**C语言程序设计结课作业**

**图书信息管理系统**

**学 院： 软件学院**

**班 级： 2106**

**学生姓名： 王忆嘉**

**日 期： 2021/12/16**

**指导教师： 赵宏**

**目 录**

项目简介……………………………………………………………………………… 3

一、数据需求………………………………………………………………………… x

二、功能需求………………………………………………………………………… x

三、操作流程设计…………………………………………………………………… x

四、总体设计（文件及函数）…………………………………………………… 4

五、函数及接口设计……………………………………………………………… 5

六、运行截图………………………………………………………………………… 10

七、程序源码………………………………………………………………………… 14

致 谢…………………………………………………………………………………… 39

**项目简介**

本系统为面向图书管理员的图书管理系统，实现图书信息的查询与更新、图书的借阅与归还。

1. **数据需求**

图书编号、书名、作者、出版社、价格、库存量、借出数

1. **功能需求**
2. 浏览（全局浏览）
3. 查找（ID的精确查找，书名、作者、出版社的模糊查找）
4. 增加
5. 删除
6. 修改
7. 借阅
8. 归还
9. **流程设计**
10. 循环显示菜单：获取用户的输入。
11. 进入功能模块，执行并输出到控制台。
12. 结束前询问是否保存到文件。
13. **总体设计（文件及函数）**

|  |  |  |
| --- | --- | --- |
| 源文件 | 函数原型 | 功能 |
| main.c | int main() | 总控，选择进入功能模块 |
| management.c | void browse(BOOK \*head) | 打印链表的全部内容 |
| void search(BOOK \*head) | 根据ID精确查找结点，  根据“书名、作者、出版社”模糊查找结点 |
| BOOK \*add(BOOK \*head) | 新增结点 |
| BOOK \*\_delete  (BOOK \*head) | 查找并删除结点 |
| void change(BOOK \*head) | 查找并修改结点 |
| void borrow(BOOK \*head) | 借阅：更改指定结点“库存”和“借出数”的值 |
| void back(BOOK \*head) | 归还：更改指定结点“库存”和“借出数”的值 |
| util.c | int menu() | 显示菜单，获取输入 |
| BOOK \*creatlinkedlist  (void) | 读取文件，头插法创建链表 |
| FILE \*openfile  (char filename[50],  char filetype[5]) | 打开指定文件 |
| void save(BOOK \*head) | 保存至文件 |
| BOOK \*sort(BOOK \*head) | 排序 |
| BOOK \*getin(BOOK \*p) | 获取一个结点内容的输入 |
| void putout(BOOK \*p) | 打印一个结点 |

**五、函数和接口设计**

1.主函数

函数原型：int main(）

功能：读取文件，创建并排序链表，选择进入功能模块，保存至文件

参数：无

返回值：0

调用：creatlinkedlist()、sort(head)、menu()、save(head)

2.浏览

函数原型：void browse(BOOK \*head)

功能：浏览链表并打印

参数：BOOK \*head

返回值：无

调用：putout(p)

3.查找

函数原型：void search(BOOK \*head)

功能：根据输入（检查输入）查找并打印所有相关结点。ID精确查找，书名、作者、

出版社模糊查找

参数：BOOK \*head

返回值：无

调用：putout(p)、search(head)（递归）

要求：下划线代替空格

4.增加

函数原型：BOOK \*add(BOOK \*head)

功能：获取输入（检查输入），新增结点至链表

参数：BOOK \*head

返回值：BOOK \*head

调用:p=getin(p)、putout(p)

要求：下划线代替空格

5.删除

函数原型：BOOK \*\_delete(BOOK \*head)

功能：根据输入ID（检查输入）查找并删除结点

参数：BOOK \*head

返回值：BOOK \*head

调用:无

6.修改

函数原型：void change(BOOK \*head)

功能：根据输入ID（检查输入）查找并修改结点内容

参数：BOOK \*head

返回值：无

调用：p=getin(p)、putout(p)

7.借阅

函数原型：void borrow(BOOK \*head)

功能：根据输入ID（检查输入）查找并修改库存量和借出数

参数：BOOK \*head

返回值：无

调用：无

8.归还

函数原型：void back(BOOK \*head)

功能：根据输入ID（检查输入）查找并修改库存量和借出数

参数：BOOK \*head

返回值：无

调用：无

9.菜单

函数原型：int menu(void)

功能：显示菜单，获取输入（检查输入）

参数：无

返回值：int choose

调用：无

10.创建链表

函数原型：BOOK \*creatlinkedlist(void)

功能：读取文件、头插法创建链表

参数：无

返回值：BOOK \*head

调用：file = openfile("Book Information.txt","r")

11.打开文件

函数原型：FILE \*openfile(char filename[50],char filetype[5])

功能：打开文件

参数：char filename[50]、char filetype[5]

返回值：FILE \*file

调用：无、

要求：文件名不要过长，防止字符数组越界

1. 保存到文件

函数原型：void save(BOOK \*head,char filename[50],char filetype[5]);

功能：将链表内容打印到文件

参数：BOOK \*head、char filename[50]、char filetype[5]

返回值：无

调用：openfile(filename,filetype)

要求：文件名不要过长，防止字符数组越界

13.排序

函数原型：BOOK \*sort(BOOK \*head)

功能：使用冒泡算法，将链表的结点按ID的值升序排列

参数：BOOK \*head

返回值：BOOK \*head

调用：无

14.获得输入

函数原型：BOOK \*getin(BOOK \*p)

功能：获得输入（书名 ，作者，出版社，价格，库存，借出），检查输入

参数：BOOK \*p

返回值：BOOK \*p

调用：无

14.打印

函数原型：void putout(BOOK \*p)

功能：打印一个结点的内容

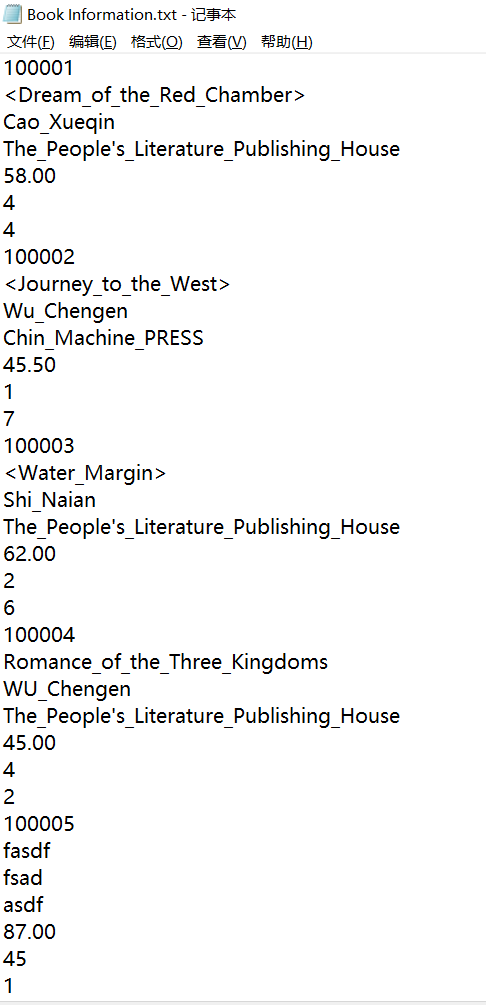
参数：BOOK \*p

返回值：无

调用：无

**六、运行截图**

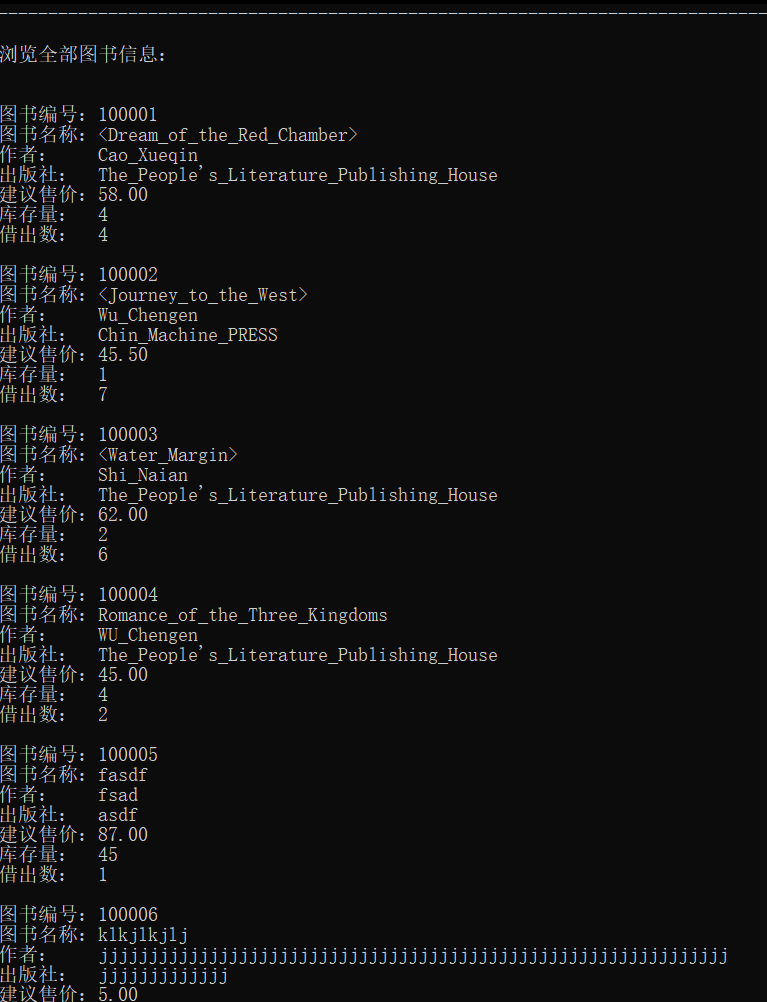
“Book Information.txt”

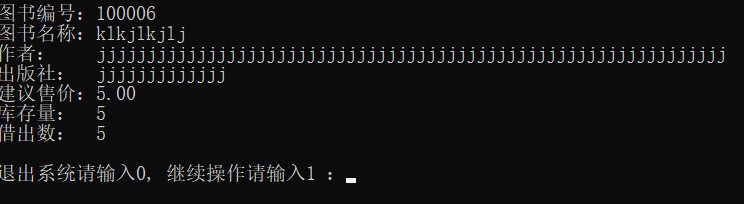


菜单：

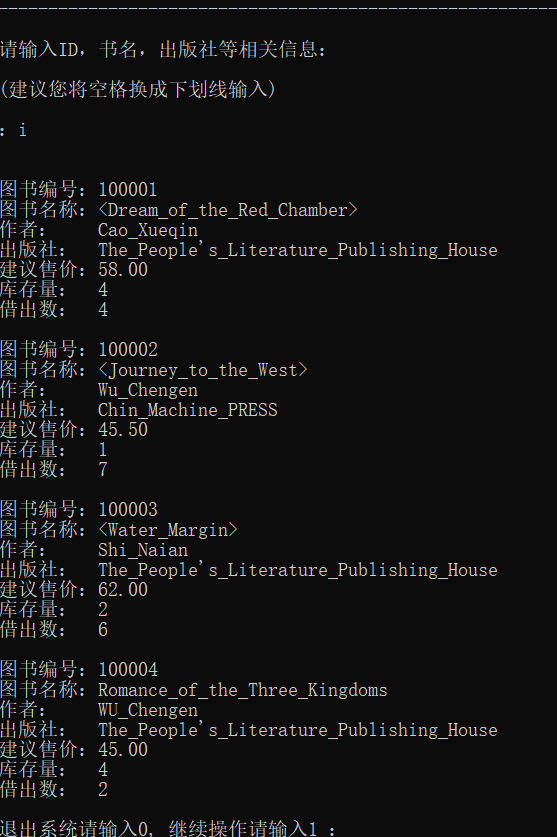


1. 浏览：

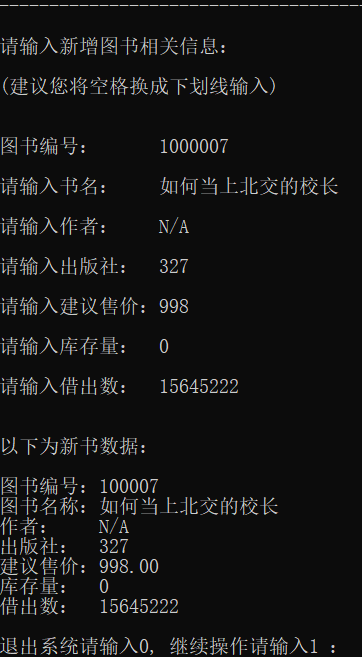




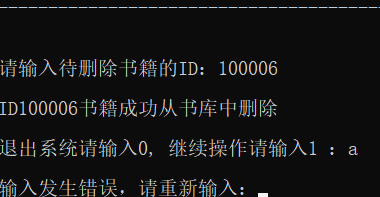
1. 查找：



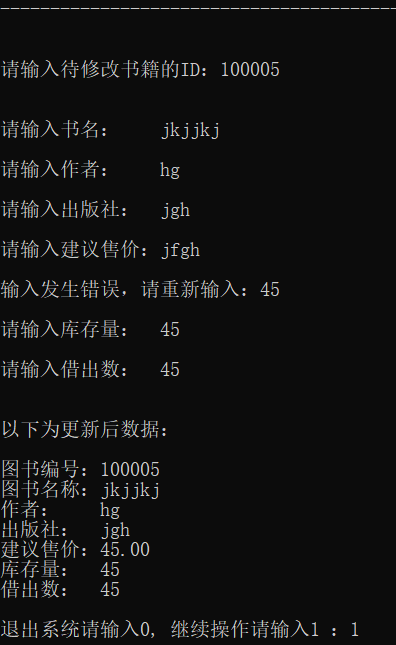
1. 新增：



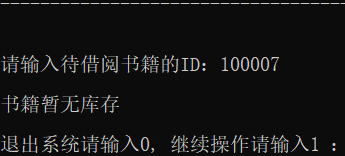
1. 删除：

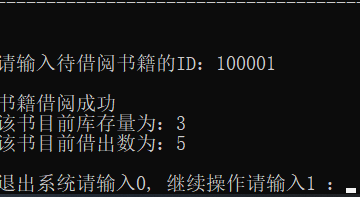


1. 修改：

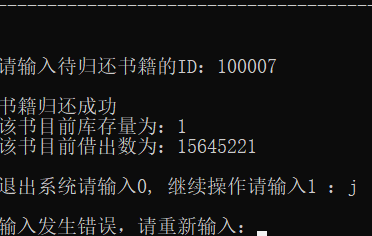


1. 借阅：

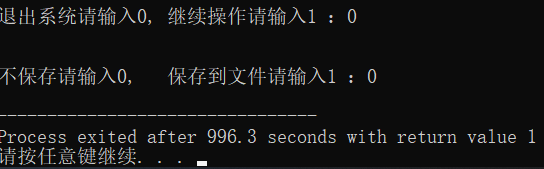




1. 归还：



结束：



**七、程序源码**

**1.book.h**

#ifndef H\_BOOK\_H

#define H\_BOOK\_H

#include<stdio.h>

#include<stdlib.h>

#include<ctype.h>

#include<string.h>

#include<time.h>

struct book

{

int id;

char bookname[50];

char author[50];

char publisher[50];

float price;

int stock;

int out;

struct book \*ptr;

};

typedef struct book BOOK;//简称为BOOK

//符号常量定义

#define LEN sizeof(struct book)

//函数原型声明

int menu(void);

void browse(BOOK \*head);

BOOK \*creatlinkedlist(void);

FILE \*openfile(char filename[50],char filetype[5]);

void save(BOOK \*head,char filename[50],char filetype[5]);

BOOK \*sort(BOOK \*head);

BOOK \*add(BOOK \*head);

BOOK \*\_delete(BOOK \*head);

void change(BOOK \*head);

void borrow(BOOK \*head);

void back(BOOK \*head);

BOOK \*getin(BOOK \*p);

void search(BOOK \*head);

void putout(BOOK \*p);

#endif

1. **main.c**

#include "book.h"

int main()

{

int choose;

BOOK \*head;

//创建链表，获得头指针

head=creatlinkedlist();

//将链表排序

head=sort(head);

do

{

choose=menu(); //从菜单获取指令

switch(choose)

{

case 1: browse(head);break;

case 2: search(head);break;

case 3: head=add(head);break;

case 4: head=\_delete(head);break;

case 5: change(head);break;

case 6: borrow(head);break;

case 7: back(head);break;

}

printf("\n\n退出系统请输入0, 继续操作请输入1 ：");

while(scanf("%d",&choose)==0||choose!=choose\*choose)

{printf("\n输入发生错误，请重新输入：");rewind(stdin);}

}while(choose==1);

//询问是否保存到文件

printf(" \n\n不保存请输入0, 保存到文件请输入1 ：");

while(scanf("%d",&choose)==0||choose!=choose\*choose)

{printf("\n输入发生错误，请重新输入：");rewind(stdin);}

if(choose==0)

{

exit(1);

}

else

{

//保存到文件

save(head,"Book Information.txt","w");

}

return 0;

}

1. **management.c**

#include "book.h"

//函数功能：浏览链表

void browse(BOOK \*head)

{

printf("\n\n\n-----------------------------------------------------------------------------------\n\n");

printf("浏览全部图书信息：\n");

BOOK \*p;

p=head;

while(p!=NULL)

{

putout(p);//打印一个结点

p=p->ptr;

}

}

//函数功能：模糊查找

void search(BOOK \*head)

{

BOOK \*p;

p=head;

char in[50];

char id[50];

int sgn0;

int choose;

printf("\n\n-----------------------------------------------------------------------------------\n\n");

printf("请输入ID，书名，出版社等相关信息：\n\n");

printf("(建议您将空格换成下划线输入)\n\n：");

scanf("%s",in);//获取用户输入

sgn0=0;

while(p!=NULL)/\*第一层循环，遍历链表\*/

{

sprintf(id,"%d",p->id);

if(strcmp(id,in)==0)

{

sgn0=1;

putout(p);//打印一个结点

}

else if(strstr(p->bookname,in)!=NULL)

{

sgn0=1;

putout(p);//打印一个结点

}

else if(strstr(p->author,in)!=NULL)

{

sgn0=1;

putout(p);

}

else if(strstr(p->publisher,in)!=NULL)

{

sgn0=1;

putout(p);

}

p=p->ptr;

}

if(sgn0==0)

{

printf("并未查找到相关信息\n\n");

printf("继续查找请输入1，退出查找请输入0：");

while(scanf("%d",&choose)==0||choose<0||choose>1)

{printf("\n输入发生错误，请重新输入：");rewind(stdin);}

switch (choose)

{

case 0:break;

case 1:search(head);//递归调用

}

}

}

//函数功能：新增结点

BOOK \*add(BOOK \*head)

{

BOOK \*p=head;

//获得结点的个数

int num=0;

while(p!=NULL)

{

num+=1;

p=p->ptr;

}

p=(BOOK\*)malloc(LEN);//为p分配内存

printf("\n\n-----------------------------------------------------------------------------------\n\n");

printf("请输入新增图书相关信息：\n\n");

printf("(建议您将空格换成下划线输入)\n\n\n");

printf("图书编号： %d",1000001+num);

p->id=100001+num;

p=getin(p);//获得输入

printf("\n\n以下为新书数据：");

putout(p);//打印一个结点

p->ptr=head;

head=p;

head=sort(head);//将链表重新排序

return(head);

}

//函数功能：删除结点

BOOK \*\_delete(BOOK \*head)

{

BOOK \*p1,\*p2,\*p0;

p1 = head;

printf("\n\n\n-----------------------------------------------------------------------------------\n\n");

if(head==NULL)

{

printf("\n书库中空空如也......");

return;

}

int id;

printf("\n请输入待删除书籍的ID：");

while(scanf("%d",&id)!=1)

{printf("\n输入发生错误，请重新输入：");rewind(stdin);}

if(head->id==id)

{

head=head->ptr;

printf("\nID%d书籍成功从书库中删除");

p0=head;

while(p0!=NULL)

{

p0->id-=1;

p0=p0->ptr;

}

}

else

{

p2=head;

p1=head->ptr;

/\*while(p1->id!=id&&p1!=NULL) 当p1为空指针时不可以指向结构体的成员 ！！！无效地址！！！\*/

while(p1!=NULL&&p1->id!=id) /\*换一下判断顺序，当p1为空指针时会直接结束循环，p1->id不会执行 \*/

{

p1=p1->ptr;

p2=p2->ptr;

}

if(p1==NULL)

{

printf("\n未找到您所输入的ID");

}

else

{

p2->ptr=p1->ptr;

printf("\nID%d书籍成功从书库中删除",id);

p0=p1->ptr;

while(p0!=NULL)

{

p0->id-=1;

p0=p0->ptr;

}

}

}

return(head);

}

//函数功能：修改结点的值

void change(BOOK \*head)

{

BOOK \*p = head;

printf("\n\n\n-----------------------------------------------------------------------------------\n\n");

if(head==NULL)

{

printf("\n书库中空空如也......");

return;

}

int id;

printf("\n请输入待修改书籍的ID：");

while(scanf("%d",&id)!=1)

{printf("\n输入发生错误，请重新输入：");rewind(stdin);}

while(p!=NULL&&p->id!=id) /\*当p1为空指针时会直接结束循环，p1->id不会执行 \*/

{

p=p->ptr;

}

if(p==NULL)

{

printf("\n未找到您所输入的ID");

}

else

{

p=getin(p);//获得输入

printf("\n\n以下为更新后数据：");

putout(p);//打印一个结点

}

}

//函数功能：借阅

void borrow(BOOK \*head)

{

BOOK \*p = head;

printf("\n\n\n-----------------------------------------------------------------------------------\n\n");

if(head==NULL)

{

printf("\n书库中空空如也......\n\n");

return;

}

int id;

printf("\n请输入待借阅书籍的ID：");

while(scanf("%d",&id)!=1)

{printf("\n输入发生错误，请重新输入：");rewind(stdin);}

while(p!=NULL&&p->id!=id) //当p1为空指针时会直接结束循环，p1->id不会执行

{

p=p->ptr;

}

if(p==NULL)

{

printf("\n未找到您所输入的ID");

}

else if(p->stock==0)

{

printf("\n书籍暂无库存");

}

else

{

p->out+=1;

p->stock-=1;

printf("\n书籍借阅成功\n");

printf("该书目前库存量为：%d\n",p->stock);

printf("该书目前借出数为：%d",p->out);

}

}

//函数功能：归还

void back(BOOK \*head)

{

BOOK \*p = head;

printf("\n\n\n-----------------------------------------------------------------------------------\n\n");

if(head==NULL)

{

printf("\n并无书籍信息，您应该是还错图书馆了");

return;

}

int id;

printf("\n请输入待归还书籍的ID：");

while(scanf("%d",&id)!=1)

{printf("\n输入发生错误，请重新输入：");rewind(stdin);}

while(p!=NULL&&p->id!=id) //当p1为空指针时会直接结束循环，p1->id不会执行

{

p=p->ptr;

}

if(p==NULL)

{

printf("\n未找到您所输入的ID");

}

else if(p->out==0)

{

printf("\n...并没有书籍外借，您应该是还错图书馆了");

}

else

{

p->out-=1;

p->stock+=1;

printf("\n书籍归还成功\n");

printf("该书目前库存量为：%d\n",p->stock);

printf("该书目前借出数为：%d",p->out);

}

}

1. **util.c**

#include "book.h"

//菜单 ：获取输入，返回指令

int menu(void)

{

int choose;

printf("\n\n\n\n");

printf(" 中华人民共和国国家典藏馆欢迎您\n");

printf(" ------------------------------\n\n\n\n\n");

printf(" 指令列表：\n");

printf(" ---------\n\n\n\n");

printf(" [1].浏览全部书籍信息 [2].查找书籍 [3].新增书籍\n");

printf(" -------------------- ------------ ------------\n\n\n\n");

printf(" [4].删除书籍 [5].修改书籍信息 [6].书籍借阅\n");

printf(" ------------ ---------------- ------------\n\n\n\n");

printf(" [7].书籍归还\n");

printf(" ------------\n\n\n\n\n");

printf(" 请输入您的指令：");

while(scanf("%d",&choose)==0||choose<1||choose>7)

{printf("\n输入发生错误，请重新输入：");rewind(stdin);}

return choose;

}

//函数功能：创建链表-返回头指针

BOOK \*creatlinkedlist(void)

{

FILE \*file;

file = openfile("Book Information.txt","r");//打开文件

BOOK \*head,\*p;

char c[50];

int sgn;

head=NULL;

p=(BOOK\*)malloc(LEN);//头插法创建链表

sgn=fscanf(file,"%d%s%s%s%f%d%d",&p->id,p->bookname,p->author,p->publisher,&p->price,&p->stock,&p->out);

while(sgn==7)//利用fscanf返回值控制循环

{

p->ptr=head;

head=p;

p=(BOOK\*)malloc(LEN);

sgn=fscanf(file,"%d%s%s%s%f%d%d",&p->id,p->bookname,p->author,p->publisher,&p->price,&p->stock,&p->out);

}

fclose(file);//关闭文件

return(head);

}

//函数功能：打开文件

FILE \*openfile(char filename[50],char filetype[5])

{

FILE \*file;

file=fopen(filename,filetype);

if(file==NULL)

{

printf("openfile: Failed to open the txt file.\n");

exit(1);

}

// else

// { printf("\nopenfile: Succeed to open the txt file.\n");}

return(file);

}

//函数功能：保存到文件

void save(BOOK \*head,char filename[50],char filetype[5])

{

FILE\*file;

file = openfile(filename,filetype);//打开文件

BOOK \*p=head;

while(p!=NULL)

{

fprintf(file,"%d\n%s\n%s\n%s\n%.2f\n%d\n%d\n",

p->id,p->bookname,p->author,p->publisher,p->price,p->stock,p->out);

p=p->ptr;

}

fclose(file);

// printf("\nsave:succeed to save the file.\n");

}

//函数功能：排序

BOOK \*sort(BOOK \*head)

{

BOOK \*p,\*p0,\*p1,\*p2,\*temp,\*a,\*b;

p=head;

int i=0,j=0,num=0,sgn1=0,sgn0=0;

//获得结点的个数

while(p!=NULL)

{

num+=1;

p=p->ptr;

}

//冒泡排序

for(i=1;i<num;i++)

{

a=head;

b=head->ptr;

sgn0=0;//第一轮循环未发生交换，sgn0值为0

for(j=0;j<num-i;j++)

{

sgn1=0;//两个结点未发生交换时，sgn1值为0

if((a->id)>(b->id))//将结点按id的升序排列

{

sgn1=1;//发生交换，sgn1，sgn0值更改为1

sgn0=1;

if(j==0)//1、2结点的交换涉及到头指针，单独操作

{

p0=head;

p1=p0->ptr;//p0,p1,p2指向前三个结点

p2=p1->ptr;

head=p1;

p0->ptr=p2;//前两个结点交换

p1->ptr=p0;

temp=p1;

p1=p0; // 恢复p0,p1,p2的顺序

p0=temp;

}

else

{

p0->ptr=p2;

p1->ptr=p2->ptr;//后两个结点交换

p2->ptr=p1;

temp=p1;

p1=p2; // 恢复p0,p1,p2的顺序

p2=temp;

p0=p0->ptr;

p1=p1->ptr;//p0,p1,p2向后位移一个结点

p2=p2->ptr;

}

}

if(sgn1==1)//若两个结点发生交换，恢复a,b的顺序

{

temp=a;

a=b;

b=temp;

}

a=a->ptr;

b=b->ptr;//判断完两个结点后，a,b向后位移一个结点

}

if(sgn0==0)

{

break;//如果第一轮没有发生交换，排序结束

}

}

return(head);

}

//函数功能：获得输入（书名 ，作者，出版社，价格，库存，借出）

BOOK \*getin(BOOK \*p)

{

printf("\n\n请输入书名： ");

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

while(strlen(p->bookname)>=50)

{

printf("\n输入过长，请重新输入："); //防止字符数组长度溢出

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

}

printf("\n请输入作者： ");

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

while(strlen(p->author)>=50)

{

printf("\n输入过长，请重新输入：");

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

}

printf("\n请输入出版社： ");

scanf("%s",p->publisher);

while(strlen(p->publisher)>=50)

{

printf("\n输入过长，请重新输入：");

scanf("%s",p->publisher);

}

printf("\n请输入建议售价：");

while(scanf("%f",&p->price)!=1)

{printf("\n输入发生错误，请重新输入：");rewind(stdin);}//误输字符串后，字符串会停留在缓冲区，

printf("\n请输入库存量： ");

while(scanf("%d",&p->stock)!=1)

{printf("\n输入发生错误，请重新输入：");rewind(stdin);}//清空缓冲区防止陷入死循环

printf("\n请输入借出数： ");

while(scanf("%d",&p->out)!=1)

{printf("\n输入发生错误，请重新输入：");rewind(stdin);}

return (p);

}

//函数功能：打印一个结点

void putout(BOOK \*p)

{

printf("\n\n图书编号：%d\n图书名称：%s\n作者： %s\n出版社： %s\n建议售价：%.2f\n库存量： %d\n借出数： %d",

p->id,p->bookname,p->author,p->publisher,p->price,p->stock,p->out);

}

**［致谢］**

感谢赵宏老师细致而生动的讲解，

感谢老师耐心地解答我所问的各种奇怪的问题。