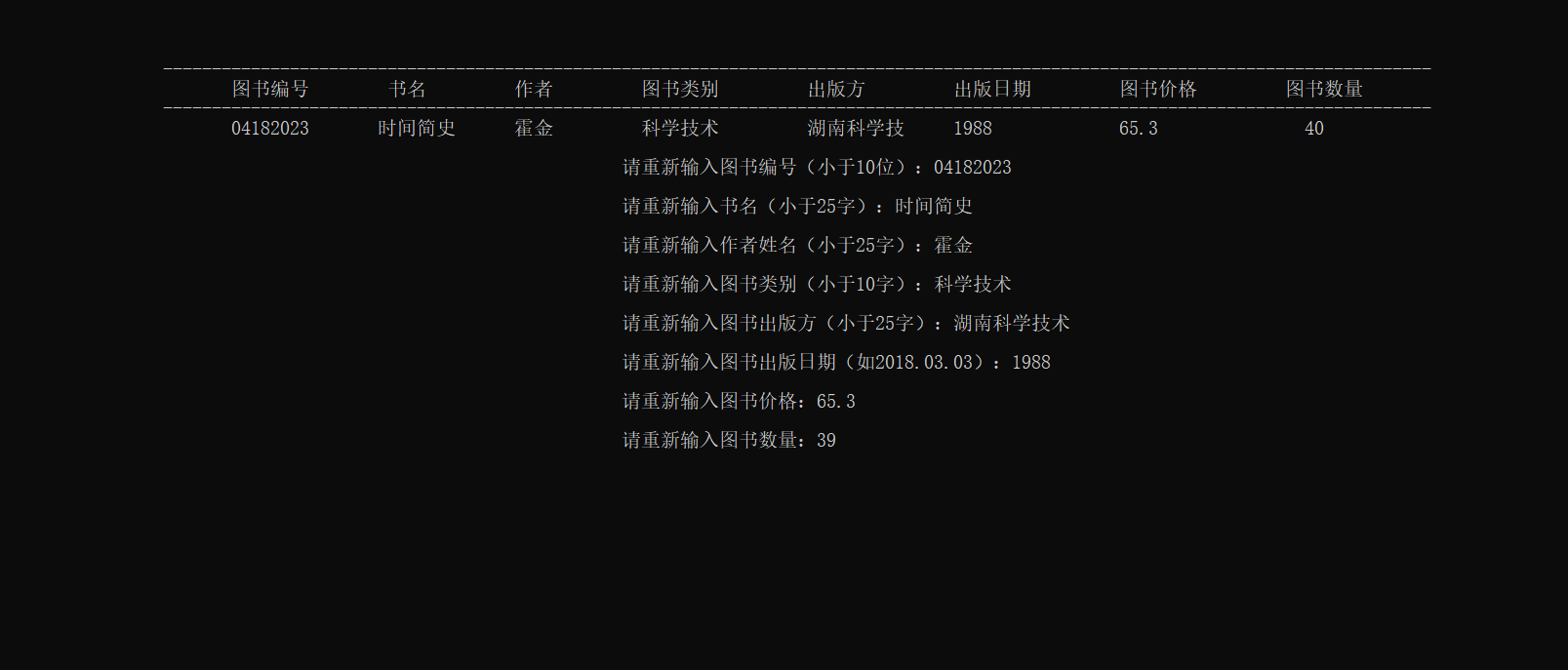


图5.1 修改界面

（6）图书信息总览界面。

选择选项5，进入排序选择菜单，如图6.1所示。选择选项1，将文件中的信息按编号升序排序并输出，如图6.2所示。选择选项2，将文件中的信息按价格升序排序并输出，如图6.3所示。然后按任意键返回排序选择菜单。

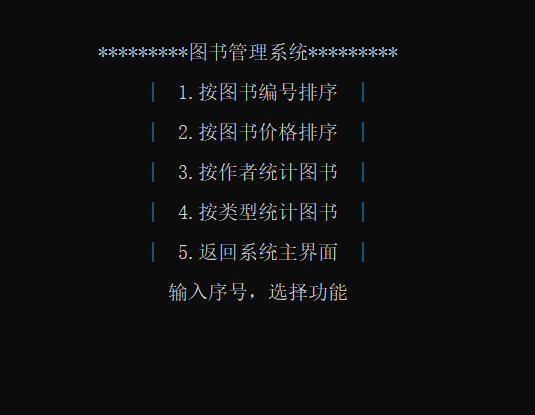


图6.1 排序选择菜单

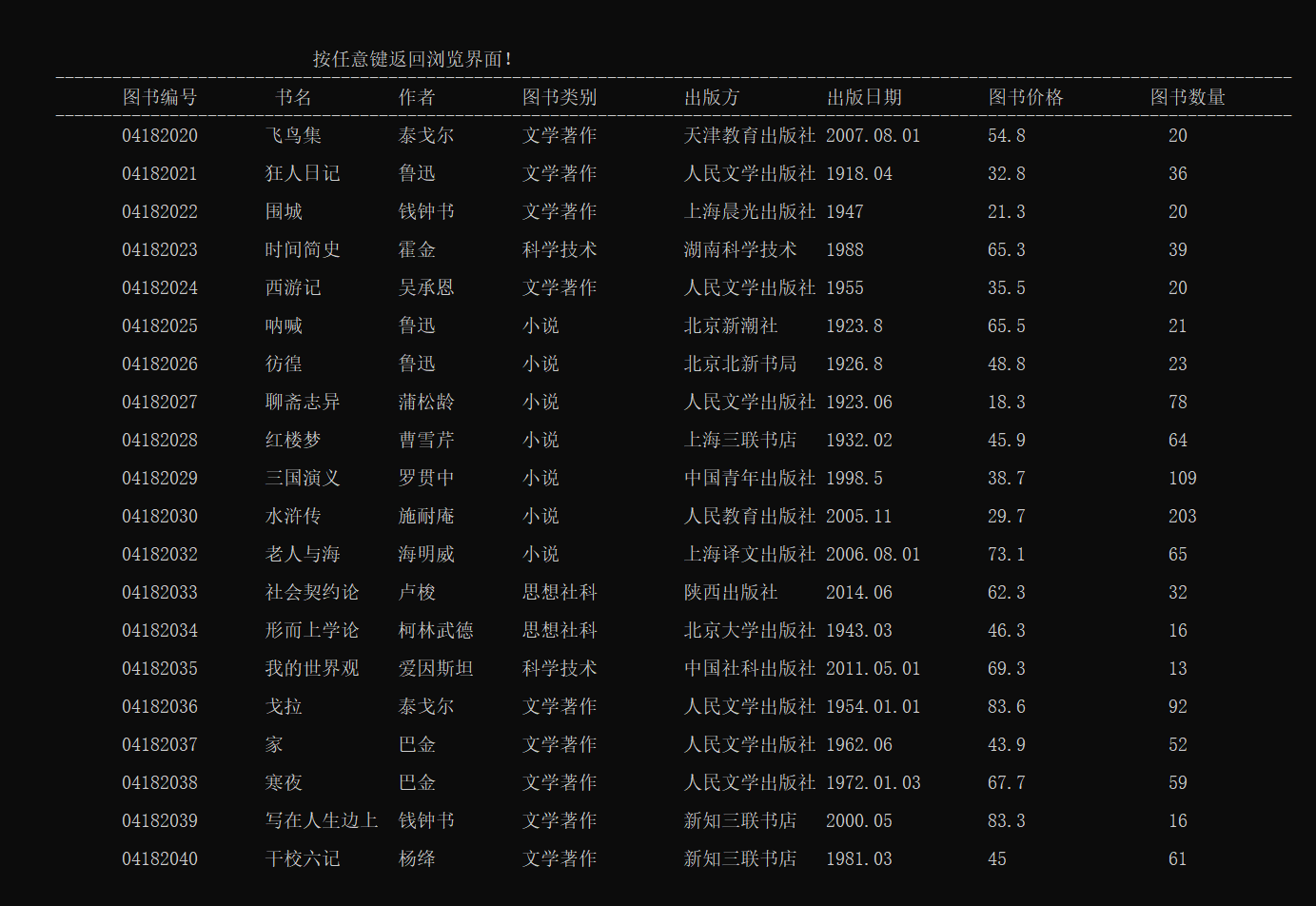


图6.2 按编号升序排序输出

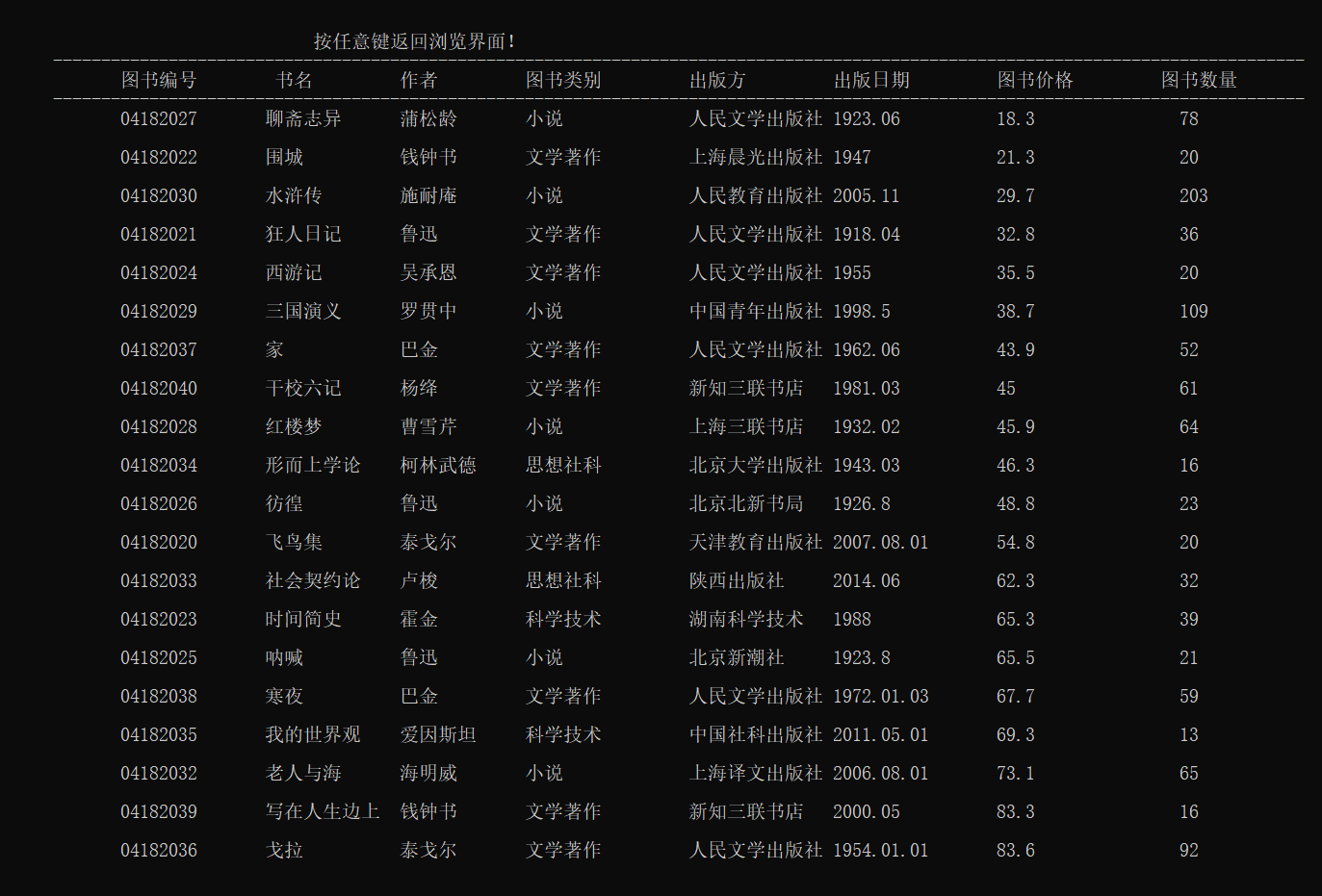


图6.3 按图书价格升序排序输出

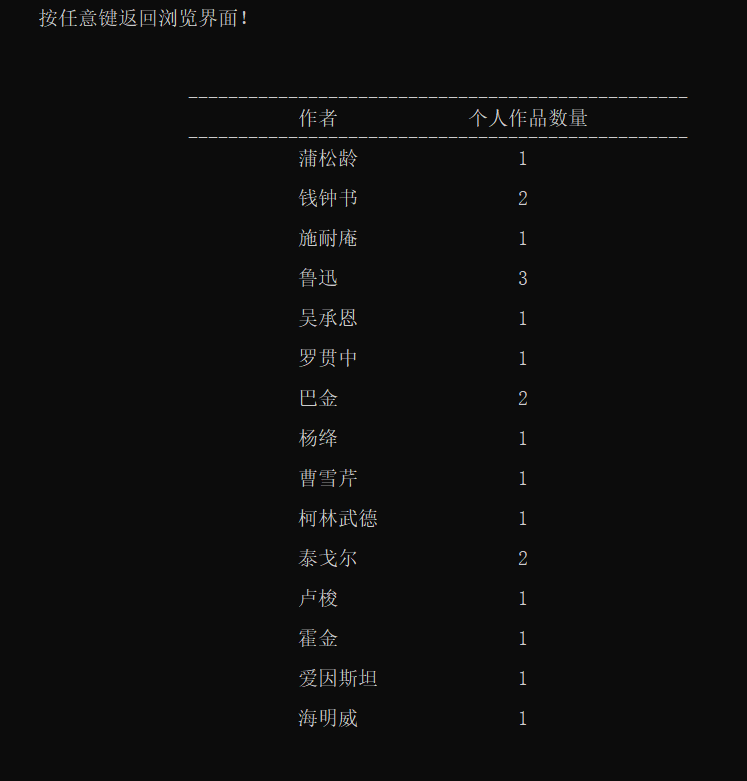


图 6.4 按作者进行统计

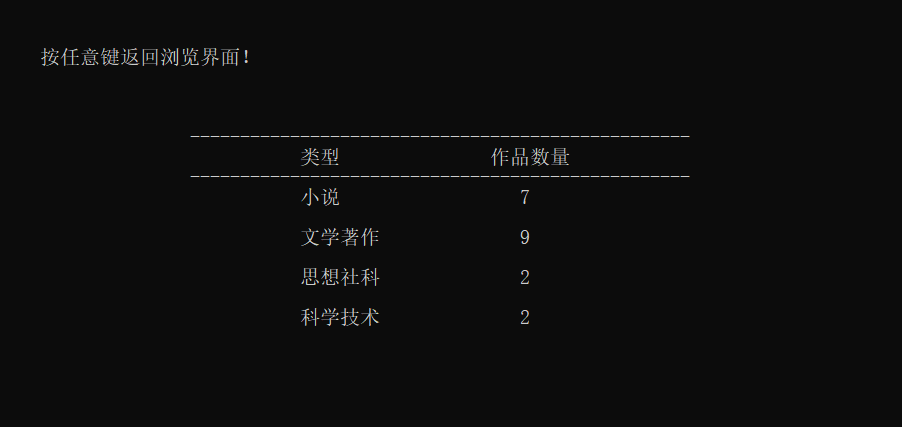


图 6.5 按图书类型进行统计

**四．设计技巧与体会**

设计系统时将系统分为5大功能模块，然后不断计划和编写新的功能模块，并在编写完成后不断调试，写完所有功能后将各个模块结合在一起调试、运行，所以节省了不少时间和精力。

然而编写程序时也不是一帆风顺，其中遇到了不少麻烦，有时候觉得想法没有什么问题，但运行结果却出乎意料，仔细检查后却发现只是一个小小的错误。所以即使有一个微小的错误也会导致程序出错。经过这次编写较大规模程序，我会更加注重细节。

其次在我将所有模块拼接在一起时，用了不少跳转语句，导致程序理解起来很难，而且主程序比较复杂，之后通过函数递归调用的方式大大简化主函数，也更易读懂程序，因此在以后的编写程序中我会多考虑更加简洁、更加方便的算法和结构。

当然，我认为这个图书管理系统也有很多不足之处，虽然经过了无数次调试，解决了不少明显的bug，但依然存在很多缺点，比如输出界面不合理：可能会存在某一项信息过长而无法全部显示的问题。又比如很多界面也都太简陋啦。。。可能我不太适合设计一些漂亮的界面，不过都是要练习的嘛。最后也肯定会有一些隐藏的bug还没有被我发现。每个bug的存在都是因为考虑不周到的问题，通过这次编写程序也使得我拥有更好的“视野”，让我懂得大局观，让我更加成熟。