# 班级：智科232 姓名：刘糠杰

# 学号：202305130222

# 项目介绍

该程序是一个图书管理系统，主要用于学校、图书馆或者一些书店进行书籍管理和会员管理。会员可以查询书籍信息、借书、还书并查看个人账户信息，管理员可以添加、删除书籍，修改书籍信息，管理所有书籍信息和会员信息。这种系统可以提高书籍管理的效率，减少人工操作中的错误，方便用户和管理员进行相关操作。

# 设计思想

## 设计的数据结构

使用结构体(struct)来存储书籍信息、借书信息和会员信息，以及用这些结构体数组实现对书籍信息，借书信息和会员信息的增删改查。

## 文件操作

通过文件操作(fopen, fprintf, fscanf等)将数据持久化，存储在本地文件中。

## 用户交互

通过控制台输入输出函数printf和scanf实现与用户的交互。

## 流程控制

使用while循环和条件判断(if, switch)实现系统的流程控制。

## 用户角色区分

系统区分管理员和普通会员，提供不同的功能菜单，管理员具有更多的权限。

# 图书管理系统主体框架

# 主要的功能模块

## 主菜单

会员登录：会员可以通过输入账号和密码进行登录。

会员注册：新用户可以注册成为会员。

管理员登录：通过输入管理员密码登录，进行图书管理和查看所有会员信息。

## 会员功能菜单

查找图书信息：通过图书的编号、书名或作者查询。

借阅图书：会员可以借阅书籍，系统会记录借阅信息。

归还图书：会员可以归还之前借阅的书籍。

查看所有图书信息：显示书库中所有书籍的详细信息。

查看账户信息：显示用户的个人信息和借书信息，以及修改密码。

## 管理员功能菜单

添加图书：管理员可以添加新的图书信息。

删减图书：根据书名删除图书。

修改图书信息：修改书库中已有图书的信息。

查找图书信息：和会员功能中的查找图书信息一致。

显示全部图书信息：显示所有图书的详细信息。

查看会员信息：查看所有会员的个人信息。

查看所有图书借阅信息：查看所有会员的借书信息。

# 设计难点和解决方案

## 数据存储与文件操作：

难点：需要通过文件读写操作，保持数据的一致性和持久化。

解决方案：使用文件指针（如FILE \*p1）和文件操作函数（如fscanf, fprintf）将书籍和会员信息实时读写到文件中。

系统数据使用文件进行持久化存储library.txt存储图书信息，infomember.txt存储会员信息，memberbook.txt存储借阅信息。方便系统重启后数据的保存和加载。

## 用户输入的有效性检查：

难点：输入的有效性和错误处理，如重复的ID、非法字符输入等。

解决方案：在用户输入后通过循环和条件判断进行检查，例如输入是否为有效的Y或N，设计函数check\_repeat\_id(char id[30]);来检查 ID是否重复。

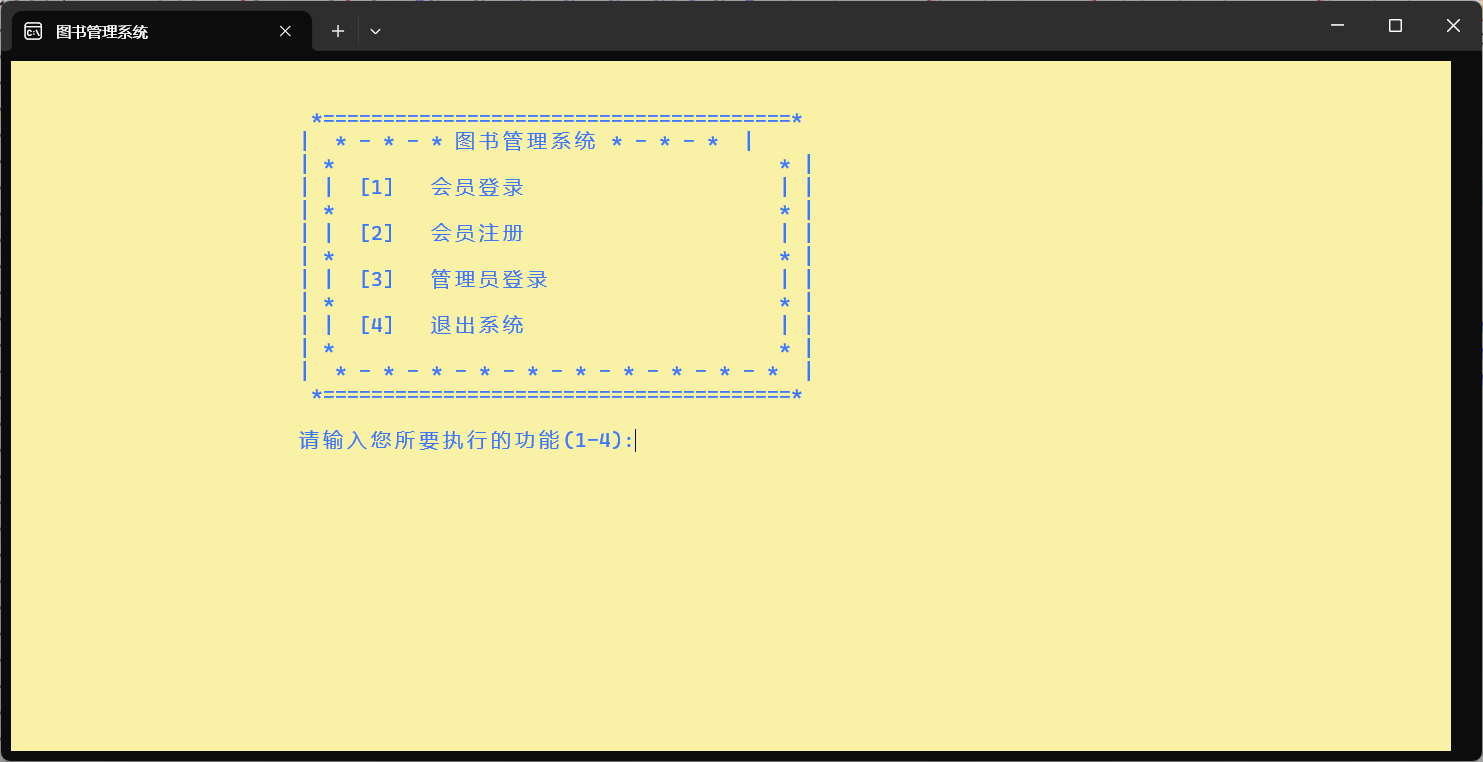
## 用户体验：

难点：提升用户与系统交互的友好度，如输入提示、错误消息和确认操作等。

解决方案：在用户输入前提供明确提示，通过printf输出详细的信息。在关键操作如删除、修改前添加二次确认的机制。

# 运行结果图及源代码

主界面1



会员登录界面

图形用户界面, 文本

描述已自动生成

图形用户界面, 文本

描述已自动生成

会员注册界面

图片包含 形状

描述已自动生成

主界面2（会员界面）

图形用户界面

低可信度描述已自动生成

图形用户界面

低可信度描述已自动生成

[1]查找图书信息

图形用户界面, 应用程序

描述已自动生成

[2]借阅图书

形状, 矩形

描述已自动生成

[3]归还图书

图形用户界面, 应用程序

描述已自动生成

[4]书库所有图书信息

图形用户界面, 文本

描述已自动生成

[5]您的账户信息

图形用户界面, 应用程序, 表格

描述已自动生成

管理员登录界面

图片包含 表格

描述已自动生成

主界面3（管理员界面）

文本

描述已自动生成

[1]添加图书

图片包含 图形用户界面

描述已自动生成

[2]删减图书

形状

中度可信度描述已自动生成

[3]修改图书信息

图形用户界面, 应用程序

描述已自动生成

[4]查找图书信息

图形用户界面, 应用程序

描述已自动生成

[5]显示全部图书信息

图形用户界面, 文本

描述已自动生成

[6]显示所有会员信息

图形用户界面, 应用程序

描述已自动生成

[7]显示所有借阅信息

图形用户界面, 应用程序

描述已自动生成

源代码

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#include <time.h>

/\*宏定义区\*/

#define  ADMIN\_password "123456" /\*管理员登录密码\*/

//书籍信息

typedef struct book

{

    char id[50];       //编号

    char name[50];    //书名

    char author[50];   //作者

    char press[50];    //出版社

    char category[50]; //分类

    double price;       //价格

} BOOKINFO;

int library\_quantity = 0;

//会员借出书籍信息

typedef struct book\_loans

{

    char id[50];       //编号

    char name[50];    //书名

    char author[50];   //作者

    char press[50];    //出版社

    char category[50]; //分类

    double price;       //价格

    char member\_id[50];//会员账号

} BOOKLOANS;

int book\_loans = 0;//总的书籍借阅量

//会员信息

typedef struct member

{

    char id[50];       //账户

    char password[50]; //密码

    char name[50];     //姓名

    char sex[10];      //性别

    char phone[12];    //手机号

} MEMBERINFO;

int member\_number = 0;

/\*菜单函数区\*/

void menu1();

void menu2();

void menu3();

/\*工具函数区\*/

int compare\_password(char password[20]);/\*密码比对函数 \*/

void clock\_delay(); /\*时间延时函数\*/

void password\_circle();/\*密码框装饰函数\*/

int is\_library\_empty();/\*判断书库中是否有图书信息\*/

/\*会员信息系统层级函数区\*/

int user\_account(char account[20]);/\*个人账户信息\*/

void member\_login\_up();/\*会员注册函数\*/

int member\_login\_in(char account[20]);/\*会员登录函数\*/

void ADMIN\_watch\_member();/\*查看会员信息函数\*/

void display\_memberbook();/\*查看所有图书借阅信息函数\*/

int check\_repeat\_id(char id[30]);/\*检查账户重复性函数\*/

/\*图书信息系统层级函数区\*/

void add\_book();/\*输入新进的图书信息\*/

void delete\_book();/\*根据图书名称对图书信息进行删除\*/

void modify\_book();/\*修改图书信息\*/

void display\_book();/\*显示全部图书信息\*/

void search\_book();/\*根据图书名称显示图书的信息\*/

void borrow\_book(char account[20]);/\*借书\*/

void return\_book(char account[20]);/\*还书\*/

int check\_repeat\_book(char number[30]);/\*查重书籍编号\*/

int main()

{

    int flag1=1,flag2=1,flag3=1;/\* flag是判断条件,flag为 1 时为真，为 0 时为假 \*/

    char choice;

    FILE \*p1,\*p2,\*p3;

    system("title 图书管理系统");

    system("color E9");

    if((p1 = fopen("library.txt","r")) == NULL) {

        p1 = fopen("library.txt","w"); /\*p1,书籍信息文件指针\*/

        fclose(p1);

    }

    if((p2 = fopen(".txt","r")) == NULL) {

        p2 = fopen("infomember.txt","w"); /\*p2,会员个人信息文件指针\*/

        fclose(p2);

    }

    if((p3 = fopen("memberbook.txt","r")) == NULL) {

        p3 = fopen("memberbook.txt","w"); /\*p3,会员借阅书籍信息文件指针\*/

        fclose(p3);

    }

    while (1)

    {

        system("cls");

        menu1();/\*调出主菜单 \*/

        printf("\n");

        printf("\t\t\t请输入您所要执行的功能(1-4):");

        choice=getchar();

        while(choice!='1'&&choice!='2'&&choice!='3'&&choice!='4'&&choice!='5') {

            choice=getchar();

        }

        fflush(stdin);

        if(choice == '1') {

            /\*会员功能\*/

            int flag=0;

            char account[20],account\_choice;

            system("cls");

            password\_circle();

            gets(account);

            flag = member\_login\_in(account); /\*登录\*/

            /\*没注册的会调用注册函数\*/

            if(flag == 2) { //没注册，注册完了重新登录

                system("cls");

                password\_circle();

                fflush(stdin);

                gets(account);

                flag=member\_login\_in(account);

            }

            if(flag == 1) {

                while (flag2 && flag3)

                {

                    system("cls");

                    menu2();

                    printf("\t\t\t请您选择(1-6):");

                    account\_choice=getchar();

                    while(account\_choice!='1'&&account\_choice!='2'&&account\_choice!='3'&&account\_choice!='4'&&account\_choice!='5'&&account\_choice!='6'&&account\_choice!='7') {

                        account\_choice=getchar();

                    }

                    switch (account\_choice)

                    {

                        case '1':search\_book(); break;

                        case '2':borrow\_book(account); break;

                        case '3':return\_book(account); break;

                        case '4':display\_book(); break;

                        case '5':if(user\_account(account))

                                    {

                                        printf("\n\n登录信息改变,请重新登录...");

                                        clock\_delay();

                                        flag3=0;

                                    }

                                break;

                        case '6':

                                printf("\n\n\t\t\t正在退出用户界面...");

                                clock\_delay();

                                flag2 = 0;

                                break;

                        default:

                            break;

                    }

                }

                flag2 = flag3 = 1;

            }

        } else if(choice == '2') {

            /\*会员注册\*/

            system("cls");

            printf("\n\n\n\t\t\t正在进入用户注册界面...");

            clock\_delay();

            member\_login\_up();

        } else if(choice == '3') {

            /\*管理员登录\*/

            if(compare\_password(ADMIN\_password)) {

                printf("\n\n\t\t\t\t  --输入密码正确!--\n\n\t\t\t\t==正在进入管理员界面==\n");

                clock\_delay();

                char manager\_choice;

                while(flag2) {

                    system("cls");

                    menu3();

                    printf("\n");

                    printf("\t\t\t请您选择(1-8):");

                    manager\_choice=getchar();

                    while(manager\_choice!='1'&&manager\_choice!='2'&&manager\_choice!='3'&&manager\_choice!='4'&&manager\_choice!='5'&&manager\_choice!='6'&&manager\_choice!='7'&&manager\_choice!='8')

                        manager\_choice=getchar();

                    switch(manager\_choice)

                    {

                        case '1':add\_book(); break;

                        case '2':delete\_book(); break;

                        case '3':modify\_book(); break;

                        case '4':search\_book(); break;

                        case '5':display\_book(); break;

                        case '6':ADMIN\_watch\_member();break;

                        case '7':display\_memberbook();break;

                        case '8':flag2=0;

                                printf("\n\n\t\t\t正在退出管理员界面...");

                                clock\_delay();

                                break;

                        default:

                            break;

                    }

                }

            } else {

                printf("\n\n\t\t\t\t  --输入密码错误!--\n");

                clock\_delay();

            }

        } else {

            /\*退出系统\*/

            break;

        }

    }

    fflush(stdin);

    system("cls");

    printf("您已安全地退出系统!\n(按任意键关闭界面)");

    printf("欢迎您的下次使用!\n");

    getchar();

    return 0;

}

/\*菜单函数区\*/

void menu1()

{

    system("color E9");

    printf("\n\n");

    printf("\t\t\t \*=======================================\*\n");

    printf("\t\t\t|  \* - \* - \* 图书管理系统 \* - \* - \*  |\n");

    printf("\t\t\t| \*                                     \* |\n");

    printf("\t\t\t| |  [1]   会员登录                     | |\n");

    printf("\t\t\t| \*                                     \* |\n");

    printf("\t\t\t| |  [2]   会员注册                     | |\n");

    printf("\t\t\t| \*                                     \* |\n");

    printf("\t\t\t| |  [3]   管理员登录                   | |\n");

    printf("\t\t\t| \*                                     \* |\n");

    printf("\t\t\t| |  [4]   退出系统                     | |\n");

    printf("\t\t\t| \*                                     \* |\n");

    printf("\t\t\t|  \* - \* - \* - \* - \* - \* - \* - \* - \* - \*  |\n");

    printf("\t\t\t \*=======================================\*\n");

}

void menu2()

{

    system("color F9");

    printf("\n\n");

    printf("\t\t\t \*=====================================\*\n");

    printf("\t\t\t| | \* - \* - \* -会-员-界-面- \* - \* - \* | |\n");

    printf("\t\t\t| \*                                   \* |\n");

    printf("\t\t\t| |  [1]   查找图书信息               | |\n");

    printf("\t\t\t| \*  [2]   借阅图书                   \* |\n");

    printf("\t\t\t| |  [3]   归还图书                   | |\n");

    printf("\t\t\t| \*  [4]   书库所有图书信息           \* |\n");

    printf("\t\t\t| |  [5]   您的账户信息               | |\n");

    printf("\t\t\t| \*  [6]   退出用户界面               \* |\n");

    printf("\t\t\t| |                                   | |\n");

    printf("\t\t\t| \* - \* - \* - \* - \* - \* - \* - \* - \* - \* |\n");

    printf("\t\t\t \*=====================================\*\n");

}

void menu3()

{

    system("color F9");

    printf("\n\n");

    printf("\t\t\t \*======================================\*\n");

    printf("\t\t\t| | \* - \* - \* 管-理-员-界-面 \* - \* - \* | |\n");

    printf("\t\t\t| \*                                    \* |\n");

    printf("\t\t\t| |  [1]   添加图书                    | |\n");

    printf("\t\t\t| \*  [2]   删减图书                    \* |\n");

    printf("\t\t\t| |  [3]   修改图书信息                | |\n");

    printf("\t\t\t| \*  [4]   查找图书信息                \* |\n");

    printf("\t\t\t| |  [5]   显示全部图书信息            | |\n");

    printf("\t\t\t| \*  [6]   显示所有会员信息            \* |\n");

    printf("\t\t\t| |  [7]   显示所有借阅信息            | |\n");

    printf("\t\t\t| \*  [8]   退出管理员界面              \* |\n");

    printf("\t\t\t| |                                    | |\n");

    printf("\t\t\t| \* - \* - \* - \* - \* -- \* - \* - \* - \* - \* |\n");

    printf("\t\t\t \*======================================\*\n");

}

/\*工具函数区\*/

/\*时间延时函数\*/

#include <time.h>

void clock\_delay()

{

    clock\_t start\_time = clock();

    // 1秒 = CLOCKS\_PER\_SEC个时钟周期

    // 1000毫秒 = CLOCKS\_PER\_SEC / 1000个时钟周期

    while ((clock() - start\_time) < (CLOCKS\_PER\_SEC / 1000) \* 1000);

}

/\*密码框装饰函数\*/

void password\_circle()

{

    system("cls");

    printf("\n\n\n\t\t\t\*\*\*================================\*\*\* \n");

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

    printf("\t\t\t   账户 (account):");

}

/\*判断书库中没有任何书籍信息\*/

int is\_library\_empty(int library\_quantity)

{

    if(!library\_quantity) {

        printf("书库中没有任何书籍信息！\n");

        printf("按任意键返回主菜单。");

        getchar();

        return 1;

    }

    return 0;

}

/\*图书信息系统层级函数区\*/

//查重书籍编号

int check\_repeat\_book(char id[30])

{

    BOOKINFO books[200];

    FILE \*p1;

    int i;

    library\_quantity = 0;

    p1 = fopen("library.txt","r");

    while(fscanf(p1, "%s %s %s %s %s %lf",books[library\_quantity].id, books[library\_quantity].name, books[library\_quantity].author,

                  books[library\_quantity].press, books[library\_quantity].category,&books[library\_quantity].price) == 6)

    {

        library\_quantity++;

    }

    fclose(p1);

    for(i = 0;i < library\_quantity;i++) {

        if(strcmp(books[i].id,id) == 0) {

            return 1;

        }

    }

    return 0;

}

void add\_book()

{

    BOOKINFO newbook;

    FILE\* p1;

    char choice1,choice2;//choice1,是否保存添加的书籍,choice2,是否继续添加图书信息

    system("cls");

    /\*该书籍已经被添加到书库中\*/

    /\*成功保存书籍信息\*/

    /\*没保存书籍信息\*/

    while(1) {

        fflush(stdin);

        printf("请输入需要添加的书籍编号：");

        gets(newbook.id);

        if(check\_repeat\_book(newbook.id))

        {

            printf("该书籍已经被添加到书库中。\n");

            getchar();

        } else {

            fflush(stdin);

            printf("请输入书名:");

            gets(newbook.name);

            printf("请输入作者:");

            gets(newbook.author);

            printf("请输入出版社:");

            gets(newbook.press);

            printf("请输入类别(文学/科学/理学/工学/等):");

            gets(newbook.category);

            printf("请输入价格:");

            scanf("%lf",&newbook.price);

            fflush(stdin);

            printf("是否保存该书籍信息(Y/N):");

            choice1 = getchar();

            while(choice1 != 'Y' && choice1 != 'N') {

                choice1 = getchar();

            }

            if(choice1 == 'Y') {

                p1 = fopen("library.txt","a");

                fprintf(p1,"%s %s %s %s %s %lf\n",

                    newbook.id,newbook.name,newbook.author,

                    newbook.press,newbook.category,newbook.price);

                fclose(p1);

                printf("已经成功保存新的书籍信息。\n");

            } else {

                printf("未保存该书籍信息。\n");

            }

        }

        printf("是否继续添加图书信息?(Y/N):");

        choice2 = getchar();

        while(choice2 != 'Y'&& choice2 != 'N') {

            choice2 = getchar();

        }

        if(choice2 == 'N') {

            break;

        }

    }

}

void delete\_book()

{

    char delete\_book[20] = ""; //用来存放要删除的书名

    BOOKINFO books[200];

    FILE \*p1;

    char choice1,choice2;

    int i;

    system("cls");

    while (1)

    {

        fflush(stdin);

        printf("请输入要删除的书名:");

        gets(delete\_book);

        library\_quantity = 0;

        p1 = fopen("library.txt","r");

        while(fscanf(p1, "%s %s %s %s %s %lf",books[library\_quantity].id, books[library\_quantity].name, books[library\_quantity].author,

                    books[library\_quantity].press, books[library\_quantity].category,&books[library\_quantity].price) == 6)

        {

            library\_quantity++;

        }

        fclose(p1);

        is\_library\_empty(library\_quantity);

        for(i = 0;i < library\_quantity;i++) {

            if(strcmp(books[i].name,delete\_book) == 0) {

                break;

            }

        }

        if(i >= library\_quantity) {

            printf("书库中没有《%s》的书籍信息。\n",delete\_book);

        } else {

            printf("是否删除《%s》的书籍信息?(Y/N):",delete\_book);

            choice1 = getchar();

            while (choice1 != 'Y' && choice1 != 'N') {

                choice1 = getchar();

            }

            if(choice1 == 'Y') {

                for(;i < library\_quantity;i++) { //越界怎么处理?

                    books[i] = books[i+1];

                }

                p1=fopen("library.txt","w");

                for(i = 0;i < library\_quantity;i++) {

                    fprintf(p1,"%s %s %s %s %s %lf\n",

                        books[i].id, books[i].name, books[i].author,

                        books[i].press, books[i].category, books[i].price);

                }

                fclose(p1);

                library\_quantity--;

                printf("已成功删除《%s》的书籍信息。\n",delete\_book);

            } else {

                printf("《%s》的书籍信息没有被删除。\n",delete\_book);

            }

        }

        printf("是否继续进行删除操作?(Y/N):"); //用户没执行删除操作，没有找到要删除的书籍

        choice2 = getchar();

        while (choice2 != 'Y' && choice2 != 'N') {

            choice2 = getchar();

        }

        if(choice2 == 'N') {

            break;

        }

    }

}

/\*修改书籍信息\*/

void modify\_book()

{

    char id[50];       //编号

    char name[50];     //书名

    char author[50];   //作者

    char press[50];    //出版社

    char category[50]; //分类

    double price;      //价格

    char search\_name[30]; //要修改的书名

    char change; //要修改的书籍信息编号

    FILE \*p1;

    BOOKINFO books[200];

    char choice1, choice2;

    int i;

    int flag1; // flag1 = 1,没找到要修改的图书信息和正常修改完一本图书信息

    int flag2; // flag2 = 1,要修改的图书信息与馆内信息重复

    system("cls");

    while (1)

    {

        flag1 = 0;

        flag2 = 0;

        fflush(stdin);

        library\_quantity = 0;

        p1 = fopen("library.txt","r");

        while(fscanf(p1, "%s %s %s %s %s %lf",books[library\_quantity].id, books[library\_quantity].name, books[library\_quantity].author,

                    books[library\_quantity].press, books[library\_quantity].category,&books[library\_quantity].price) == 6)

        {

            library\_quantity++;

        }

        fclose(p1);

        is\_library\_empty(library\_quantity);

        fflush(stdin);

        printf("请输入要修改信息的书籍名: ");

        gets(search\_name);

        for (i = 0; i < library\_quantity; i++) {

            if (strcmp(books[i].name, search\_name) == 0) {

                break;

            }

        }

        if (i >= library\_quantity) {

            printf("未找到《%s》的书籍信息。\n", search\_name);

            flag1 = 1;

        } else {

            printf("\n");

            printf("|---------------------------------图书信息----------------|\n");

            printf("|编号  书名        作者      出版社          类别    单价  |\n");

            printf("|                                                                          |\n");

            printf("|%-6s%-12s%-10s%-16s%-8s%-8.1f|\n",

                books[i].id, books[i].name, books[i].author,

                books[i].press, books[i].category, books[i].price);

            while (1)

            {

                printf("您想修改的书籍信息是?\n");

                printf("1.编号\n2.书名\n3.作者\n4.出版社\n5.类别\n6.单价\n");

                printf("输入选项编号: ");

                change = getchar();

                while (change < '1' || change > '6')

                {

                    change = getchar();

                }

                fflush(stdin);

                printf("请输入新的词条: ");

                switch (change)

                {

                case '1':

                    gets(id);

                    flag2 = check\_repeat\_book(id);

                    if (!flag2) {

                        strcpy(books[i].id, id);

                        flag1 = 1;

                    }

                    break;

                case '2':

                    gets(name);

                    strcpy(books[i].name, name);

                    flag1 = 1;

                    break;

                case '3':

                    gets(author);

                    strcpy(books[i].author, author);

                    flag1 = 1;

                    break;

                case '4':

                    gets(press);

                    strcpy(books[i].press, press);

                    flag1 = 1;

                    break;

                case '5':

                    gets(category);

                    strcpy(books[i].category, category);

                    flag1 = 1;

                    break;

                case '6':

                    scanf("%lf", &price);

                    books[i].price = price;

                    flag1 = 1;

                    fflush(stdin);

                    break;

                }

                if (flag2) {

                    printf("您要修改的图书信息与馆内信息重复!\n");

                    break;

                } else {

                    printf("是否继续修改《%s》图书信息?(Y/N):", search\_name);

                    fflush(stdin);

                    choice2 = getchar();

                    while (choice2 != 'Y' && choice2 != 'N')

                    {

                        choice2 = getchar();

                    }

                    if (choice2 == 'N') {

                        break;

                    }

                }

            }

        }

        //重新输入要修改的图书信息：1,没找到图书,2,正常修改完一本图书信息,3,要修改的图书信息与管内信息重复

        if (flag1 || flag2) {

            printf("是否继续修改图书信息?(Y/N):");

            fflush(stdin);

            choice1 = getchar();

            while (choice1 != 'Y' && choice1 != 'N')

            {

                choice1 = getchar();

            }

            if (choice1 == 'N') {

                p1 = fopen("library.txt","w");

                for(i = 0;i < library\_quantity;i++) {

                    fprintf(p1,"%s %s %s %s %s %lf\n",books[i].id,books[i].name,books[i].author,

                        books[i].press,books[i].category,books[i].price);

                }

                fclose(p1);

                printf("已经成功保存新的书籍信息。\n");

                break;

            }

        }

    }

}

/\*根据图书名称显示图书的信息\*/

void search\_book()

{

    char search[30] = "";  // 将要查找的字段放到该数组中，可以为id,name,author

    char compare[30] = ""; // 将书库中的字段放到该数组中

    BOOKINFO books[200];

    FILE \*p1;

    char choice1, choice2;

    // choice1,用于选择用什么字段查询

    // choice2,用于是否继续查询

    int flag1; // flag1 = 1，查询的字段是作者,一位作者可以对应多本书的情况

    int flag2; // flag2 = 1，是否查询到符合要求的书籍

    int i;

    system("cls");

    fflush(stdin);

    library\_quantity = 0;

    p1 = fopen("library.txt","r");

    while(fscanf(p1, "%s %s %s %s %s %lf",books[library\_quantity].id, books[library\_quantity].name, books[library\_quantity].author,

                books[library\_quantity].press, books[library\_quantity].category,&books[library\_quantity].price) == 6)

    {

        library\_quantity++;

    }

    fclose(p1);

    is\_library\_empty(library\_quantity);

    while (1) {

        printf("您想通过哪种方式查询书籍?\n");

        printf("1.编号\n2.书名\n3.作者\n");

        choice1 = getchar();

        while (choice1 != '1' && choice1 != '2' && choice1 != '3') {

            choice1 = getchar();

        }

        printf("请输入查找的字段信息: ");

        fflush(stdin);

        gets(search);

        flag1 = 1;

        flag2 = 0;

        for (i = 0; i < library\_quantity; i++) {

            switch (choice1) {

            case '1':

                strcpy(compare, books[i].id);

                flag1 = 0; break;

            case '2':

                strcpy(compare, books[i].name);

                flag1 = 0; break;

            case '3':

                strcpy(compare, books[i].author);

                break;

            default:

                break;

            }

            if (strcmp(search, compare) == 0) {

                printf("|--------------------------图书信息-------------------------|\n");

                printf("|编号  书名        作者      出版社          类别    单价   |\n");

                printf("|                                                           |\n");

                printf("|%-6s%-12s%-10s%-15s%-8s%-8.1f|\n",

                       books[i].id, books[i].name, books[i].author,

                       books[i].press, books[i].category, books[i].price);

                flag2 = 1;

                if (!flag1) { // flag1 = 1，查询字段是作者

                    break;

                }

            }

        }

        if (!flag2) {

            printf("没有找到符合您要求的书籍!\n");

        }

        printf("是否继续查询?(Y/N):");

        choice2 = getchar();

        while (choice2 != 'Y' && choice2 != 'N') {

            choice2 = getchar();

        }

        if (choice2 == 'N') {

            break;

        }

        fflush(stdin);

    }

}

void borrow\_book(char account[20])

{

    BOOKINFO books[200];

    FILE \*p1,\*p3;

    char search[30];

    char choice;

    int flag;

    int i;

    system("cls");

    fflush(stdin);

    library\_quantity = 0;

    p1 = fopen("library.txt","r");

    while(fscanf(p1, "%s %s %s %s %s %lf",books[library\_quantity].id, books[library\_quantity].name, books[library\_quantity].author,

                books[library\_quantity].press, books[library\_quantity].category,&books[library\_quantity].price) == 6)

    {

        library\_quantity++;

    }

    fclose(p1);

    is\_library\_empty(library\_quantity);

    while (1)

    {

        printf("请输入要借的书籍名称:\n");

        fflush(stdin);

        gets(search);

        for(i = 0;i < library\_quantity;i++) {

            if(strcmp(books[i].name,search) == 0) {

                break;

            }

        }

        if(i >= library\_quantity) {

            printf("书库中没有《%s》的书籍信息。\n",search); //是否继续借书

        } else {

            library\_quantity--;

            p3=fopen("memberbook.txt","a");

            fprintf(p3,"%s %s %s %s %s %f %s\n",

                    books[i].id,books[i].name,books[i].author,books[i].press,

                    books[i].category,books[i].price,account);

            fclose(p3);

            printf("借书成功!\n"); /\*更新书库\*/

        }

        printf("您是否继续借书?(Y/N):");

        choice = getchar();

        while (choice != 'Y' && choice != 'N')

        {

            choice = getchar();

        }

        if(choice == 'N') {

            break;

        }

    }

}

void return\_book(char account[20])

{

    FILE \*p3;

    BOOKLOANS bookgrasp[100];

    char search[30];

    char choice;

    int i;

    int book\_loans = 0;//总的被借的书籍数量

    int member\_loans = 0;//该用户借阅书籍数量

    int memberbooks[30]; //用户借阅书籍在bookgrasp中的数组下标

    system("cls");

    fflush(stdin);

    p3= fopen("memberbook.txt", "r");

    while (fscanf(p3, "%s %s %s %s %s %lf %s", bookgrasp[book\_loans].id, bookgrasp[book\_loans].name, bookgrasp[book\_loans].author, bookgrasp[book\_loans].press,

                bookgrasp[book\_loans].category, &bookgrasp[book\_loans].price,bookgrasp[book\_loans].member\_id) != EOF) {

        book\_loans++;  //所有的借书信息

    }

    fclose(p3);

    for(i = 0;i < book\_loans;i++) {

        if(strcmp(bookgrasp[member\_loans].member\_id,account) == 0) {

            memberbooks[member\_loans++] = i; //该用户的借书信息

        }

    }

    if(member\_loans <= 0) {

        printf("您没有任何借书信息。\n");

        printf("请按任意键返回。\n");

        getchar();  //终止

    } else {

        printf("您所借阅的所有图书信息如下:\n");

        printf("|----------------------------图书借阅信息--------------------|\n");

        printf("|编号  书名        作者      出版社          类别    单价    |\n");

        for(i = 0;i < member\_loans;i++) {

            int ret = memberbooks[i];

            printf("|%-6s%-12s%-10s%-16s%-8s%-8.1f|\n",bookgrasp[ret].id,

                        bookgrasp[ret].name,bookgrasp[ret].author,bookgrasp[ret].press,

                        bookgrasp[ret].category,bookgrasp[ret].price);

        }

        while (1)

        {

            printf("\n输入要归还的书本名:\n");

            gets(search);

            fflush(stdin);

            int j; //用j保存找到的search

            for(j = 0;j < member\_loans;j++) {

                int ret = memberbooks[j];

                if(strcmp(bookgrasp[ret].name,search) == 0) {

                    break;

                }

            }

            if(j >= member\_loans) {

                printf("您没有这条借书信息!请核对书名。\n");

            } else {

                book\_loans--;

                for(i = memberbooks[j];i < book\_loans;i++) {

                    bookgrasp[i] = bookgrasp[i+1];  //数组越界

                }

                member\_loans--;

                for(i = j;i < member\_loans;i++) {

                    memberbooks[i] = memberbooks[i+1]; //数组越界

                }

                printf("还书成功!\n");

            }

            printf("是否继续还书?(Y/N):");

            choice = getchar();

            while (choice != 'Y' && choice != 'N')

            {

                choice = getchar();

            }

            if(choice == 'N') {

                p3 = fopen("memberbook.txt","w");

                for(i = 0;i < book\_loans;i++) {

                    fprintf(p3, "%s %s %s %s %s %lf %s\n", bookgrasp[i].id, bookgrasp[i].name, bookgrasp[i].author, bookgrasp[i].press,

                            bookgrasp[i].category, bookgrasp[i].price,bookgrasp[i].member\_id);

                }

                fclose(p3);

                break;

            }

        }

    }

}

/\*显示全部图书信息\*/

void display\_book()

{

    FILE \*p1;

    BOOKINFO books[200];

    int i;

    system("cls");

    fflush(stdin);

    library\_quantity = 0;

    p1 = fopen("library.txt","r");

    while(fscanf(p1, "%s %s %s %s %s %lf",books[library\_quantity].id, books[library\_quantity].name, books[library\_quantity].author,

                books[library\_quantity].press, books[library\_quantity].category,&books[library\_quantity].price) == 6)

    {

        library\_quantity++;

    }

    fclose(p1);

    is\_library\_empty(library\_quantity);

    printf("\t\t\t书库中共有%d本书。\n", library\_quantity);

    printf("|------------------------图书信息-------------------------|\n");

    printf("|编号  书名        作者      出版社          类别    单价  |\n");

    printf("|                                                        |\n");

    for (i = 0; i < library\_quantity; i++) {

        printf("|%-6s%-12s%-10s%-16s%-8s%-8.1f|\n",

               books[i].id, books[i].name, books[i].author,

               books[i].press, books[i].category, books[i].price);

    }

    printf("\n按任意键返回上一级。\n");

    getchar();

    fflush(stdin);

}

/\*会员信息系统层级\*/

/\*个人账户信息\*/

int user\_account(char account[20])

{

    MEMBERINFO members[200];

    BOOKLOANS bookgrasp[200]; //会员借出书籍信息表

    int loans\_pos[100]; //用于存放用户借的书在BOOKLOANS books中的数组下标

    int totally\_borrow = 0; //用户借书数量计数/借书总量

    char old\_password[30];

    char new\_password1[30];

    char new\_password2[30];

    char choice1; //是否修改密码

    char choice2; //是否继续修改密码

    FILE \*p2, \*p3;

    int i;

    system("cls");

    member\_number = 0;

    p2 = fopen("infomember.txt", "r");

    while (fscanf(p2, "%s %s %s %s %s", members[member\_number].id, members[member\_number].password,

               members[member\_number].name, members[member\_number].sex, members[member\_number].phone) == 5)

    {

        member\_number++;

    }

    fclose(p2);

    book\_loans = 0;

    p3 = fopen("memberbook.txt", "r");

    while (fscanf(p3, "%s %s %s %s %s %lf %s", bookgrasp[book\_loans].id, bookgrasp[book\_loans].name, bookgrasp[book\_loans].author,

                  bookgrasp[book\_loans].press, bookgrasp[book\_loans].category, &bookgrasp[book\_loans].price, bookgrasp[book\_loans].member\_id) == 7)

    {

        book\_loans++;

    }

    fclose(p3);

    for (i = 0; i < member\_number; i++) {

        if (strcmp(members[i].id, account) == 0) {

            break;

        }

    }

    if (i >= member\_number) {

        printf("未找到用户信息。\n");

        return 0;

    }

    printf("\n");

    printf("|---------------------------个人信息-----------------------------------|\n");

    printf("|姓名        性别      ID              密码             手机号         |\n");

    printf("|%-12s%-10s%-16s%-17s%-15s|\n",members[i].name, members[i].sex, members[i].id, members[i].password, members[i].phone);

    printf("|                                                                      |\n");

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

    for (int j = 0; j < book\_loans; j++) {

        if (strcmp(bookgrasp[j].member\_id, account) == 0) {

            loans\_pos[totally\_borrow++] = j;

        }

    }

    if (totally\_borrow <= 0) {

        printf("您在图书馆中没有任何借书信息。\n");

    } else {

        printf("|--------------------------图书借阅信息---------------------------|\n");

        printf("|编号       书名      作者      出版社       类别       单价      |\n");

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

        for (int j = 0; j < totally\_borrow; j++) {

            printf("|%-9s%-8s%-8s%-10s%-9s%-8.1f|\n",

                   bookgrasp[loans\_pos[j]].id, bookgrasp[loans\_pos[j]].name, bookgrasp[loans\_pos[j]].author,

                   bookgrasp[loans\_pos[j]].press, bookgrasp[loans\_pos[j]].category, bookgrasp[loans\_pos[j]].price);

        }

        if (strcmp(members[i].sex, "male") == 0) {

            printf("%s先生,您当前借阅的书籍共有%d本。\n", members[i].name, totally\_borrow);

        } else if (strcmp(members[i].sex, "female") == 0) {

            printf("%s女士,您当前借阅的书籍共有%d本。\n", members[i].name, totally\_borrow);

        } else {

            printf("%s,您当前借阅的书籍共有%d本。\n", members[i].name, totally\_borrow);

        }

    }

    printf("是否需要修改密码?(Y/N): ");

    fflush(stdin);

    choice1 = getchar();

    while (choice1 != 'Y' && choice1 != 'N') {

        choice1 = getchar();

    }

    if (choice1 == 'Y') {

        while (1) {

            printf("请输入原来的密码:\n");

            fflush(stdin);

            gets(old\_password);

            if (strcmp(old\_password, members[i].password) == 0) {

                printf("密码正确!请输入新密码:\n");

                fflush(stdin);

                gets(new\_password1);

                printf("请再次输入新密码:\n");

                fflush(stdin);

                gets(new\_password2);

                if (strcmp(new\_password1, new\_password2) == 0) {

                    printf("密码修改成功!\n");

                    strcpy(members[i].password, new\_password2);

                    p2 = fopen("infomember.txt", "w");

                    for (int j = 0; j < member\_number; j++) {

                        fprintf(p2, "%s %s %s %s %s\n", members[j].id, members[j].password,

                                members[j].name, members[j].sex, members[j].phone);

                    }

                    fclose(p2);

                    return 1;

                } else {

                    printf("两次输入的密码不一致，是否继续修改密码?(Y/N):\n");

                    fflush(stdin);

                    choice2 = getchar();

                    while (choice2 != 'Y' && choice2 != 'N') {

                        choice2 = getchar();

                    }

                    if (choice2 == 'N') {

                        break;

                    }

                }

            } else {

                printf("密码错误，是否重新输入?(Y/N):");

                fflush(stdin);

                choice2 = getchar();

                while (choice2 != 'Y' && choice2 != 'N') {

                    choice2 = getchar();

                }

                if (choice2 == 'N') {

                    break;

                }

            }

        }

    }

    return 0; /\* 成功修改密码后，返回 1

                 其他情况下，返回 0 \*/

}

 /\*查重会员id\*/

int check\_repeat\_id(char id[30])

{

    MEMBERINFO members[200];

    FILE \*p2;

    int i;

    member\_number = 0;

    p2 = fopen("infomember.txt", "r");

    while (fscanf(p2, "%s %s %s %s %s", members[member\_number].id, members[member\_number].password,

               members[member\_number].name, members[member\_number].sex, members[member\_number].phone) == 5)

    {

        member\_number++;

    }

    fclose(p2);

    for(i = 0;i < member\_number;i++) {

        if(strcmp(members[i].id,id) == 0) {

            return 1;

        }

    }

    return 0;

}

/\*密码比对函数 \*/

int compare\_password(char \*password)

{

    char input[50]; // 提供足够长度以确保能够储存任何输入

    printf("\t\t\t请输入密码: ");

    gets(input);

    if (strcmp(input, password) == 0) {

        return 1; // 密码正确

    } else {

        return 0;

    }

}

/\*会员注册函数\*/

void member\_login\_up()

{

    MEMBERINFO member;

    FILE \*p2;

    char choice1, choice2;

    while(1) {

        system("cls");

        fflush(stdin);

        printf("请输入您想注册的id名: ");

        gets(member.id);

        if(check\_repeat\_id(member.id)) {

            printf("您输入的id已被使用!\n");

        } else {

            printf("请输入您的密码: (请不要告诉他人！) ");

            gets(member.password);

            printf("请输入您的昵称: ");

            gets(member.name);

            while(1) {

                printf("请输入您的性别: (male/female) ");

                gets(member.sex);

                if(strcmp(member.sex, "male") == 0 || strcmp(member.sex, "female") == 0) {

                    break;

                } else {

                    printf("性别输入有误，请重新输入!\n");

                }

            }

            printf("请输入您的手机号: ");

            gets(member.phone);

            printf("是否保存此次注册信息？(Y/N)");

            fflush(stdin);

            choice1 = getchar();

            while (choice1 != 'Y' && choice1 != 'N') {

                choice1 = getchar();

            }

            if(choice1 == 'Y') {

                p2 = fopen("infomember.txt", "a");

                fprintf(p2, "%s %s %s %s %s\n", member.id, member.password,

                        member.name, member.sex, member.phone);

                fclose(p2);

                printf("您的账户已注册成功!\n");

            } else {

                printf("您本次未注册。\n");

            }

        }

        printf("是否重新注册?(Y/N):");

        fflush(stdin);

        choice2 = getchar();

        while (choice2 != 'Y' && choice2 != 'N') {

            choice2 = getchar();

        }

        if(choice2 == 'N') {

            break;

        }

    }

}

/\* 会员登录函数 \*/

int member\_login\_in(char account[20])

{

    FILE \*p2;

    MEMBERINFO members[200];

    char choice1, choice2;

    int flag1 = 0; // 用于判断登录的会员信息是否注册

    int i;

    member\_number = 0;

    p2 = fopen("infomember.txt", "r");

    while (fscanf(p2, "%s %s %s %s %s\n", members[member\_number].id, members[member\_number].password,

               members[member\_number].name, members[member\_number].sex, members[member\_number].phone) != EOF)

    {

        member\_number++;

    }

    fclose(p2);

    // 查找账号是否存在

    for (i = 0; i < member\_number; i++) {

        if (strcmp(members[i].id, account) == 0) {

            flag1 = 1;

            break;

        }

    }

    if (flag1) {

        while (1) {

            char password\_input[20];

            printf("\t\t\t   请输入密码: ");

            gets(password\_input);

            if (strcmp(members[i].password, password\_input) == 0) {

                printf("\n\n\n\t\t\t   核对正确！正在进入会员界面...");

                clock\_delay();

                return 1;

            } else {

                printf("\n\n\n\t\t\t   您的密码输入有误!\n");

                printf("\t\t\t   您是否要继续输入密码(Y/N):");

                fflush(stdin);

                choice1 = getchar();

                while (choice1 != 'Y' && choice1 != 'N') {

                    choice1 = getchar();

                }

                if (choice1 == 'N') {

                    return 0;

                }

            }

        }

    } else {

        printf("\n\n\n\t\t\t  该账号还未注册, 是否需要注册会员(Y/N):");

        fflush(stdin);

        choice2 = getchar();

        while (choice2 != 'Y' && choice2 != 'N') {

            choice2 = getchar();

        }

        if (choice2 == 'Y') {

            member\_login\_up();

            return 2;

        } else {

            return 0;

        }

    }

}

/\*查看会员信息函数\*/

void ADMIN\_watch\_member()

{

    FILE \*p2;

    int i;

    MEMBERINFO members[200];

    system("cls");

    fflush(stdin);

    member\_number = 0;

    p2 = fopen("infomember.txt", "r");

    while (fscanf(p2, "%s %s %s %s %s\n", members[member\_number].id, members[member\_number].password,

               members[member\_number].name, members[member\_number].sex, members[member\_number].phone) !=EOF)

    {

        member\_number++;

    }

    fclose(p2);

    if(member\_number <= 0) {

        printf("没有任何会员信息!\n");

    } else {

        //显示会员信息

        printf("系统注册用户有%d位\n",member\_number);

          printf("\n");

        printf("|---------------------------个人信息-----------------------------------|\n");

        printf("|姓名        性别      ID              密码             手机号         |\n");

        for (i = 0; i < member\_number; i++) {

            printf("|%-12s%-10s%-16s%-17s%-15s|\n",members[i].name, members[i].sex, members[i].id, members[i].password, members[i].phone);

        }

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

    }

    printf("按任意键返回上一级。\n");

    getchar();

}

/\*查看所有图书借阅信息函数\*/

void display\_memberbook()

{

    FILE \*p3;

    BOOKLOANS bookgrasp[200];

    int book\_loans = 0;

    system("cls");

    book\_loans = 0;

    p3= fopen("memberbook.txt", "r");

    while (fscanf(p3, "%s %s %s %s %s %lf %s", bookgrasp[book\_loans].id, bookgrasp[book\_loans].name, bookgrasp[book\_loans].author, bookgrasp[book\_loans].press,

                bookgrasp[book\_loans].category, &bookgrasp[book\_loans].price,bookgrasp[book\_loans].member\_id) != EOF) {

        book\_loans++;  //所有的借书信息

    }

    fclose(p3);

    if(book\_loans <= 0) {

        printf("当前没有借阅信息");

    }

    else {

        printf("当前共有%d书籍被借阅。",book\_loans);

        printf("\n");

        printf("|---------------------------借阅图书信息-----------------------------------|\n");

        printf("|编号  书名        作者      出版社          类别    单价    借阅ID        |\n");

        printf("|                                                                          |\n");

        for(int i=0;i<book\_loans;i++) {

            printf("|%-4s%-10s%-8s%-13s%-6s%-6.1f%-12s|\n",bookgrasp[i].id,

                    bookgrasp[i].name,bookgrasp[i].author,bookgrasp[i].press,

                    bookgrasp[i].category,bookgrasp[i].price,bookgrasp[i].member\_id);

        }

    }

    printf("\n\n输入任意键返回...");

    fflush(stdin);

    getchar();

}