#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<windows.h>

// #include"book.h"

//关于图书功能定义5个常量，分别用来定义书名、出版社、出版日期、作者和ISBN的最大长度

#define MAX\_BOOK\_NAME 20 //书名最大长度

#define MAX\_PUBLISHER 20 //出版社最大长度

#define MAX\_DATE 20 //出版日期最大日期

#define MAX\_AUTHOR 20 //作者最大长度

#define MAX\_ISBN 20 //ISBN 最大长度

typedef struct \_book\_info

{

char book\_name[MAX\_BOOK\_NAME];

char author[MAX\_AUTHOR];

char publisher[MAX\_PUBLISHER];

char pub\_date[MAX\_DATE];

char ISBN[MAX\_ISBN];

int pages;

}book\_info;

typedef struct \_book

{

book\_info bi;

struct \_book\* next;

}book;

void init\_book();

void load\_books();

void add\_book();

void view\_book();

void delete\_book();

void save\_books();

void clear\_books();

void search\_book\_by\_name();

void search\_book\_by\_author();

void search\_book\_by\_publisher();

void search\_book\_by\_pubdate();

void search\_book\_by\_isbn();

int findstr(char\* source,char\* str);

void save\_books\_to\_file();

book\* get\_last\_book();

book\* get\_previous\_book(book\* p);

void input\_book(book\_info\* info);

void show\_book(book\_info\* info);

// #include"user.h"

#define MAX\_USERNAME 10

#define MAX\_PASSWORD 10

typedef enum \_USER\_TYPE

{

ADMIN=0,

USER

}USER\_TYPE;

typedef struct \_user\_info

{

char username[MAX\_USERNAME];

char password[MAX\_PASSWORD];

USER\_TYPE user\_type;

}user\_info;

typedef struct \_user

{

user\_info ui;

struct \_user\* next;

}user;

void view\_user();

void init\_user();

void load\_users();

USER\_TYPE login();

void add\_user();

void search\_user();

void save\_users();

void clear\_users();

void save\_users\_to\_file();

user\* get\_last\_user();

user\* get\_previous\_user(user\*p);

user\* find\_user(char\* name);

void show\_user(user\_info\* info);

void input\_user(user\_info\* info);

void delete\_user(user\* p);

void update\_user(user\* p);

// #include"managenment.h"

void dongtai1();

void show\_admin\_menu();

void show\_search\_book\_menu();

void show\_user\_menu();

void admin\_exit();

void user\_exit();

#define BOOK\_FILE "books.dat"

book\* first\_book=NULL;//book结构体链表的头结点

void add\_book()//增添图书信息

{

char try\_again='Y';

book\* p=NULL;

book\* new\_book=NULL;

while(try\_again=='Y'||try\_again=='y')

{

new\_book=(book\*)malloc(sizeof(book));//创建一个new\_book

memset(new\_book,0,sizeof(book));//初始化new\_book

new\_book->next=NULL;

printf("->新增图书...\n");

input\_book(&(new\_book->bi));//调用函数input\_book()为new\_book赋值

p=get\_last\_book();//调用get\_last\_book(),取得链表中最后一个节点，赋值给p

if(p==NULL)//如果链表为空

{

first\_book=new\_book;//将new\_book赋值给头结点

}

else

{

p->next=new\_book;//将p连接到最后一个节点之后

}

printf("->继续添加图书吗?(y or n)");

getchar();

try\_again=getchar();

}

}

void view\_book()//浏览图书信息

{

book\* p=NULL;

char input='Y';

int count=0;

while(input=='Y'||input=='y')

{

count=0;

p=first\_book;

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

printf("|序号| 书名 | 作者 | ISBN |\n");

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

while(p!=NULL)

{

count++;

printf("|%4d|%20s|%21s|%19s|\n",count,p->bi.book\_name,p->bi.author,p->bi.ISBN);

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

if(count%5==0&&p->next!=NULL)

{

printf("->显示下一页吗?(y or n):");

getchar();

input=getchar();

if(input!='Y'&&input!='y')

{

break;

}

}

p=p->next;

}

printf("->再次浏览图书吗?(y or n):");

getchar();

input=getchar();

}

}

void search\_book\_by\_name()//按书名查找图书信息

{

book\* p=NULL;

char s[MAX\_BOOK\_NAME]={0};

char input='Y';

int count=0;

int i=0;

printf("->查找图书...\n");

while(input=='Y'||input=='y')

{

count=0;

p=first\_book;//p指向第一个节点

memset(s,0,MAX\_BOOK\_NAME);//清空s

printf("->请输入书名(最大长度为 %d):",MAX\_BOOK\_NAME);

scanf("%s",s);

while(p!=NULL)//p不为空时，调用findstr函数查找书名中是否包含输入的字符串s，这里实现了模糊查找

{

if(findstr(p->bi.book\_name,s)!=-1)

{

show\_book(&(p->bi));//显示查到的图书信息

count++;

}

p=p->next;

}

if(count==0)

{

printf("->没有找到图书 %s .继续查找吗?(y or n):",s);

getchar();

input=getchar();

continue;

}

printf("->共找到 %d 本图书...\n",count);

printf("->继续查找吗?(y or n)");

getchar();

input=getchar();

}

}

void search\_book\_by\_author()//按作者查找图书信息

{

book\* p=NULL;

char s[MAX\_AUTHOR]={0};//作者信息

char input='Y';

int count=0;

int i=0;

printf("->查找图书...\n");

while(input=='Y'||input=='y')

{

count=0;

p=first\_book;//p指向第一个节点

memset(s,0,MAX\_AUTHOR);

printf("->请输入作者(最大长度为 %d):",MAX\_AUTHOR);

scanf("%s",s);

while(p!=NULL)//p不为空时，调用findstr函数查找数名中是否包含字符串s

{

if(findstr(p->bi.author,s)!=-1)

{

show\_book(&(p->bi));

count++;

}

p=p->next;

}

if(count==0)

{

printf("->没有找到作者为 %s 的图书,继续查找吗?(y or n):",s);

getchar();

input=getchar();

continue;

}

printf("->共找到 %d 本图书...\n",count);

printf("->继续查找吗?(y or n):");

getchar();

input=getchar();

}

}

void search\_book\_by\_publisher()//按出版社查找图书信息

{

book\* p=NULL;

char s[MAX\_PUBLISHER]={0};

char input='Y';

int count=0;

int i=0;

printf("->查找图书...\n");

while(input=='Y'||input=='y')

{

count=0;

p=first\_book;

memset(s,0,MAX\_AUTHOR);

printf("->请输入出版社(最大长度为 %d):",MAX\_PUBLISHER);

scanf("%s",s);

while(p!=NULL)

{

if(findstr(p->bi.publisher,s)!=-1)

{

show\_book(&(p->bi));

count++;

}

p=p->next;

}

if(count==0)

{

printf("->没有找到出版社为 %s 的图书.继续查找吗?(y or n):",s);

getchar();

input=getchar();

continue;

}

printf("->共找到 %d 本图书...\n",count);

printf("->继续查找吗?(y or n):");

getchar();

input=getchar();

}

}

void search\_book\_by\_pubdate()//按出版日期查找图书信息

{

book\* p=NULL;

char s[MAX\_DATE]={0};//出版日期

char input='Y';

int count=0;

int i=0;

printf("->查找图书...\n");

while(input=='Y'||input=='y')

{

count=0;

p=first\_book;

memset(s,0,MAX\_DATE);

printf("->请输入出版日期(最大长度为%d):",MAX\_DATE);

scanf("%s",s);

while(p!=NULL)

{

if(findstr(p->bi.pub\_date,s)!=-1)

{

show\_book(&(p->bi));

count++;

}

p=p->next;

}

if(count==0)

{

printf(">没有找到出版日期为 %s 的图书.继续查找吗?(y or n):",s);

getchar();

input=getchar();

continue;

}

printf("->共找到 %d 本图书...\n",count);

printf("->继续查找吗?(y or n):");

getchar();

input=getchar();

}

}

void search\_book\_by\_isbn()//按ISBN查找图书信息

{

char input='Y';

char isbn[MAX\_ISBN]={0};

book\* p=NULL;

book\* result=NULL;

while(input=='Y'||input=='y')

{

printf("->查找图书...\n");

printf("->请输入ISBN(最大长度为 %d):",MAX\_ISBN);

scanf("%s",isbn);

p=first\_book;//p指向第一个节点

result=NULL;

while(p!=NULL)//在图书链表中查找输入的ISBN是否存在

{

if(strcmp(p->bi.ISBN,isbn)==0)

{

result=p;

break;

}

p=p->next;

}

if(result!=NULL)

{

printf("查找到图书...\n");

show\_book(&(result->bi));//调用show\_book 显示查找到的图书信息

}

else

{

printf("->没有找到ISBN为%s的图书.\n",isbn);

}

printf("继续查找吗?(y or n)");

getchar();

input=getchar();

}

}

void delete\_book()//删除图书

{

char input='Y';

char isbn[MAX\_ISBN]={0};

book\* p=NULL;

book\* result=NULL;

while(input=='Y'||input=='y')

{

printf("->删除图书...\n");

printf("请输入ISBN(最大长度为 %d):",MAX\_ISBN);

scanf("%s",isbn);

p=first\_book;

result=NULL;

while(p!=NULL)

{

if(strcmp(p->bi.ISBN,isbn)==0)

{

result=p;

break;

}

p=p->next;

}

if(result!=NULL)

{

show\_book(&(result->bi));

printf("->确认删除吗?(y or n):");

getchar();

input=getchar();

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

{

if(get\_previous\_book(p)==NULL)first\_book=p->next;

else

get\_previous\_book(p)->next=p->next;

free(p);

}

}

else

{

printf("->没有找到ISBN为 %s 的图书.\n",isbn);

}

printf("->继续删除其他图书吗?(y or n):");

getchar();

input=getchar();

}

}

void save\_books()//保存图书

{

save\_books\_to\_file();

printf("->保存图书成功!按任意键返回...");

getchar();

getchar();

}

void save\_books\_to\_file()

{

FILE\* fp=fopen(BOOK\_FILE,"wb");

book\* p=first\_book;

while(p!=NULL)

{

fwrite(&(p->bi),sizeof(book\_info),1,fp);

fseek(fp,0,SEEK\_END);

p=p->next;

}

fclose(fp);

}

void init\_book()

{

FILE\* fp=NULL;

fp=fopen(BOOK\_FILE,"r");

if(fp==NULL)//如果文件不存在

{

fp=fopen(BOOK\_FILE,"w");//创建文件

if(fp==NULL)

{

printf("不能创建文件,按任意键退出...");

getchar();

exit(0);

}

}

fclose(fp);

}

void load\_books()//加载图书信息

{

book\* b=NULL;

book\* last=NULL;

FILE\* fp=NULL;

int count=0;

b=(book\*)malloc(sizeof(book));

memset(b,0,sizeof(book));//初始化b

b->next=NULL;

fp=fopen(BOOK\_FILE,"rb");//打开图书文件

while(fread(&(b->bi),sizeof(book\_info),1,fp)==1)//从文件中逐个读出图书信息

{

if(first\_book==NULL)//如果读取的是第一个节点，即第一本书

{

first\_book=b;//将头节点指向b

}

else

{

last=get\_last\_book();//否则找到链表中最后一个图书节点，即最后一本书

last->next=b;//将b连接到最后一个节点之后

}

count++;

fseek(fp,count\*sizeof(book\_info),SEEK\_SET);//将文件指针指到下一本书

b=(book\*)malloc(sizeof(book));

memset(b,0,sizeof(book));

b->next=NULL;

}

free(b);

b=NULL;

fclose(fp);

}

void clear\_books()//清除图书链表

{

book\* p=NULL;

while(first\_book!=NULL)//链表不为空

{

if(first\_book->next!=NULL)//如果链表中有两本以上的书

{

p=first\_book;

first\_book=first\_book->next;//头结点向后移一位

free(p);//释放原来的头结点

p=NULL;

}

else

{

free(first\_book);

first\_book=NULL;

}

}

}

int findstr(char\* source,char\* str)//查找字符串

{

int pos=-1;

int i=0;

int j=0;

int m=strlen(source);

int n=strlen(str);

if(m==0||n==0||m<n)//str长度为0,或者str长度比source长的情况

{

return(pos);

}

if(m==n)//str和source长度相等的情况

{

if(strcmp(source,str)==0)

{

return(0);

}

else

{

return(-1);

}

}

for(i=0;i<(m-n);i++)

{

pos=i;

for(j=0;j<n;j++)

{

if(source[i+j]!=str[j])

{

pos=-1;

break;

}

}

if(pos!=-1)

{

break;

}

}

return(pos);

}

book\* get\_last\_book()//取得链表中最后一个节点

{

book\* p=first\_book;

if(p==NULL)

{

return(p);

}

while((p!=NULL)&&(p->next!=NULL))

{

p=p->next;

}

return(p);

}

void input\_book(book\_info\* info)//输入图书信息

{

printf("->请输入书名(最大长度为 %d):",MAX\_BOOK\_NAME);

scanf("%s",info->book\_name);

printf("->请输入作者(最大长度为 %d):",MAX\_AUTHOR);

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

printf("->请输入出版社(最大长度为 %d):",MAX\_PUBLISHER);

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

printf("->请输入出版日期(最大长度为 %d):",MAX\_DATE);

scanf("%s",info->pub\_date);

printf("->请输入ISBN(最大长度为 %d):",MAX\_ISBN);

scanf("%s",info->ISBN);

printf("->请输入页数:");

scanf("%d",&(info->pages));

}

void show\_book(book\_info\* info)//显示图书信息

{

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

printf("书名:%s\n",info->book\_name);

printf("作者:%s\n",info->author);

printf("出版社:%s\n",info->publisher);

printf("出版日期:%s\n",info->pub\_date);

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

printf("ISBN:%s\n",info->ISBN);

printf("页数:%d\n",info->pages);

printf("\n");

}

book\* get\_previous\_book(book\* p)//取得图书节点p的前一个节点

{

book\* previous=first\_book;

while(previous!=NULL)

{

if(previous->next==p)

{

break;

}

previous=previous->next;

}

return(previous);

}

#define USER\_FILE "user.dat"

user\* first\_user=NULL;//结构体链表的头结点

void add\_user()//新增用户

{

char try\_again='Y';user\* p=NULL;

user\* new\_user=(user\*)malloc(sizeof(user));//创建一个节点new\_user

while(try\_again=='Y'||try\_again=='y')

{

memset(new\_user,0,sizeof(user));//初始化new\_user

new\_user->next=NULL;

printf("->新增用户信息...\n");

input\_user(&(new\_user->ui));//调用函数inpue\_user()为new\_user赋值

p=find\_user(new\_user->ui.username);

if(p==NULL)

{

p=get\_last\_user();//调用函数，取链表中最后一个节点，赋值给p

p->next=new\_user;

break;

}

printf("->用户[%s]已存在.重新输入吗?(y or n):",new\_user->ui.username);

getchar();

try\_again=getchar();

if(try\_again!='Y'&&try\_again!='y')

{

free(new\_user);

}

}

}

void view\_user()//浏览用户信息

{

user\* p=NULL;

char input='Y';

int count=0;

while(input=='Y'||input=='y')

{

count=0;

p=first\_user;

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

printf("| 账号 | 密码 | 用户类型 |\n");

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

while(p!=NULL)

{

printf("|%20s|%21s|%12s|\n",p->ui.username,p->ui.password,p->ui.user\_type==ADMIN?"管理员":"普通用户");

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

count++;

if(count==5&&p->next!=NULL)

{

count=0;

printf("->显示下一页吗?(y or n):");

getchar();

input=getchar();

if(input!='Y'&&input!='y')

{

break;

}

}

p=p->next;

}

printf("->再次浏览用户吗?(y or n):");

getchar();

input=getchar();

}

}

void search\_user()//查找用户

{

char input\_char='Y';

char username[MAX\_USERNAME]={0};

user\* p=NULL;

while(input\_char=='Y'||input\_char=='y')

{

printf("->查找用户信息...\n");

printf("->请输入用户名(最大长度为%d):",MAX\_USERNAME);

scanf("%s",username);

p=find\_user(username);//调用函数，查找用户是否存在，若存在，赋值给P

if(p==NULL)//该用户不存在

{

printf("->未找到用户:%s 的信息.继续查找吗?(y or n):",username);

getchar();

input\_char=getchar();

continue;

}

show\_user(&(p->ui));//显示该用户信息

printf("->查找成功!按d键删除该用户,按u键更新该用户信息,按其他键返回!");

getchar();

input\_char=getchar();

if(input\_char=='d'||input\_char=='D')

{

delete\_user(p);

}

else if(input\_char=='u'||input\_char=='U')

{

update\_user(p);

}

printf("->要查找其他用户吗?(y or n):");

getchar();

input\_char=getchar();

}

}

void delete\_user(user\* p)//删除用户信息

{

char input\_char='N';

user\* previous=NULL;

printf("->确定要删除用户[%s]吗?(y or n):",p->ui.username);

getchar();

input\_char=getchar();

if(input\_char=='Y'||input\_char=='y')

{

previous=get\_previous\_user(p);

if(previous==NULL)

{

first\_user=p->next;

}

else

{

previous->next=p->next;

}

free(p);

p=NULL;

}

}

void update\_user(user\* p)//更新用户信息

{

char input='Y';

user\* exist\_p=NULL;

user\_info\* new\_p=(user\_info\*)malloc(sizeof(user\_info));//创建一个用户信息

while(input=='Y'||input=='y')

{

memset(new\_p,0,sizeof(user\_info));

input\_user(new\_p);//输入用户信息

exist\_p=find\_user(new\_p->username);//查找输入的用户名是否已经存在

if(exist\_p!=NULL&&exist\_p!=p)//更新后的用户不存在，并不是用户P

{

printf("->用户[%s]已存在,请选择其他用户名.\n",exist\_p->ui.username);

printf("->重新输入吗?(y or n):");

getchar();

input=getchar();

}

else

{

strcpy(p->ui.username,new\_p->username);

strcpy(p->ui.password,new\_p->password);

p->ui.user\_type=new\_p->user\_type;

break;

}

}

free(new\_p);

}

void save\_users()//保存用户

{

save\_users\_to\_file();

printf("->保存成功!按任意键返回...");

getchar();

getchar();

}

void save\_users\_to\_file()//将保存用户信息保存到文件

{

FILE\* fp=fopen(USER\_FILE,"wb");

user\* p=first\_user;

while(p!=NULL)

{

fwrite(&(p->ui),sizeof(user\_info),1,fp);

fseek(fp,0,SEEK\_END);

p=p->next;

}

fclose(fp);

}

void init\_user()//用户模块初始化

{

FILE\* fp=NULL;

user\_info default\_admin;

strcpy(default\_admin.username,"admin");

strcpy(default\_admin.password,"123");

default\_admin.user\_type=ADMIN;

fp=fopen(USER\_FILE,"r");

if(fp==NULL)//文件不存在

{

fp=fopen(USER\_FILE,"wb");

fwrite(&default\_admin,sizeof(user\_info),1,fp);

}

fclose(fp);

}

void load\_users()//加载图书信息

{

user\* u=NULL;

user\* last=NULL;

FILE\* fp=NULL;

int count=0;

u=(user\*)malloc(sizeof(user));

memset(u,0,sizeof(user));

u->next=NULL;

fp=fopen(USER\_FILE,"rb");

fseek(fp,0,SEEK\_SET);

while(fread(&(u->ui),sizeof(user\_info),1,fp)==1)

{

if(first\_user==NULL)

{

first\_user=u;

}

else

{

last=get\_last\_user();

last->next=u;

}

count++;

fseek(fp,count\* sizeof(user\_info),SEEK\_SET);

u=(user\*)malloc(sizeof(user));

memset(u,0,sizeof(user));

u->next=NULL;

}

free(u);

u=NULL;

fclose(fp);

}

USER\_TYPE login()//判断用户类型

{

char username[MAX\_USERNAME]={0};

char password[MAX\_PASSWORD]={0};

char try\_again='Y';

user\* p=NULL;

while(try\_again=='Y'||try\_again=='y')

{

printf("->请输入用户名:");

scanf("%s",username);

printf("->请输入密码:");

scanf("%s",password);

p=find\_user(username);//查找用户名是否存在

if(p==NULL)

{

printf("用户名输入错误,请重新输入!");

}

else if(strcmp(p->ui.password,password)!=0)//比较密码是否正确

{

printf("密码输入错误,请重试!");

}

else

{

return(p->ui.user\_type);

}

printf("->重新输入吗?(y or n):");

getchar();

try\_again=getchar();

}

exit(0);

}

void clear\_users()//清空用户链表

{

user\* p=NULL;

while(first\_user!=NULL)

{

if(first\_user->next!=NULL)

{

p=first\_user;

first\_user=first\_user->next;

free(p);

p=NULL;

}

else

{

free(first\_user);

first\_user=NULL;

}

}

}

user\* get\_last\_user()//取用户链表中最后一个节点

{

user\* p=first\_user;

while((p!=NULL)&&(p->next!=NULL))

{

p=p->next;

}

return(p);

}

user\* get\_previous\_user(user\* p)//取某节点的前一个节点

{

user\* previous=first\_user;

while(previous!=NULL)

{

if(previous->next==p)

{

break;

}

previous=previous->next;

}

return(previous);

}

void show\_user(user\_info\* info)//显示一个用户信息

{

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

printf("用户名:%s\n",info->username);

printf("密码:%s\n",info->password);

printf("用户类型:%s\n",info->user\_type==ADMIN?"admin":"user");

printf("\n");

}

void input\_user(user\_info\* info)//输入用户信息

{

printf("->请输入用户名(最大长度为 %d):",MAX\_USERNAME);

scanf("%s",info->username);

printf("->请输入密码(最大长度为 %d:)",MAX\_PASSWORD);

scanf("%s",info->password);

printf("->请输入用户类型(%d 是管理员,%d 是普通用户)",ADMIN,USER);

scanf("%d",&(info->user\_type));

}

user\* find\_user(char\* name)//查找一个用户

{

user\* p=first\_user;

int is\_found=0;

while(p!=NULL)

{

if(strcmp(p->ui.username,name)==0)

{

is\_found=1;

break;

}

p=p->next;

}

if(is\_found)

{

return(p);

}

return(NULL);

} //第一个文件开始

#define MENU\_ADMIN\_COUNT 10//管理员操作主菜单的选项个数

#define MENU\_SEARCH\_BOOK\_COUNT 6//管理员查询菜单的选项个数

#define MENU\_USER\_COUNT 7//普通用户操作主菜单的选择个数

char menu\_title[]=

" ╱╮╮╮╮ 图书管理系统 ╭╭╭╭╲ \n"

"│ ╮╮╮ ╭╭╭ │\n";

char menu\_admin[]=

"│ ╮╮ ╭╭ │\n"

"│ 图书管理 ╮ ╭ 用户管理 │\n"

"│ ╮ ╭ │\n"

"│ <1>新增图书 ▏ <6>新增用户 │\n"

"│ <2>浏览图书 ▏ <7>浏览用户 │\n"

"│ <3>查找图书 ▏ <8>查找用户 │\n"

"│ <4>删除图书 ▏ <9>保存用户 │\n"

"│ <5>保存图书 ▏ │\n"

"│ ▏ <10>退出系统 │\n"

"│╱╮╮╮╮╮ ▏ ╭╭╭╭╭╲│\n"

" ╮╮╮╮╮ ▏ ╭╭╭╭╭ \n"

" ╮╮╮ ▏ ╭╭╭ \n"

" ─ ─ \n";

void (\*admin\_func[])()=//函数指针数组储存的是管理员权限操作主菜单所对应的函数

{

add\_book,

view\_book,

show\_search\_book\_menu,

delete\_book,

save\_books,

add\_user,

view\_user,

search\_user,

save\_users,

admin\_exit

};

char menu\_admin\_search\_book[]=

"│ ╮╮ ╭╭ │\n"

"│ 查找图书 ╮ ╭ │\n"

"│ ╮ ╭ │\n"

"│ <1>按书名查找 <4>按出版日期查找 │\n"

"│ ▏ │\n"

"│ <2>按作者查找 ▏ <5>按书号(ISBM)查找 │\n"

"│ ▏ │\n"

"│ <3>按出版社查找 ▏ <6>返回主菜单 │\n"

"│ ▏ │\n"

"│╱╮╮╮╮╮ ▏ ╭╭╭╭╭╲│\n"

" ╮╮╮╮╮ ▏ ╭╭╭╭╭ \n"

" ╮╮╮ ▏ ╭╭╭ \n"

" ─ ─ \n";

void (\*admin\_search\_book\_func[])()=//函数指针数组储存的是管理员权限登陆后，查询图书的函数

{

search\_book\_by\_name,

search\_book\_by\_author,

search\_book\_by\_publisher,

search\_book\_by\_pubdate,

search\_book\_by\_isbn,

};

char menu\_user[]=

"│ ╮╮ ╭╭ │\n"

"│ ╮ ╭ │\n"

"│ ╮ ╭ │\n"

"│ <1>浏览图书 <5>按出版日期查找图书│\n"

"│ <2>按书名查找图书 ▏ <6>按书号查找图书 │\n"

"│ <3>按作者查找图书 ▏ │\n"

"│ <4>按出版社查找图书 ▏ <7>退出系统 │\n"

"│ ▏ │\n"

"│ ▏ │\n"

"│╱╮╮╮╮╮ ▏ ╭╭╭╭╭╲│\n"

" ╮╮╮╮╮ ▏ ╭╭╭╭╭ \n"

" ╮╮╮ ▏ ╭╭╭ \n"

" ─ ─ \n";

void (\*user\_func[])()=//普通用户权限登录主菜单所对应的函数

{

view\_book,

search\_book\_by\_name,

search\_book\_by\_author,

search\_book\_by\_publisher,

search\_book\_by\_pubdate,

search\_book\_by\_isbn,

user\_exit

};

void dongtai1()//动态欢迎界面

{

char \*wenzi1="欢迎使用 图书管理系统2.0";

char \*wenzi2="制作:'hzj'研发小组";

char \*hua[13],\*p,\*zifuhua[12];

hua[0]=" ┏┓ b \n";

hua[1]=" ┃┃ b \n";

hua[2]=" ┏━┛┗━┓┏┓b \n";

hua[3]=" ┗━┓┏┓┃┗┛b \n";

hua[4]=" ┃┃┃┃ b┏┓ ┏┓ \n";

hua[5]="┏━━┛┗┛┗━┓b┃┃ ┃┃ ┏┓ \n";

hua[6]="┗━━┓┏━━┓┃b┃┗━┳━━┳━━┫┃ ┛┛ \n";

hua[7]=" ┃┃ ┏ ┛b┃┏┓┃┏┓┃┏┓┃┣┛┛ \n";

hua[8]=" ┃┃ ┏┏ b┃┃┃┃┃┃┃┃┃┃┣┓┓ \n";

hua[9]=" ┃┃ ┏┏ b┃┗┛┃┗┛┃┗┛┃┃┗┓┓\n";

hua[10]=" ┗┛ ┗┛ b┗━━┻━━┻━━┻┛ ┗┛\n";

zifuhua[0]=" ┏┓ \n";

zifuhua[1]=" ┃┃ \n";

zifuhua[2]=" ┏━┛┗━┓┏┓ \n";

zifuhua[3]=" ┗━┓┏┓┃┗┛ \n";

zifuhua[4]=" ┃┃┃┃ ┏┓ ┏┓ \n";

zifuhua[5]="┏━━┛┗┛┗━┫┃ ┃┃ ┏┓ \n";

zifuhua[6]="┗━━┓┏━━┓┃┗━┳━━┳━━┫┃ ┛┛ \n";

zifuhua[7]=" ┃┃ ┏ ┫┏┓┃┏┓┃┏┓┃┣┛┛ \n";

zifuhua[8]=" ┃┃ ┏┏┃┃┃┃┃┃┃┃┃┃┣┓┓ \n";

zifuhua[9]=" ┃┃ ┏┏ ┃┗┛┃┗┛┃┗┛┃┃┗┓┓\n";

zifuhua[10]=" ┗┛ ┗┛ ┗━━┻━━┻━━┻┛ ┗┛\n";

zifuhua[11]=" 请登录系统... \n";

int i,k,n;

for(n=10;n>=0;n--)

{

for(i=0;i<11;i++)

{

p=hua[i];

while(\*p!='\0')

{

if(\*p=='b')

{

for(k=0;k<n;k++)

{

printf(" ");

}

}

else

{

printf("%c",\*p);

}

p++;

}

}

Sleep(100);

system("cls");

}

for(i=0;i<11;i++)

{

printf(zifuhua[i]);

}

p=wenzi1;

printf("\t");

while(\*p!='\0')

{

printf("%c",\*p);

Sleep(100);

p++;

}

p=wenzi2;

printf("\n\t");

while(\*p!='\0')

{

printf("%c",\*p);

Sleep(100);

p++;

}

system("cls");

for(i=0;i<12;i++)

{

printf(zifuhua[i]);

}

}

void show\_admin\_menu()//显示管理员操作的菜单

{

int selected=0;

while(selected<1||selected>MENU\_ADMIN\_COUNT)

{

system("cls");

printf(menu\_title);

printf(menu\_admin);

printf("->请选择要进行的操作:");

scanf("%d",&selected);

if(selected<1||selected>MENU\_ADMIN\_COUNT)

{

printf("->输入出错啦!请选择[%d-%d]之间的数字!按任意键重试试吧...",1,MENU\_ADMIN\_COUNT);

getchar();

getchar();

}

else

{

admin\_func[selected-1]();

}

selected=0;

}

}

void show\_search\_book\_menu()//显示管理员查询图书的菜单

{

int selected=0;

while(selected<1||selected>MENU\_SEARCH\_BOOK\_COUNT)

{

system("cls");

printf(menu\_title);

printf(menu\_admin\_search\_book);

printf("->请选择要进行的操作:");

scanf("%d",&selected);

if(selected==MENU\_SEARCH\_BOOK\_COUNT)

{

break;

}

if(selected<1||selected>MENU\_SEARCH\_BOOK\_COUNT)

{

printf("->输入出错啦!请选择[%d-%d]之间的数字!按任意键重试试吧...",1,MENU\_ADMIN\_COUNT);

getchar();

getchar();

}

else

{

admin\_search\_book\_func[selected-1]();

}

selected=0;

}

}

void show\_user\_menu()//显示普通用户操作的菜单

{

int selected=0;

while(selected<1||selected>MENU\_USER\_COUNT)

{

system("cls");

printf(menu\_title);

printf(menu\_user);

printf("->请选择要进行的操作:");

scanf("%d",&selected);

if(selected<1||selected>MENU\_USER\_COUNT)

{

printf("->输入出错啦!请选择[%d-%d]之间的数字!按任意键重试试吧...",1,MENU\_USER\_COUNT);

getchar();

getchar();

}

else

{

user\_func[selected-1]();

}

selected=0;

}

}

void admin\_exit()//管理员退出

{

char sure='N';

printf(">确定要退出吗?(y or n)");

getchar();

sure=getchar();

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

{

save\_users\_to\_file();//将用户链表中的数据保存到文件

clear\_users();//清空用户链表

save\_books\_to\_file();//将图书链表中的数据保存到文件

clear\_books();//清空图书链表

exit(0);

}

}

void user\_exit()//普通用户退出

{

char sure='N';

printf(">确定要退出吗?(y or n)");

getchar();

sure=getchar();

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

{

clear\_users();//清空用户链表

clear\_books();//清空图书链表

exit(0);

}

}

int main()

{

char input='N';

init\_user();

load\_users();

init\_book();

load\_books();

dongtai1();

if(login()==ADMIN)

{

show\_admin\_menu();

}

else

{

show\_user\_menu();

}

return(0);

}