



**C语言程序设计项目报告**

学院 物理与光电工程学院

专业 光电信息科学与工程

班级 22级光电信息科学与工程(3)班

题目 酒店房间管理系统

组长 蒋卓然 3122006814

组员 陈熙宇 3122006806 麦观钊 3122006829

刘纪颖 3122006822 温仲鸣 3122006835

2023年5月20日

广东工业大学物理与光电工程学院制

|  |  |
| --- | --- |
| 课程设计任务书 | |
| 一、课程设计题目 | 学习运用 C 语言程序设计的知识，利用合适的 C 编译环境为某个应用背景设计和开发 一个实用的小型系统。 |
| 问二、课程设计内容 | |
| 完成一个应用系统的设计全过程：需求分析、用户界面设计、算法设计、代码实现、系统 测试（包括测试数据的设计）等。 课程设计步骤及每个步骤的内容如下：  1、需求分析  画出系统框图，并进行功能需求分析、信息需求分析。  2、用户界面设计  设计提示清晰、操作方便的界面，包括菜单、输入界面、输出界面等。  3、算法设计  画出系统总体流程图及各个模块的程序流程图（可以用传统流程图或者 N-S流程图）。  4、代码实现  编写代码，实现各个模块。  5、系统测试  设计实用有效的测试数据，对系统功能进行测试。 | |
|
|
|
|
| 建三、课程设计要求议**处理方法：** | |
| 1、设计环境  不限制应用开发环境和工具：Visual C++ 、Turbo C 等均可。  2、技术要求  （1） 使用文件保存读入和读出的数据  （2） 使用结构体数组存储数据（可以将结构体数组定义为静态或者动态）  （3） 参考提示：  1)参考教辅书案例 13.3 2)参考教材例题 10.4 3)C 语言课程设计报告  3、系统基本功能要求  （1） 系统以菜单方式工作  （2） 系统具有信息录入功能  （3） 系统具有信息浏览功能  （4） 系统具有信息查询功能  （5） 系统具有信息修改功能  （6） 系统具有信息删除功能  （7） 系统必须考虑一定的安全性  4、系统演示  每个小组进行5-10 分钟的系统演示和答辩，讲解设计方案，演示系统运行并汇报合作和分工情况。  5、任务要求  任务完成形式：为了培养学生的团队精神，本课程设计以小组为单位完成任务。小组成员采用自愿组合的原则分组（但为了讨论方便，建议以宿舍为单位进行分组），每组的成员以5 人为最佳。每个小组上交课程设计报告并进行答辩，报告内容及撰写规范参考后续内容。 | |
|
|
|
| 四、课程设计报告要求位**处理意见：** | |
| 课程设计报告为 WORD 文档，整个报告是一个长文档格式，由封面、任务书、摘要、  小组分工情况、目录、正文五部分组成。其中摘要主要说明创建系统使用的工具、系统的功能、方案设计（使用的设计技术）及结果情况，约 200 字左右；封面参见附件；目录通过“插入”→“引用”→“索引与目录”自动生成三级目录结构。  课程设计报告一般应包含以下主要内容。  1、需求分析（分析介绍系统功能）  画出系统功能结构图，并进行功能需求分析、信息需求分析等。  2、总体设计方案  1）设计技术说明  介绍设计过程中使用的主要知识点，如指针、数组、结构体、文件等。  2）结构体设计及说明  3）函数关系示意图（主函数及子函数的关系）  4）总体设计的流程图（可以用传统流程图或者 N-S 流程图）  3、函数设计与实现  具有代表性的程序运行过程的截图、功能的实现方法的说明和主要代码（要有较完整的注释），要求能够体现程序设计中的重点和难点及设计者的能力及主要工作。  1）主函数 （函数名、功能、主要变量及说明、流程图及说明、代码及注释）  2）插入函数 （函数名、功能、主要变量及说明、流程图及说明、代码及注释）  3）浏览函数 （函数名、功能、主要变量及说明、流程图及说明、代码及注释）  ……  4、系统测试  系统测试方案和测试数据。包括系统调试方法，调试过程中遇到的问题及解决办法。设计实用有效的测试数据，对系统功能进行测试。  5、系统使用说明  6、总结  包括设计者对这次课程设计的收获、感想和体会，系统的优点和不足，系统进一步完善的设想等。  8、致谢（不排编号）：对在设计中提供帮助的学生和教师表示感谢。  9、参考文献（不排编号）：列出课程设计过程中参考过的文献资料  10、附录（不排编号）：程序的完整代码 | |
|
|
|
| 五、合作方式**处理结果的意见：** | |
| 合作人数原则上为 5 人，并符合以下要求，否则根据情况成绩下调一个档次或成绩为不及格。  （1） 小组成员既要有相互合作的精神，又要分工明确（每人至少负责一个模块的完整设计，包括需求分析、流程图、代码、测试、结果分析、撰写相关部分的报告等）。每个学生都必须充分了解整个设计的全过程。从开始的系统需求分析到最后的软件测试，都要有详细的计划，设计文档应按照要求书写。  （2） 在报告中明确标明合作者的学号姓名，并说明分工情况。 | |
|
|
|

**摘 要**

随着个人计算机的普及，越来越多的重复性工作被交由计算机来完成。在这种大趋势下，传统的纸质档案管理方式已不适应现代服务业的发展。数字化的管理成为酒店等行业的必要条件。酒店客房管理的电脑化对于优化员工工作效率、降低成本以及改善服务质量等方面都有十分积极的作用。

本项目基于C语言的酒店房间管理系统，旨在为酒店提供一个高效、自动化的房间管理解决方案。该系统实现了客房信息管理、预订管理、入住管理、结账管理等功能，使得酒店工作人员可以更加轻松地处理房间管理相关事宜，同时提高了客户满意度和酒店竞争力。

本程序运用Dev-C++编译器编写，涉及C语言文件的读写、结构体链表的应用以及字符串库函数、C++界面可视化函数、日期功能算法、用户输入限制等相关知识。经过不断的优化，该程序的代码健壮性、用户体验感以及功能安全性都达到了较高的水平，满足了实际应用场景中的大部分需求。

**关键词**：C/C++；单链表；酒店房间；数字化管理；权限分离；体验感；自动化

**小 组 分 工 情 况**

**以下是小组项目的工作表，其中组长提供技术帮助并审核完成情况**

|  |  |  |  |
| --- | --- | --- | --- |
| **项 目 文 档 工 作 表** | | | |
| 负责人 | 负责模块 | 备注 | 完成情况 |
| **蒋卓然** | 总体设计方案及总结 | 项目最终解释权所有人 | √ |
| 陈熙宇 | 封面及任务书 | 项目任务对接人 | √ |
| 麦观钊 | 系统测试 | 项目上线把关人 | √ |
| 刘纪颖 | 使用说明 | 项目使用讲解员 | √ |
| 温仲鸣 | 程序分析 | 项目市场分析员 | √ |
| 函数设计与实现模块的报告由相关代码负责人撰写，详情参考程序模块工作表 | | | |

表 1-项目文档工作表

|  |  |  |  |
| --- | --- | --- | --- |
| **程 序 模 块 工 作 表** | | | |
| 负责人 | 负责模块 | 备注 | 函数及注释完成情况 |
| **蒋卓然** | 管理端模块 | 额外负责光标等组件类函数 | √ |
| 陈熙宇 | 客户端模块  游客端模块 | 额外负责代码风格检查 | √ |
| 温仲鸣 | 额外负责函数声明区域 | √ |
| 麦观钊 | 额外负责头文件引用整合 | √ |
| 刘纪颖 | 登录面模块 | 额外负责结构体 | √ |
| 退出当前界面以及关闭程序的实现引用相关文献资料 | | | |

表 2-程序模块工作表

目录

[封 面 1](#_Toc31010)

[课程设计任务书 2](#_Toc2036)

[摘 要 4](#_Toc8654)

[小 组 分 工 情 况 5](#_Toc22722)

[一、 程序分析 7](#_Toc22694)

[（一）功能需求分析 7](#_Toc19106)

[（二） 信息需求分析 7](#_Toc3014)

[（三）系统功能结构 8](#_Toc17345)

[二、 总体设计方案 9](#_Toc10123)

[（一） 设计技术说明 9](#_Toc7978)

[（二） 结构体设计及说明 10](#_Toc15560)

[（三） 函数关系示意图 12](#_Toc20015)

[（四） 总体设计流程图 13](#_Toc13350)

[三、 函数设计与实现 13](#_Toc26861)

[（一） 登录面主要函数与流程图 13](#_Toc14082)

[（二） 客户端主要函数与流程图 21](#_Toc26171)

[（三） 管理端主要函数与流程图 39](#_Toc12312)

[四、 系统测试 55](#_Toc8219)

[（一） 测试标准 55](#_Toc10931)

[（二） 测试数据 58](#_Toc28852)

[（三） 测试结果 60](#_Toc19520)

[五、 使用说明 63](#_Toc21536)

[六、 总结 64](#_Toc17256)

[（一） 心得与体会 64](#_Toc7562)

[（二） 优点与不足 66](#_Toc8486)

[（三） 优化与设想 66](#_Toc5435)

致 谢......................................................................................................................................................................69

参考文献..................................................................................................................................................................70

[附 录 71](#_Toc30362)

# 程序分析

## （一）功能需求分析

本系统程序主要面向酒店管理者和顾客，功能需求如下：

* 系统需要实现用户注册、登录、修改密码等基本的账号管理功能。
* 管理员可以对房间进行管理，包括添加、删除、修改房间信息，查看房间状态等。
* 顾客可以浏览酒店房间信息，并进行预定、取消预定、入住、退房等操作。
* 系统需要能够提供房间查询功能，顾客和管理员均可以通过订单号或者时间段进行查询。
* 系统需要支持酒店房间结算功能，涉及房间单价和入住时长等。
* 系统需要提供数据报表功能，以便管理员随时查看经营状况，能需求分析。
* 对房间或预定信息的修改应逐个地进行，一个房间或预定信息的更改不应影响其他的房间或预定记录。
* 所设计的系统应以菜单方式工作，应为用户提供清晰的使用提示，依据用户的选择来进行各种处理，并要求在此过程中能够尽可能的兼容用户使用过程中的异常情况。

## 信息需求分析

本系统主要面向对象为酒店房间和系统用户，信息需求如下：

* 用户信息：包括顾客和管理员的个人信息，例如姓名、性别、联系方式等。
* 房间信息：包括房间类型、价格、可住人数、是否被预定等。此外还需要记录每个房间的入住时间、退房时间等信息，以便系统能够计算出在特定时间段内的房间可用情况。
* 预订信息：包括房间信息、预定用户、预订时间、预订状态等。其中，预订状态需要标识为待审核、已通过、已拒绝等。
* 费用信息：包括结算用户、房间信息、入住时间、退房时间、金额等。
* 支付信息：包括支付金额、支付状态等。支付金额需要通过一系列算法来实现。
* 报表信息：包括经营状况、房间详情等。需要根据上述各类信息进行统计分析，并生成相应的报表。

## （三）系统功能结构

本系统分为两个模块，分别是管理员模块以及用户端模块，这两个模块分别实现了对酒店房间信息的添加、查看、修改、删除以及统计等功能，其系统功能结构图如下：

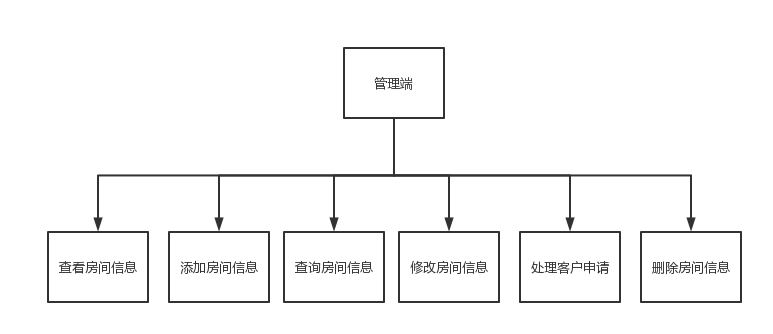


图 2-管理模块结构

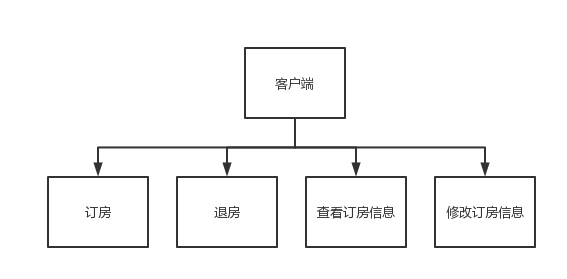


图 3-客户模块结构

# 总体设计方案

## 设计技术说明

本系统采用以C语言为主，C++为辅的原生技术栈。

**C语言**是一门面向过程的、抽象化的通用程序设计语言，广泛应用于底层开发。它能以简易的方式编译、处理低级存储器。C语言是仅产生少量的机器语言以及不需要任何运行环境支持便能运行的高效率程序设计语言。尽管C语言提供了许多低级处理的功能，但仍然保持着跨平台的特性，以一个标准规格写出的C语言程序可在包括类似嵌入式处理器以及超级计算机等作业平台的许多计算机平台上进行编译。

**C++（c plus plus）**是一种计算机高级程序设计语言，由C语言扩展升级而产生 ，最早于1979年由本贾尼·斯特劳斯特卢普在AT&T贝尔工作室研发。

它既可以进行C语言的过程化程序设计，又可以进行以抽象数据类型为特点的基于对象的程序设计，还可以进行以继承和多态为特点的面向对象的程序设计。C++擅长面向对象程序设计的同时，还可以进行基于过程的程序设计。其几乎可以创建任何类型的程序：游戏、设备驱动程序、HPC、云、桌面、嵌入式和移动应用等。 甚至用于其他编程语言的库和编译器也使用C++编写。同时，C++拥有计算机运行的实用性特征，同时还致力于提高大规模程序的编程质量与程序设计语言的问题描述能力。

## 结构体设计及说明

**本次设计的题目为酒店房间管理系统。首先需要设计存放信息的结构体信息，其结构体实现代码如下：**

#### 1、酒店房间结构体

struct room

{

char num[10]; //房间号

char type[11]; //房型

char site[10]; //位置

char status[10]; //状态

char check[10]; //入住

char checkOut[10]; //退房

char inputTime[14]; //加入时间

int year; //加入的年

int mon; //月

int day; //日

int days; //天数

char cost[10]; //费用

char name[10]; //姓名

char tel[12]; //电话

char idnum[20]; //身份证号

struct room \*next; //下一个结点地址

};

此结构体是用来创建房间信息链表的。其中包括了房间号、房型、位置、状态、入住时间、退房时间、加入时间、住店天数、费用以及客户信息等。通过这些成员变量，我们可以更好地管理酒店的房间信息，并实现对客户订单的查询和管理。同时，由于结构体内嵌有指向下一个结点的指针，我们可以方便地实现链表操作，如添加、删除、遍历等。

#### 2、用户数据结构体

struct ID

{

char power[10]; //权限

char account[20]; //账号

char key[20]; //密码

char keyProtect[20]; //密保

char check[10]; //入住

char checkOut[10]; //退房

char name[10]; //姓名

char idnum[20]; //身份证号

char num[10]; //房间号

char sex[3]; //性别

char inputName[10]; //入住时间

char inputTel[12]; //入住电话

int year; //入住时的年

int mon; //入住时的月

int day; //入住时的日

int years; //离开时的年

int month; //离开时的月

int days; //离开时的日

int numDay; //预定天数

int daymax; //离开日期的天数

int daymin; //入住日期的天数

int cost; //房间每日的价格

char information[100]; //信息

struct ID \* next; //下一个结点地址

};

此结构体用于记录酒店客户的详细信息，包括权限、账号、密码、密保、入住、退房情况、姓名、身份证号、房间号、性别、入住时间、入住电话、入住日期及预定天数、离开日期及天数、房间每日价格、信息等内容。同时，此结构体采用链表的方式组织，其中next指向下一个结点的地址。这种设计方便了对多个客户信息的管理和查询。

**最后基于该结构体链表实现对房间和用户信息的管理，实现酒店房间管理系统功能。**

## 函数关系示意图

由于本系统设计函数数量较多，现将其分为四类，各类函数关系如下图所示：

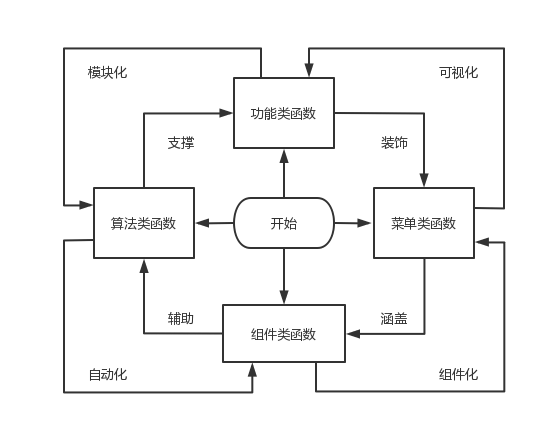


图 4-函数关系图

## 总体设计流程图

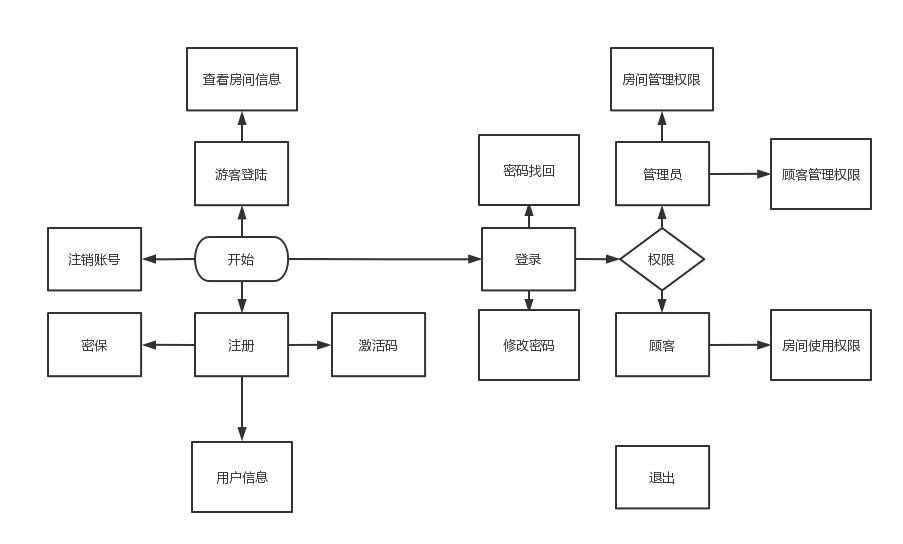


图 5-总体设计流程图

# 函数设计与实现

**本系统需要设计实现大量函数，下面选取了各部分的核心函数进行介绍：**

## 登录面主要函数与流程图

**1.登陆界主函数**

struct ID \*login()

{

struct ID \*head = NULL,\*p;

head = loginRead();

int x,i;

char t;

A:system("cls");

loginWrite(head);

loginMenu();

x = 9;

i = 0;

while (1)

{

t = arrows(x,'0','6');

switch(t)

{

case 72:

if (i > 0)

{

x = cColor(93,x,0);

if (x == 9)

puts("1.登录");

else if (x == 11)

puts("2.注册");

else if (x == 13)

puts("3.注销");

else if (x == 15)

puts("4.密码找回");

else if (x == 17)

puts("5.修改密码");

else if (x == 19)

puts("6.游客登录");

i--;

}

else

{

inversion(&i,93,&x,6,21,"0.退出");

}

break;

case 80:

if (i < 6)

{

x = cColor(93,x,1);

if (x == 11)

puts("2.注册");

else if (x == 13)

puts("3.注销");

else if (x == 15)

puts("4.密码找回");

else if (x == 17)

puts("5.修改密码");

else if (x == 19)

puts("6.游客登录");

else

puts("0.退出");

i++;

}

else

{

inversion(&i,93,&x,0,9,"1.登录");

}

break;

case 13: system("cls");

if (x == 9)

{

if ((p = sign(head)))

return p; //登录后返回该账号的基本信息

}

if (x == 11)

head = accountName(head);

if (x == 13)

head = logout(head);

if (x == 15)

signFind(head);

if (x == 17)

changePassword(head);

if (x == 19)

visitor();

if (x == 21)

exit(0);

goto A;

case 49:

system("cls");

ShowCursor();

if ((p = sign(head)))

return p;

goto A;

case 50: system("cls");ShowCursor();head = accountName(head); goto A;

case 51: system("cls");ShowCursor();head = logout(head); goto A;

case 52: system("cls");ShowCursor();signFind(head); goto A;

case 53: system("cls");ShowCursor();changePassword(head); goto A;

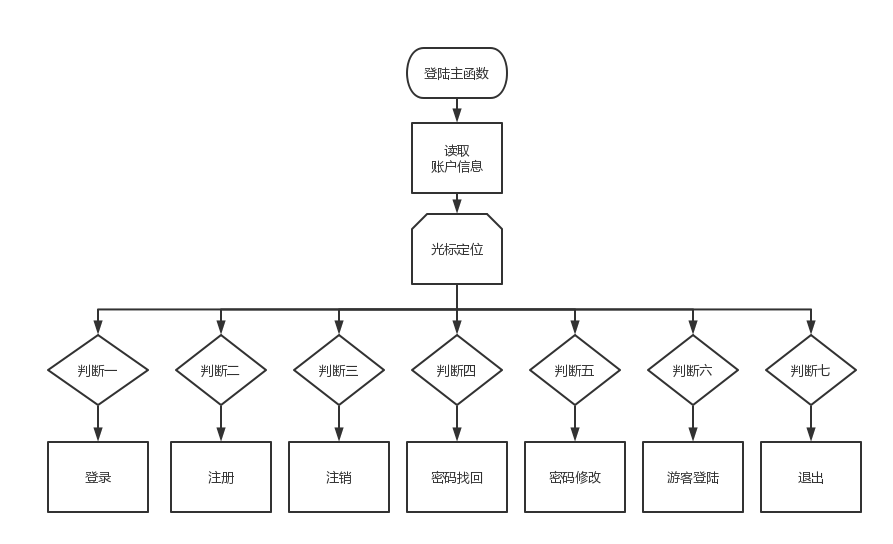
case 54: system("cls");ShowCursor();visitor(); goto A;

case 48: exit(0);

}

}

}



**2.用户登录函数**

struct ID \*sign(struct ID \*head)

{

int i = 2;

struct ID \*p1 = NULL,\*p2;

char accountt[20];

char key1[20];

char testGetCode[7];

char code[7];

if (head == NULL)

{

MessageBox(NULL,"未注册任何账号请注册后重试!","提示",MB\_ICONASTERISK);

return NULL;

}

A:form(9,2);

gotoxy(94,10);

puts("登录");

gotoxy(82,12);

puts("账号:");

gotoxy(82,14);

puts("密码:");

gotoxy(88,12);

if (p1 == NULL)

{

if (scanfPlus(accountt,1,18,0) == 1)

return NULL;

}

else

puts(accountt);

p1 = head;

while(1)

{

if (strcmp(accountt,p1->account) == 0)

break;

p1 = p1->next;

if (p1 == NULL)

{

MessageBox(NULL,"无此账号请重新输入!","错误", MB\_ICONHAND);

system("cls");

goto A;

}

}

gotoxy(88,14);

if (scanfPlus(key1,1,18,1) == 1)

return NULL;

C:system("cls");

form(9,2);

gotoxy(94,10);

printf("验证");

gotoxy(82,12);

printf("验证码:%s\n",randStr(testGetCode,4));

gotoxy(82,14);

printf("请输入验证码(区分大小写):");

gotoxy(107,14);

if (scanfPlus(code,1,4,0) == 1)

return NULL;

if (strcmp(testGetCode,code) != 0)

{

if (i>0)

{

system("cls");

gotoxy(80,14);

printf("验证码输入错误你还有%d次机会请重新输入!",i);

i--;

gotoxy(80,15);

system("pause");

}

else

{

MessageBox(NULL,"验证码输入错误你已经没有机会了!即将返回上一级界面","错误", MB\_ICONHAND);

system("cls");

return NULL;

}

goto C;

}

if (strcmp(key1,p1->key) != 0)

{

if (i>0)

{

system("cls");

gotoxy(80,14);

printf("密码输入错误你还有%d次机会请重新输入！",i);

gotoxy(80,15);

system("pause");

system("cls");

i--;

}

else

{

MessageBox(NULL,"密码输入错误你已经没有机会了!即将返回上一级界面","错误", MB\_ICONHAND);

system("cls");

return NULL;

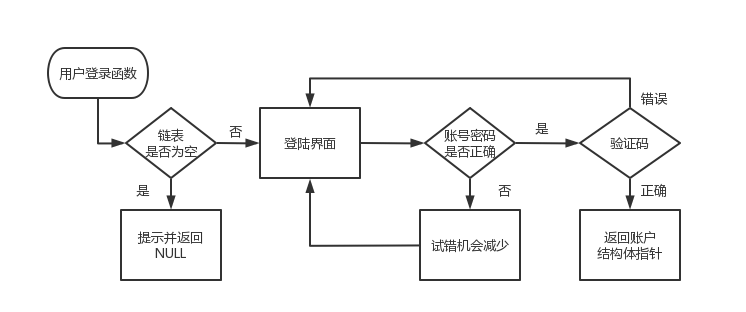
}

goto A;

}

return p1;

}



**3.账户信息读出文件**

struct ID \*loginRead()

{

FILE \*fp;

char ch;

struct ID \*head = NULL,\*p1,\*p2;

if ((fp = fopen("用户","rb")) == NULL)

return head;

A:p1 = (struct ID \*)malloc(sizeof(struct ID));

fread(p1,sizeof(struct ID),1,fp);

if(feof(fp))

{

fclose(fp);

return head;

}

if (head == NULL)

{

head = p1;

p1->next = NULL;

p2 = p1;

goto A;

}

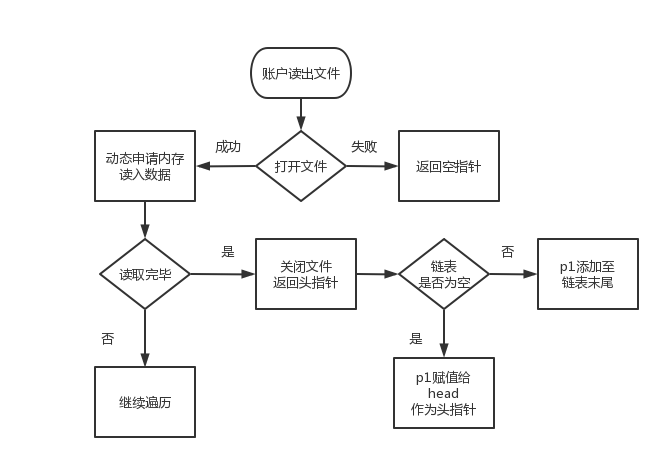
p2->next = p1;

p1->next = NULL;

p2 = p1;

goto A;

}



**4.账户信息写入文件**

void loginWrite(struct ID \*head)

{

FILE \*fp;

struct ID \*p1;

fp = fopen("用户","wb");

if (head == NULL)

return;

p1 = head;

while (p1 != NULL)

{

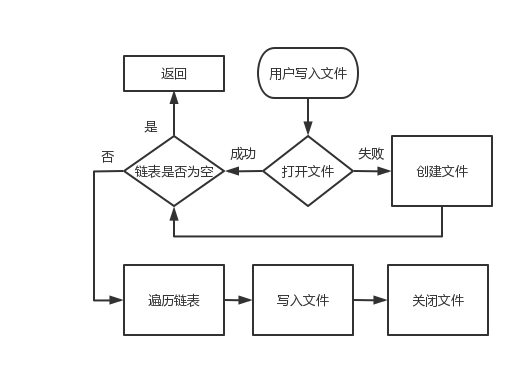
fwrite(p1,sizeof(struct ID),1,fp);

p1 = p1->next;

}

fclose(fp);

}



## 客户端主要函数与流程图

**1.客户端主函数**

int user(struct room \*head,struct ID \*p,struct ID \*head1)

{

struct room \*p1;

int x,i = 0;

char t;

system("cls");

gotoxy(80,14);

if (strcmp(p->sex,"男") == 0)

printf("尊敬的%s先生欢迎您光临446\n",p->name);

else

printf("尊敬的%s女士欢迎您光临446\n",p->name);

gotoxy(80,15);

system("pause");

system("cls");

p1 = head;

while (p1 != NULL)

{

if (strcmp(p1->num,p->num) == 0)

{

remind(p,head1);

break;

}

p1 = p1->next;

}

A:system("cls");

//write(head);

umenu(p);

head = read();

head1 = loginRead();

x = 12;

while (1)

{

t = arrows(x,'0','5');

switch(t)

{

case 72:

if (i > 0)

{

x = cColor(91,x,0);

if (x == 12)

puts("1.订房");

else if (x == 14)

puts("2.退房");

else if (x == 16)

puts("3.查看自己的订房信息");

else if (x == 18)

puts("4.修改自己的订房信息");

else

puts("5.退出登录");

i--;

}

else

{

inversion(&i,91,&x,5,22,"0.退出系统");

}

break;

case 80:

if (i < 5)

{

x = cColor(91,x,1);

if (x == 14)

puts("2.退房");

else if (x == 16)

puts("3.查看自己的订房信息");

else if (x == 18)

puts("4.修改自己的订房信息");

else if (x == 20)

puts("5.退出登录");

else

puts("0.退出系统");

i++;

}

else

{

inversion(&i,91,&x,0,12,"1.订房");

}

break;

case 13: system("cls");

if (x == 12)

reserve(head,p,head1);

else if (x == 14)

{

out(head,p,head1);

loginWrite(head1);

}

else if (x == 16)

userLook(head,p,head1);

else if (x == 18)

userChange(head,p,head1);

else if (x == 20)

return 0;

else

exit(0);

goto A;

case 49: system("cls");reserve(head,p,head1); goto A;

case 50: system("cls");out(head,p,head1);loginWrite(head1); goto A;

case 51: system("cls");userLook(head,p,head1); goto A;

case 52: system("cls");userChange(head,p,head1); goto A;

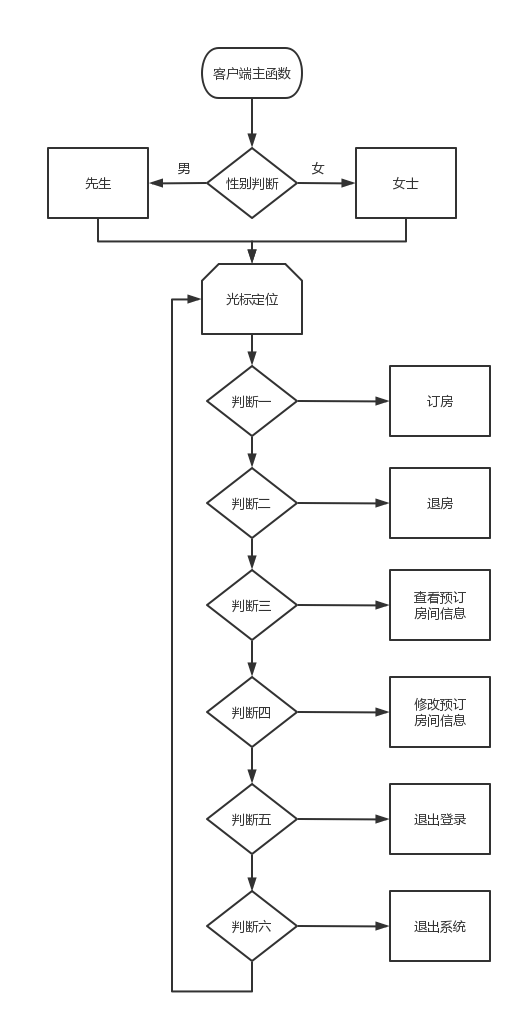
case 53: return 0;

case 48: exit(0);

}

}

}



**2.用户预定房间**

void reserve(struct room \*head,struct ID \*p,struct ID \*head1)

{

struct room \*p1,\*p2;

struct ID \*x1;

char tran[10];

char rNum[10];

int flag = 1,n=0,i,d;

int times[2][13] =

{

{0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31},

{0, 31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31}

};

if (head == NULL)

{

MessageBox(NULL,"本酒店没录入房间信息请稍后重试!","提示", MB\_ICONASTERISK);

return;

}

x1 = head1;

while (x1 != NULL)

{

if (strcmp(x1->account,p->account) == 0)

break;

x1 = x1->next;

}

p1 = head;

while (p1 != NULL)

{

if (strcmp(p1->num,x1->num) == 0)

{

MessageBox(NULL,"你已经预定了酒店房间请勿重复预定!","提示", MB\_ICONASTERISK);

return;

}

n++;

p1 = p1->next;

}

p1 = head;

while (p1 != NULL)

{

if (strcmp(p1->status,"已被预订") == 0)

strcpy(p1->status,"空闲");

p1 = p1->next;

}

puts("请输入你要入住的日期:");

gotoxy(25,0);

printf("年");

gotoxy(29,0);

printf("月");

gotoxy(33,0);

printf("日");

gotoxy(21,0);

if (scanfPlus(tran,4,4,3) == 1)

return;

p->years = p->year = atoi(tran);

gotoxy(27,0);

if (scanfPlus(tran,1,2,3) == 1)

return;

p->month = p->mon = atoi(tran);

gotoxy(31,0);

if (scanfPlus(tran,1,2,3) == 1)

return;

d = p->day = atoi(tran);

B:printf("请输入入住天数:");

if (scanfPlus(tran,1,15,3) == 1)

return;

p->days = p->numDay = atoi(tran);

if (p->days == 0)

{

MessageBox(NULL,"非法输入请重新输入！","错误", MB\_ICONHAND);

system("cls");

goto B;

}

while (1)

{

i = p->years % 4 == 0 && p->years % 100 != 0 || p->years % 400 == 0;

if (p->days + d > times[i][p->month])

{

p->days = p->days + d - times[i][p->month];

if (p->month != 12) p->month++;

else

{

p->years++;

p->month = 1;

}

d = 0;

}

if (d == 0)

break;

else

{

p->days = p->days + d;

break;

}

}

p = tranday(p);

p1 = head;

while (p1 != NULL)

{

x1 = head1;

while (x1 != NULL)

{

if (strcmp(x1->num,p1->num) == 0)

{

if (p->daymin < x1->daymin && p->daymax > x1->daymin)

strcpy(p1->status,"已被预订");

else if (p->daymin > x1->daymin && p->daymax < x1->daymax)

strcpy(p1->status,"已被预订");

else if (p->daymin < x1->daymax && p->daymax >x1->daymax)

strcpy(p1->status,"已被预订");

else if (p->daymin == x1->daymin)

strcpy(p1->status,"已被预订");

}

x1 = x1->next;

}

p1 = p1->next;

}

A:i = form4(n,head);

gotoxy(63,10+2\*i);

printf("请输入你想要的订的房间的房间号:");

if (scanfPlus(rNum,1,9,0) == 1)

return;

p1 = head;

while(p1 != NULL)

{

if (strcmp(p1->num,rNum) == 0 )

{

if (strcmp(p1->status,"空闲") != 0)

{

flag = 2;

break;

}

else

{

flag = 0;

break;

}

}

p1 = p1->next;

}

if (flag == 1)

{

MessageBox(NULL,"未查到此房间信息请重新输入!","提示", MB\_ICONASTERISK);

system("cls");

goto A;

}

if (flag == 2)

{

MessageBox(NULL,"该房间已被预定请重新输入!","提示", MB\_ICONASTERISK);

system("cls");

goto A;

}

x1 = head1;

while (x1 != NULL)

{

if (strcmp(x1->idnum,p->idnum) == 0)

{

strcpy(x1->num,p->num);

x1->year = p->year;

x1->years = p->years;

x1->mon = p->mon;

x1->month = p->month;

x1->day = p->day;

x1->days = p->days;

x1->daymin = p->daymin;

x1->daymax = p->daymax;

x1->numDay = p->numDay;

break;

}

x1 = x1->next;

}

gotoxy(63,12+2\*i);

puts("请填写以下基本信息:");

gotoxy(63,13+2\*i);

printf("姓名:");

if (scanfPlus(x1->inputName,1,9,0) == 1)

return;

gotoxy(63,14+2\*i);

printf("电话:");

if (scanfPlus(x1->inputTel,11,11,3) == 1)

return;

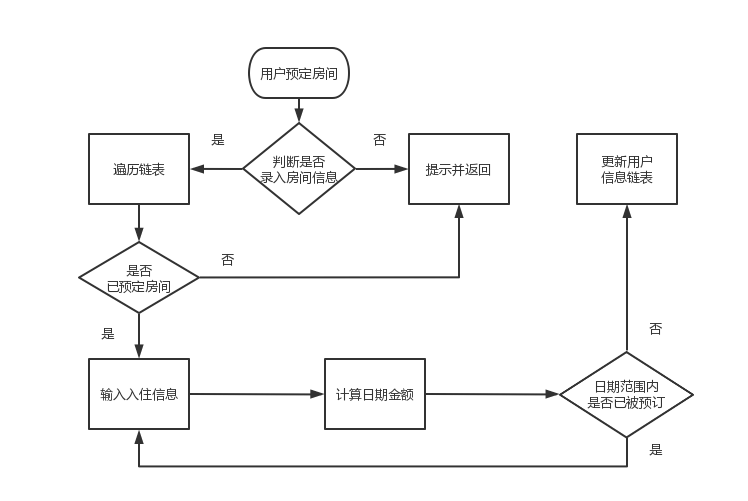
MessageBox(NULL,"订房成功!","提示", MB\_ICONASTERISK);

strcpy(x1->num,rNum);

write(head);

loginWrite(head1);

}



**3.用户退房间**

void out(struct room \*head,struct ID \*p,struct ID \*head1)

{

struct room \*p1,\*p2;

struct ID \*x1;

char judge[3];

int cost;

int s = time(0);

int d = s/86400 - 10957;

if (head == NULL)

{

MessageBox(NULL,"管理员还未录入任何房间信息!","提示", MB\_ICONASTERISK);

return;

}

x1 = head1;

while (x1 != NULL)

{

if (strcmp(x1->idnum,p->idnum) == 0)

break;

x1 = x1->next;

}

if (strcmp(x1->checkOut,"是") == 0)

{

MessageBox(NULL,"你已申请退房请等待管理员审核!","提示", MB\_ICONASTERISK);

return;

}

p1 = head;

while (p1 != NULL)

{

if (strcmp(p1->num,x1->num) == 0)

{

strcpy(p1->name,x1->inputName);

strcpy(p1->tel,x1->inputTel);

strcpy(p1->status,"已被预订");

p1->year = x1->year;

p1->mon = x1->mon;

p1->day = x1->day;

p1->days = x1->numDay;

A:printf("请核对你的房间信息是否正确\n");

form2(1);

form3(p1,0);

gotoxy(13,13);

s = time(0);

d = s/86400 - 10956;

cost = atoi(p1->cost);

x1->cost = (d - x1->daymin)\*cost;

if (x1->daymax < d)

printf("你已经超过了你的预定天数你的住宿费用为%d元",x1->cost);

else if (x1->daymax > d)

printf("你的住宿时间还未到入现在退房住宿费用为%d元",x1->cost);

else printf("你的住宿费用为%d元",x1->cost);

gotoxy(13,14);

if(MessageBox(NULL,"是否退房","询问",MB\_ICONASTERISK|MB\_OKCANCEL)!=IDCANCEL)

{

strcpy(x1->checkOut,"是");

MessageBox(NULL,"申请已提交请等待管理员审核!","提示", MB\_ICONASTERISK);

return;

}

MessageBox(NULL,"已取消退房申请!","提示", MB\_ICONASTERISK);

return;

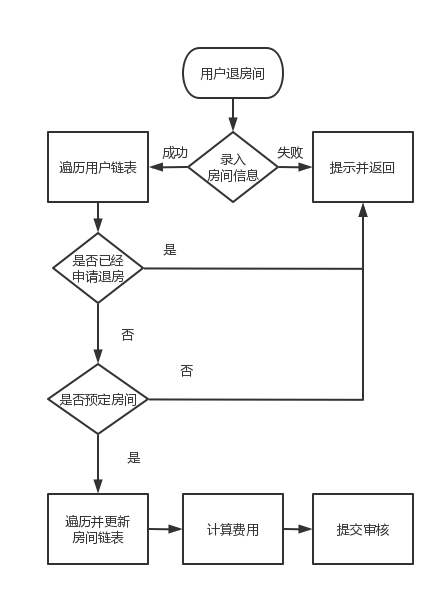
}

p1 = p1->next;

}

MessageBox(NULL,"你还未预定任何房间!","提示", MB\_ICONASTERISK);

}



**4.用户信息修改**

void userChange(struct room \*head,struct ID \*p,struct ID \*head1)

{

int x = 11,i = 0;

char t;

A:system("cls");

loginWrite(head1);

head1 = loginRead();

userChangeMenu();

while (1)

{

t = arrows(x,'0','3');

switch(t)

{

case 72:

if (i>0)

{

x = cColor(90,x,0);

if (x == 11)

puts("1.入住时间和天数");

else if (x == 13)

puts("2.入住客户姓名");

else

puts("3.电话");

i--;

}

else

inversion(&i,90,&x,3,17,"0.返回上一级界面");

break;

case 80:

if (i<3)

{

x = cColor(90,x,1);

if (x == 13)

puts("2.入住客户姓名");

else if (x == 15)

puts("3.电话");

else

puts("0.返回上一级界面");

i++;

}

else

inversion(&i,90,&x,0,11,"1.入住时间和天数");

break;

case 13: system("cls");

if (x == 11)

cInputTime(head,p,head1);

else if (x == 13)

cInputName(head,p,head1);

else if (x == 15)

cUserTel(head,p,head1);

else

return;

goto A;

case 49: system("cls");ShowCursor();cInputTime(head,p,head1); goto A;

case 50: system("cls");ShowCursor();cInputName(head,p,head1);goto A;

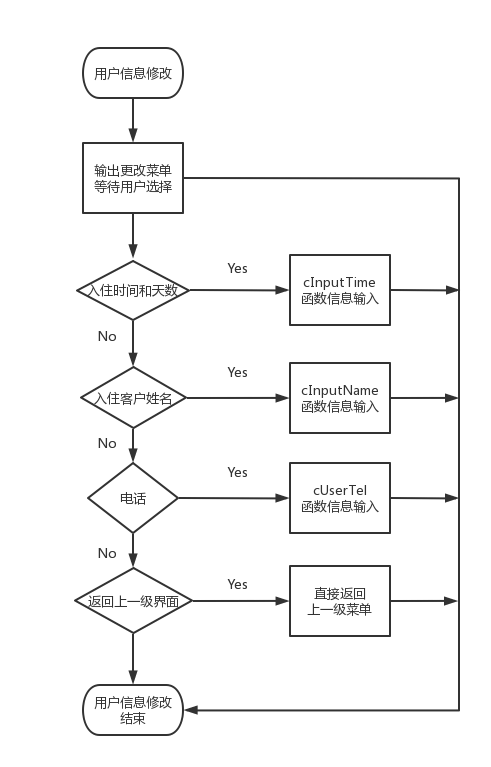
case 51: system("cls");ShowCursor();cUserTel(head,p,head1);goto A;

case 48: return;

}

}

}



**5.用户查看预定房间**

struct ID \*userAmend(struct room \*head,struct ID \*p,struct ID \*head1)

{

struct room \*p1;

struct ID \*x1;

int flag = 0;

x1 = head1;

while (x1 != NULL)

{

if (strcmp(x1->idnum,p->idnum) == 0)

break;

x1 = x1->next;

flag = 1;

}

if (flag == 1)

{

MessageBox(NULL,"你还未预定任何房间!","错误",MB\_ICONHAND);

return NULL;

}

p1 = head;

while (p1 != NULL)

{

if (strcmp(p1->num,x1->num) == 0)

{

status(p1,x1);

break;

}

p1 = p1->next;

}

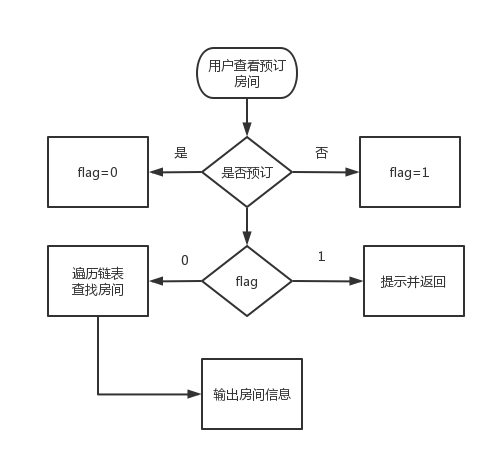
printf("你的订房信息如下");

form2(1);

form3(p1,0);

return x1;

}



**6.由日期计算天数**

**struct ID \*tranday(struct ID \*p)**

**{**

**int i;**

**int leap;**

**int times[2][13] =**

**{**

**{0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31},**

**{0, 31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31}**

**};**

**p->daymin = 0;**

**p->daymax = 0;**

**for (i = 2000; i < p->year; i++)**

**{**

**if (i % 4 == 0 && i % 100 != 0 || i % 400 == 0)**

**p->daymin += 366;**

**else**

**p->daymin += 365;**

**}**

**leap = p->year % 4 == 0 && p->year % 100 != 0 || p->year % 400 == 0;**

**for (i = 1; i < p->mon; i++)**

**p->daymin += times[leap][i];**

**p->daymin += p->day;**

**for (i = 2000; i < p->years; i++)**

**{**

**if (i % 4 == 0 && i % 100 != 0 || i % 400 == 0)**

**p->daymax += 366;**

**else**

**p->daymax += 365;**

**}**

**leap = p->years % 4 == 0 && p->years % 100 != 0 || p->years % 400 == 0;**

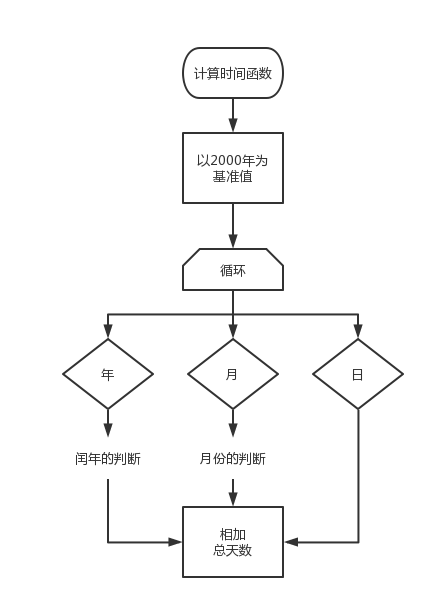
**for (i = 1; i < p->month; i++)**

**p->daymax += times[leap][i];**

**p->daymax += p->days;**

**return p;**

**}**

****

**7.房间信息覆盖**

void status(struct room \*p1,struct ID \*x1)

{

strcpy(p1->status,"已被预订");

strcpy(p1->name,x1->inputName);

strcpy(p1->tel,x1->inputTel);

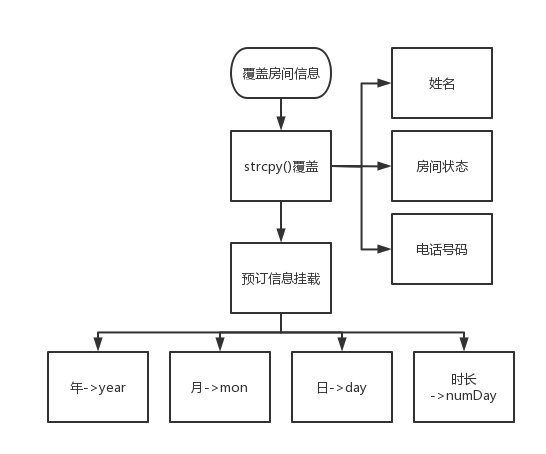
p1->year = x1->year;

p1->mon = x1->mon;

p1->day = x1->day;

p1->days = x1->numDay;

}



**8.重置房间信息**

**void reset(struct room \*head)**

**{**

**struct room \*p2;**

**p2 = head;**

**while (p2 != NULL)**

**{**

**strcpy(p2->status,"空闲");**

**strcpy(p2->name,"无");**

**strcpy(p2->tel,"无");**

**strcpy(p2->name,"无");**

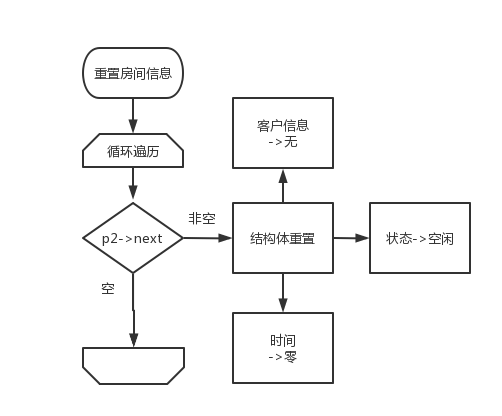
**p2->year = 1;**

**p2->days = 0;**

**p2 = p2->next;**

**}**

**}**

****

**9.提醒用户退房**

void remind(struct ID \*p,struct ID \*head1)

{

struct ID \*x1;

int s = time(0);

int d = s/86400 - 10956;

x1 = head1;

while (x1 != NULL)

{

if (strcmp(x1->idnum,p->idnum) == 0)

break;

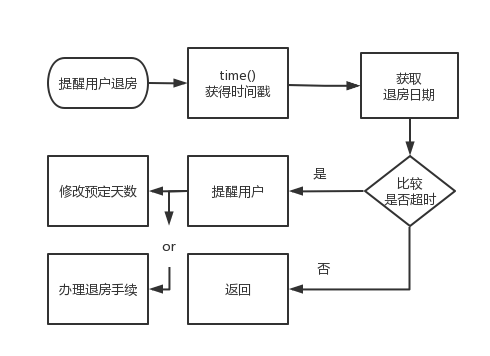
x1 = x1->next;

}

if (d > x1->daymax)

MessageBox(NULL,"你预定的退房时间到了请办理退房手续以免造成不必要的损失或者尝试修改预定天数!","提示",MB\_ICONASTERISK);

}



## 管理端主要函数与流程图

**1.管理端主函数**

int manage(struct room \*head,struct ID \*p,struct ID \*head1)

{

int x,i;

char t;

system("cls");

gotoxy(80,14);

if (strcmp(p->sex,"男") == 0)

printf("尊敬的%s管理员先生欢迎您登录酒店网上订房系统\n",p->name);

else

printf("尊敬的%s管理员女士欢迎您登录酒店网上订房系统\n",p->name);

gotoxy(80,15);

system("pause");

head = read();

A:system("cls");

write(head);

head = read();

head1 = loginRead();

menu(head1);

x = 8;

i = 0;

while (1)

{

t = arrows(x,'0','7');

switch(t)

{

case 72:

if (i > 0)

{

x = cColor(89,x,0);

if (x == 8)

puts("1.添加房间信息");

else if (x == 10)

puts("2.查看房间信息");

else if (x == 12)

puts("3.修改房间信息");

else if (x == 14)

puts("4.处理客户申请");

else if (x == 16)

puts("5.删除房间信息");

else if (x == 18)

puts("6.查找房间信息");

else

puts("7.退出登录");

i--;

}

else

inversion(&i,89,&x,7,22,"0.退出系统");

break;

case 80:

if (i < 7)

{

x = cColor(89,x,1);

if (x == 10)

puts("2.查看房间信息");

else if (x == 12)

puts("3.修改房间信息");

else if (x == 14)

puts("4.处理客户申请");

else if (x == 16)

puts("5.删除房间信息");

else if (x == 18)

puts("6.查找房间信息");

else if (x == 20)

puts("7.退出登录");

else

puts("0.退出系统");

i++;

}

else

inversion(&i,89,&x,0,8,"1.添加房间信息");

break;

case 13: system("cls");

if (x == 8)

head = add(head);

else if (x == 10)

look(head,head1);

else if (x == 12)

change(head);

else if (x == 14)

dispose(head);

else if (x == 16)

head = del(head,head1);

else if (x == 18)

finds(head,head1);

else if (x == 20)

return 0;

else

exit(0);

goto A;

case 49: system("cls");ShowCursor();head = add(head); goto A;

case 50: system("cls");ShowCursor();look(head,head1); goto A;

case 51: system("cls");ShowCursor();change(head); goto A;

case 52: system("cls");ShowCursor();dispose(head); goto A;

case 53: system("cls");ShowCursor();head = del(head,head1); goto A;

case 54: system("cls");ShowCursor();finds(head,head1);goto A;

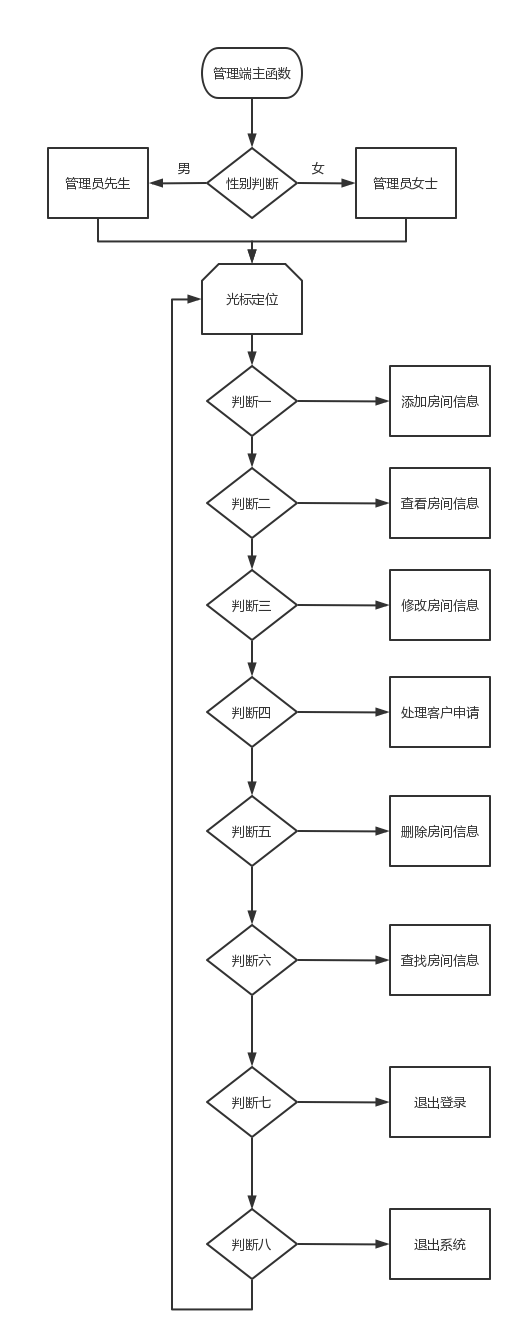
case 55: return 0;

case 48: exit(0);

}

}

}



**2.房间信息从小到大读取**

struct room \*read()

{

struct room \*head = NULL,\*p1,\*p2,\*p;

FILE \*fp;

if ((fp=fopen("房间信息","rb"))==NULL)

return head;

B:p=(struct room \*)malloc(sizeof(struct room)); //分配空间

fread(p,sizeof(struct room),1,fp); //将文件中的房间信息读取到链表中

if (feof(fp)) //判断是否读到文件结尾

{

fclose(fp);

return head;

}

if (head==NULL) //由于空链表的特殊性单独读取第一个房间信息

{

head=p;

p->next=NULL;

goto B;

}

p1=head;

if (p1->next==NULL) //判断链表之中是否只有一个头结点

{

if(strcmp(p->num,p1->num)<=0) //比较该房号与头结点房号的大小判断此房间信息写在头结点之前还是之后

{

head=p;

p->next=p1;

}

else

{

p1->next=p;

p->next=NULL;

}

goto B;

}

p2=p1;

while(p2!=NULL)

{

if (strcmp(p->num,p2->num)<0)

{

if (p2==head)

{

head=p;

p->next=p2;

goto B;

}

else

{

p1->next=p;

p->next=p2;

goto B;

}

}

else

{

p1=p2;

p2=p2->next;

}

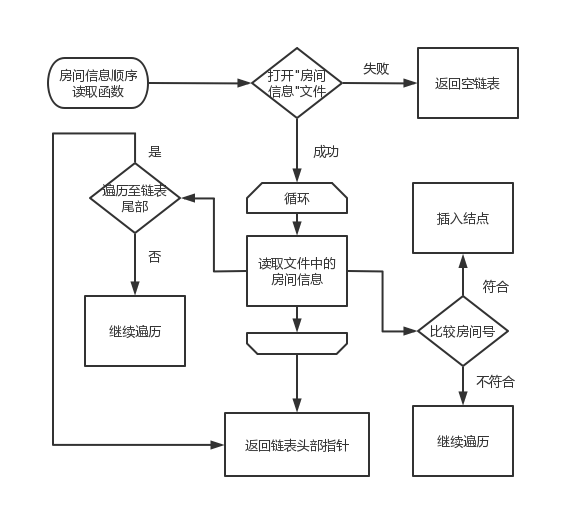
}

p1->next=p;

p->next=NULL;

goto B;

}



**3.房间信息写入**

void write(struct room \* head)

{

FILE \*fp;

struct room \*p1;

p1 = head;

fp = fopen("房间信息","wb");

while (p1 != NULL) //遍历链表中的信息写入文件中

{

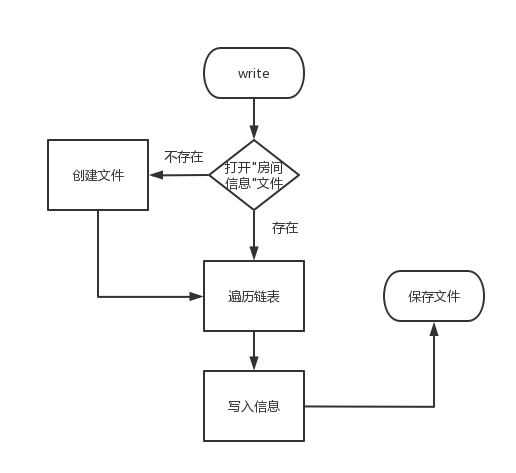
fwrite(p1,sizeof(struct room),1,fp);

p1 = p1->next;

}

fclose(fp);

}



**4.浏览房间信息**

void look(struct room \*head,struct ID \*head1)

{

int i,n = 0;

int s = time(0);

int d = s/86400 - 10956;

struct ID \*x1;

struct room \*p2;

if (head == NULL)

{

MessageBox(NULL,"无任何房间信息！","错误",MB\_ICONHAND);

return;

}

gotoxy(80,1);

printf("浏览房间信息");

p2 = head;

while (p2 != NULL)

{

x1 = head1;

while (x1 != NULL)

{

if (x1->daymin <= d && x1->daymax > d && strcmp(x1->num,p2->num) == 0)

{

strcpy(p2->status,"已被预订");//strcpy字符串复制

strcpy(p2->name,x1->inputName);

strcpy(p2->tel,x1->inputTel);

p2->year = x1->year;

p2->mon = x1->mon;

p2->day = x1->day;

p2->days = x1->numDay;

break;

}

x1 = x1->next;

}

n++;

p2 = p2->next;

}

form2(n);

p2 = head;

i = 0;

while (p2 != NULL)

{

form3(p2,i);

p2 = p2->next;

i++;

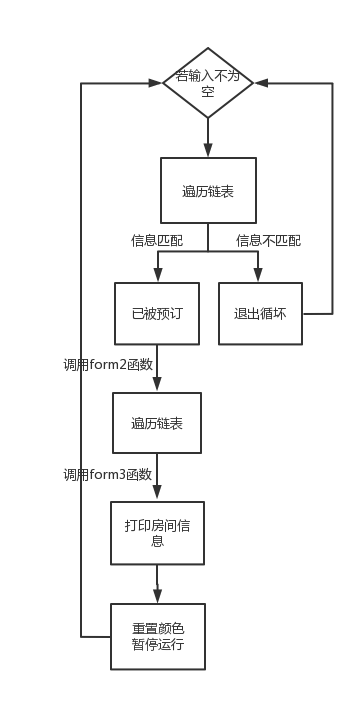
}

gotoxy(13,10+2\*i);

reset(head);

system("pause");

}



**5.修改房间信息**

void change(struct room \*head)

{

int x,i;

char t;

A:system("cls");

write(head);

x = 8;

i = 0;

cMenu();

while (1)

{

t = arrows(x,'0','5');

switch(t)

{

case 72:

if (i>0)

{

x = cColor(90,x,0);

if (x == 8)

puts("1.房间号");

else if (x == 10)

puts("2.房型");

else if (x == 12)

puts("3.位置");

else if (x == 14)

puts("4.入住");

else

puts("5.费用");

i--;

}

else

inversion(&i,90,&x,5,18,"0.返回上一级界面");

break;

case 80:

if (i<5)

{

x = cColor(90,x,1);

if (x == 10)

puts("2.房型");

else if (x == 12)

puts("3.位置");

else if (x == 14)

puts("4.入住");

else if (x == 16)

puts("5.费用");

else

puts("0.返回上一级界面");

i++;

}

else

inversion(&i,90,&x,0,8,"1.房间号");

break;

case 13: system("cls");

if (x == 8)

cNum(head);

else if (x == 10)

cType(head);

else if (x == 12)

cSite(head);

else if (x == 14)

cCheckIn(head);

else if (x == 16)

cCost(head);

else

return;

goto A;

case 49: system("cls");ShowCursor();cNum(head);goto A;

case 50: system("cls");ShowCursor();cType(head);goto A;

case 51: system("cls");ShowCursor();cSite(head);goto A;

case 52: system("cls");ShowCursor();cCheckIn(head);goto A;

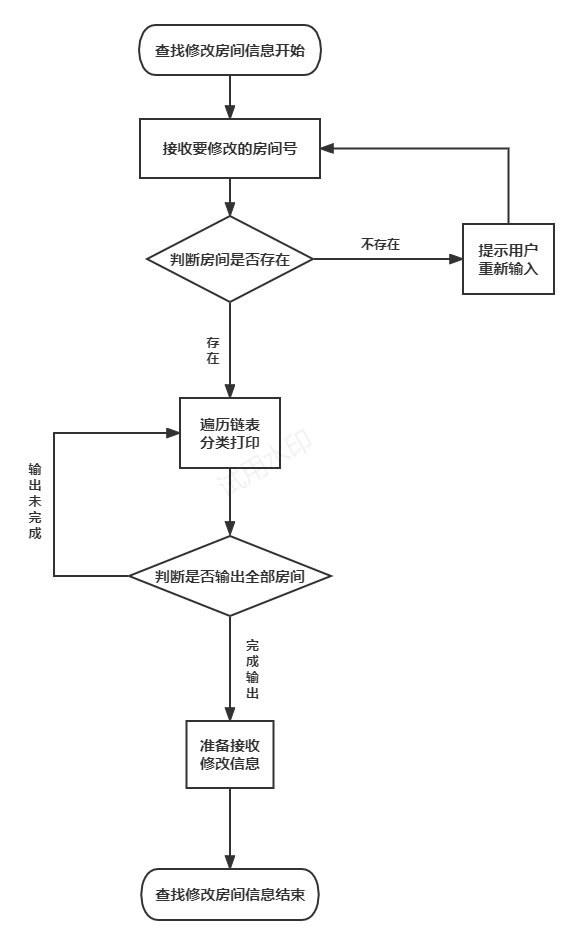
case 53: system("cls");ShowCursor();cCost(head);goto A;

case 48: return;

}

}

}



**6.删除房间信息**

struct room \*del(struct room \*head,struct ID \*head1)

{

struct room \*p2,\*p3;

struct ID \*x1;

char delnum[20];

if (head == NULL)

{

MessageBox(NULL,"无任何房间信息!","提示",MB\_ICONASTERISK);

return head;

}

printf("请输入要删除房间信息的房间号:");

if (scanfPlus(delnum,1,9,0) == 1)

return head;

x1 = head1;

while (x1 != NULL)

{

if (strcmp(x1->num,delnum) == 0)

{

MessageBox(NULL,"该房间有人预定不能删除","警告", MB\_ICONEXCLAMATION );

return head;

}

x1 = x1->next;

}

p2 = head;

while (p2 != NULL)

{

if (strcmp(delnum,p2->num) == 0)

{

A:system("cls");

form2(1);

form3(p2,0);

gotoxy(13,10);

if(MessageBox(NULL,"是否删除该房间信息","询问",MB\_ICONASTERISK|MB\_OKCANCEL)!=IDCANCEL)

{

if (p2 == head)

{

head = p2->next;

free(p2);

MessageBox(NULL,"删除成功!","提示",MB\_ICONASTERISK);

return head;

}

else

{

p3->next = p2->next;

free(p2);

MessageBox(NULL,"删除成功!","提示",MB\_ICONASTERISK);

return head;

}

}

MessageBox(NULL,"已取消删除！","提示",MB\_ICONASTERISK);

return head;

}

p3 = p2;

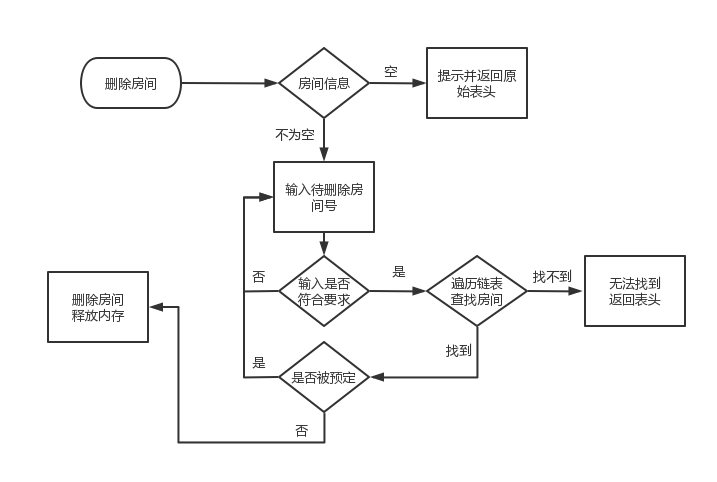
p2 = p2->next;

}

MessageBox(NULL,"查无此房间信息!","提示",MB\_ICONASTERISK);

return head;

}



7.**根据房间信息进行查找**

void finds(struct room \*head,struct ID \*head1)

{

int x,i;

char t;

A:system("cls");

x = 6;

i = 0;

fMenu();

while (1)

{

t = arrows(x,'0','9');

switch(t)

{

case 72:

if (i>0)

{

x = cColor(92,x,0);

if (x == 6)

puts("1.房间号");

else if (x == 8)

puts("2.房型");

else if (x == 10)

puts("3.位置");

else if (x == 12)

puts("4.房间状态");

else if (x == 14)

puts("5.入住时间");

else if (x == 16)

puts("6.天数");

else if (x == 18)

puts("7.费用");

else if (x == 20)

puts("8.入住客户姓名");

else if (x == 22)

puts("9.电话");

i--;

}

else

inversion(&i,92,&x,9,24,"0.返回上一级界面");

break;

case 80:

if (i<9)

{

x = cColor(92,x,1);

if (x == 8)

puts("2.房型");

else if (x == 10)

puts("3.位置");

else if (x == 12)

puts("4.房间状态");

else if (x == 14)

puts("5.入住时间");

else if (x == 16)

puts("6.天数");

else if (x == 18)

puts("7.费用");

else if (x == 20)

puts("8.入住客户姓名");

else if (x == 22)

puts("9.电话");

else

puts("0.返回上一级界面");

i++;

}

else

inversion(&i,92,&x,0,6,"1.房间号");

break;

case 13: system("cls");

if (x == 6)

fNum(head,head1);

else if (x == 8)

fType(head,head1);

else if (x == 10)

fSite(head,head1);

else if (x == 12)

fStatus(head,head1);

else if (x == 14)

fInputTime(head,head1);

else if (x == 16)

fDays(head,head1);

else if (x == 18)

fCost(head,head1);

else if (x == 20)

fName(head,head1);

else if (x == 22)

fTel(head,head1);

else

return;

goto A;

case 49: system("cls");ShowCursor();fNum(head,head1);goto A;

case 50: system("cls");ShowCursor();fType(head,head1);goto A;

case 51: system("cls");ShowCursor();fSite(head,head1);goto A;

case 52: system("cls");ShowCursor();fStatus(head,head1);goto A;

case 53: system("cls");ShowCursor();fInputTime(head,head1);goto A;

case 54: system("cls");ShowCursor();fDays(head,head1);goto A;

case 55: system("cls");ShowCursor();fCost(head,head1);goto A;

case 56: system("cls");ShowCursor();fName(head,head1);goto A;

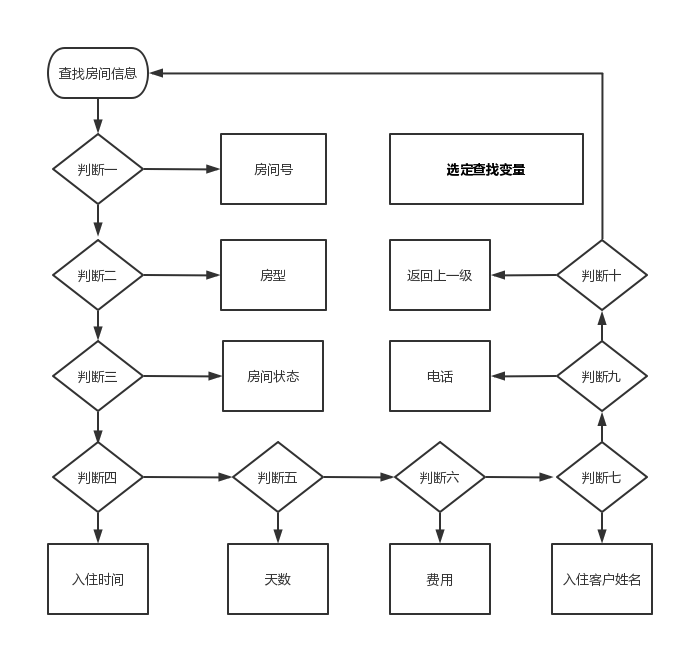
case 57: system("cls");ShowCursor();fTel(head,head1);goto A;

case 48: return;

}

}

}



# 系统测试

## 测试标准

#### 1、界面测试

**界面设计**：符合项目整体风格，色彩搭配协调，整体美感良好。

**布局排版**：页面元素布局整齐有序，不会产生错位或遮挡等问题。

**字体和字号**：字体大小适当、字体类型清晰易读，字号与内容相匹配。

**界面交互**：交互逻辑清晰，响应速度快，用户体验良好。

#### 2、功能测试

**功能覆盖率**：测试用例需全面覆盖所有功能点，测试用例应基于实际业务场景设计，覆盖所有可能出现的情况，并保证测试数据的真实性和有效性。

**功能正确性**：功能点需按照需求文档实现，不能出现错误或异常情况。

**数据有效性**：对输入数据进行有效性验证，确保数据格式正确并能被接受。

**数据完整性**：保证数据在存储、处理过程中不会丢失或损坏。

**功能安全性**：检查用户输入是否存在安全隐患，防止系统被恶意攻击。

#### 3、兼容性测试

**系统环境**：测试酒店管理系统在不同硬件和软件环境下的兼容性，包括操作系统、浏览器、数据库等。

**设备适配**：测试需覆盖PC、手机、平板等设备的兼容性。

**响应式布局**：确保页面在不同分辨率下能够正常显示。

**兼容性问题修复**：测试用例应涵盖多种硬件和软件组合，并重点测试常见的问题，如界面错位、字体显示异常等。及时解决出现的兼容性问题，确保系统能够稳定运行。

#### 4、可靠性测试

测试酒店管理系统在不同场景下的可靠性和稳定性，包括异常情况的处理、系统崩溃恢复等。测试用例应涵盖各种异常情况，如网络中断、数据库故障等，并验证系统的快速恢复能力。

#### 5、易用性测试

**操作简便性**：用户可以快速找到需要的功能，操作流程简单明了。

**可理解性**：用户能够正确理解各个功能的意义和作用。

**易学性**：新用户可以在短时间内掌握基本使用方法。

**反馈及时性**：系统反馈信息及时精准，用户能够获得良好的使用体验。

#### 6、性能测试

**加载速度**：测量界面加载速度，确保用户不会因为等待时间过长而流失。

**响应时间**：检查系统响应时间是否符合要求，避免用户操作时出现卡顿现象。

**负载性能：**测试用例应涵盖各种不同的负载情况，例如同时有多个用户使用系统，或者某个用户进行大量操作等。并记录系统在不同负载下的表现，并及时处理性能瓶颈。

**内存泄漏：**检查系统内存使用情况，防止出现内存泄漏问题。

#### 7、安全测试

**XSS攻击**：防止恶意脚本攻击，避免用户输入数据引发安全隐患。

**CSRF攻击**：防止跨站点请求伪造攻击，保护系统数据不被非法篡改。

**SQL注入攻击**：防止用户输入的恶意SQL语句对数据库进行攻击。

**权限控制：**确保用户访问权限符合角色所规定的范围，避免用户越权操作。

## 测试数据

**以下是用于测试的房间数据表格：**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **房间号** | **房型** | **位置** | **房间状态** | **入住** | **入住时间** | **天数** | **费用（天）** | **入住客户** | **电话** |
| **446** | **总统套房** | **东十一4楼** | **已被预 订** | **未入住** | **2023-05-30** | **5** | **1000** | **张三** | **13556546958** |
| **547** | **双人间** | **东十三5楼** | **已被预 订** | **未入住** | **2023-06-01** | **3** | **400** | **李四** | **18344623897** |
| **248** | **单人间** | **西九2楼** | **已被预 订** | **未入住** | **2023-05-28** | **10** | **150** | **王五** | **18645654913** |
| **1849** | **主题间** | **西一18楼** | **已被预 订** | **未入住** | **无** | **0** | **500** | **无** | **无** |
| **250** | **经济间** | **西十四2楼** | **已被预 订** | **未入住** | **无** | **0** | **99** | **无** | **无** |

**以下是用于测试的用户数据表格：**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **账户等级** | **账号** | **姓名** | **性别** | **身份证号** | **密码** | **密保** |
| **管理员** | **我一路向北** | **藤原拓海** | **男** | **441283200509075011** | **00000000** | **茂木夏树** |
| **普通用户** | **66666666** | **张三** | **男** | **44128320121207501x** | **00000000** | **唱跳rap** |
| **管理员** | **12345678** | **茂木夏树** | **女** | **441283199905189010** | **00000000** | **奔驰** |
| **普通用户** | **只因你太美** | **李四** | **女** | **44128320110107501x** | **00000000** | **1** |
| **普通用户** | **446泰裤辣** | **王五** | **男** | **441283201806014010** | **00000000** | **雪** |
| **普通用户** | **红尘客栈** | **pass** | **女** | **44128319981127801x** | **00000000** | **篮球** |

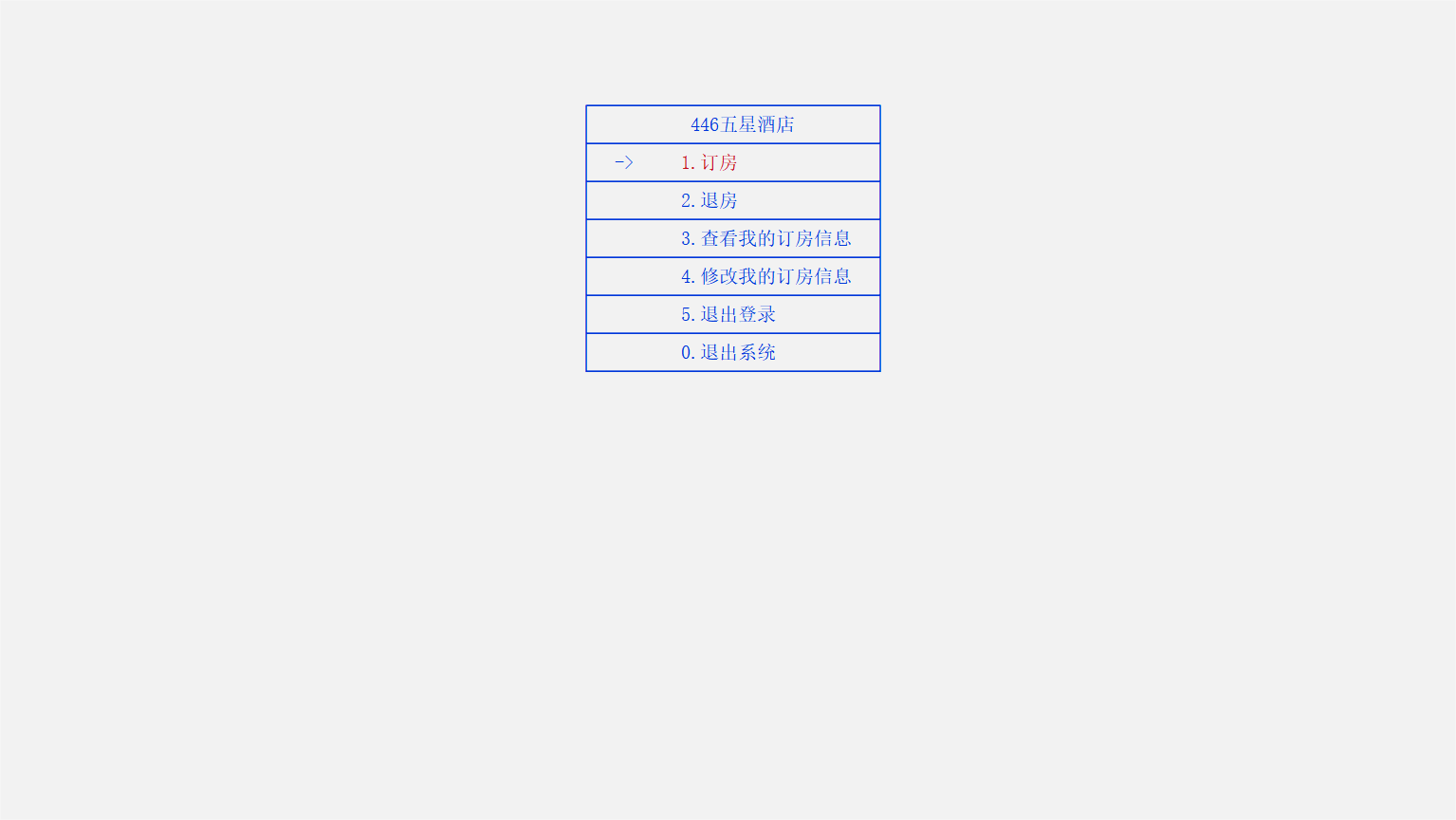
## 测试结果

**以下截图为测试的结果，基本满足测试标准：**

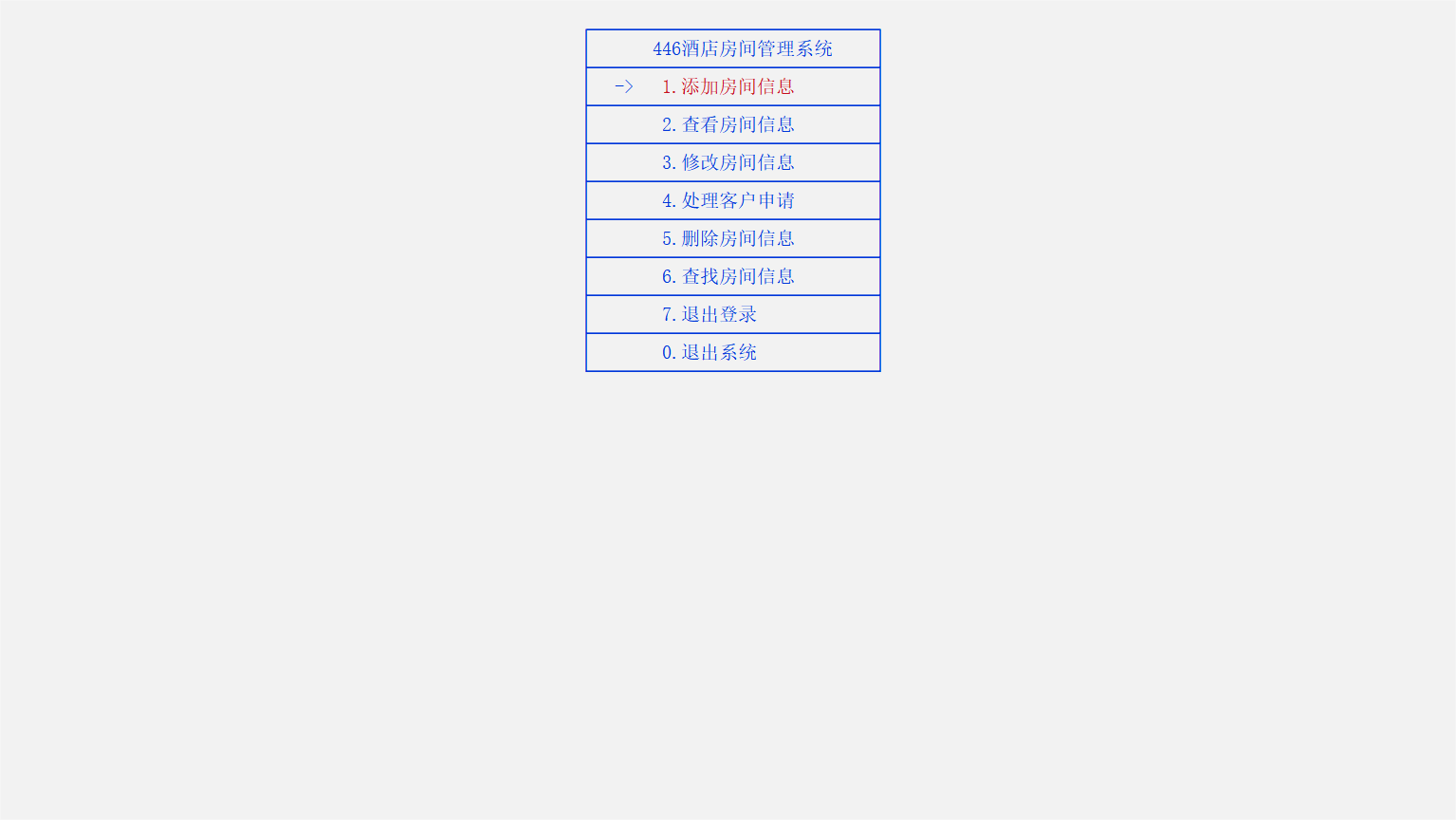
**1.登录面菜单**

****

**2.客户端菜单**

****

**3.管理端菜单**

****

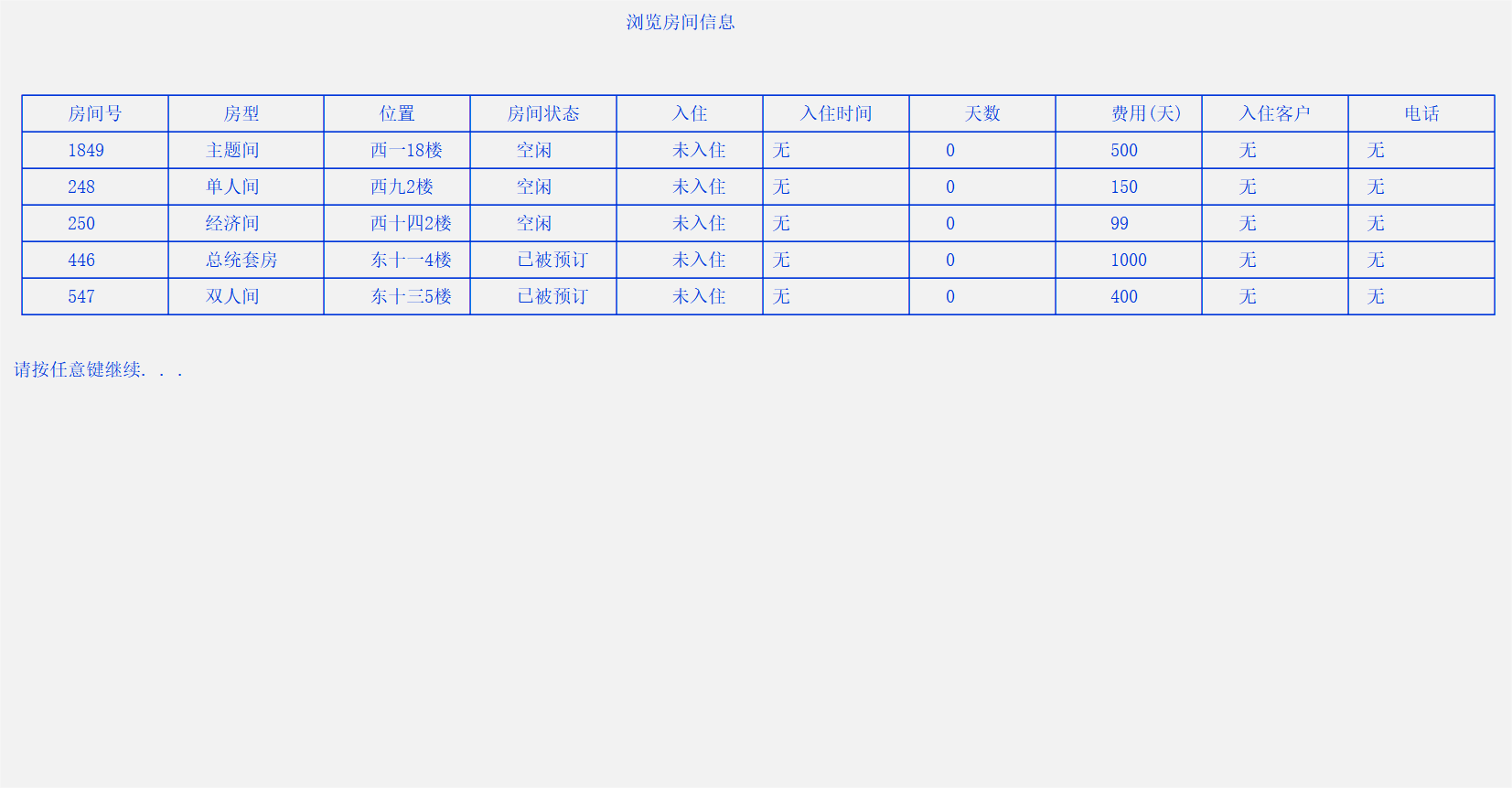
**4.游客端菜单**

****

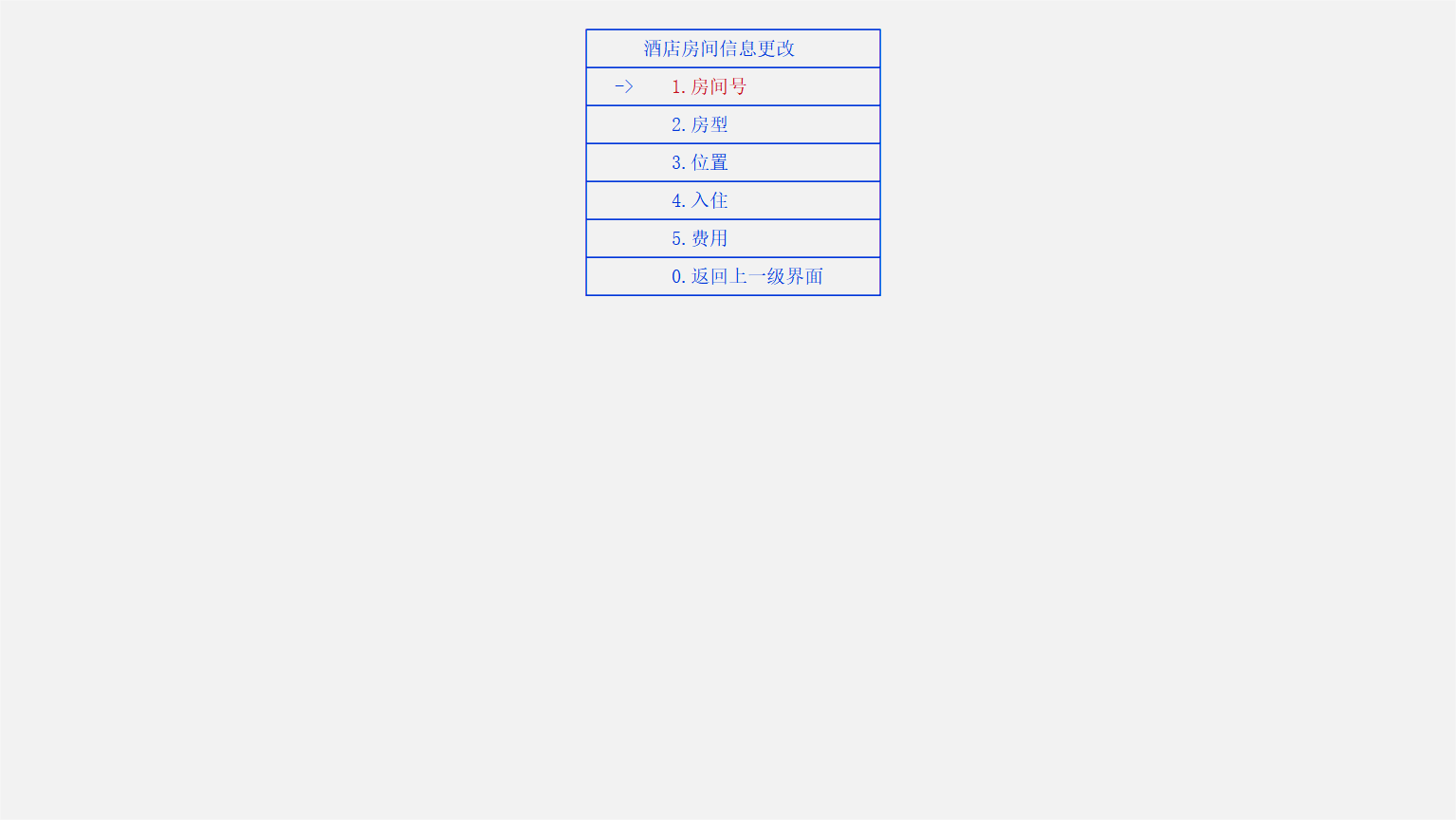
**5.房间查询菜单**

****

**6.浏览房间信息**

****

**7.房间信息更改菜单**

****

# 使用说明

**使 用 操 作**

1、主界面使用数字或上下方向键加enter进入esc键返回上一级

2、其余界面enter自动进行下一步

**管 理 员**

1、管理员登陆界面进入需先注册管理员账户，可在注销界面注销账户

2、登录后可进行对房间信息的添加、查看、修改、删除操作

3、房间信息分为房间号、房型、位置、入住情况、费用，管理员可进行编辑，查找可按不同信息查找

4、如若收到退房申请，需在处理客户申请处处理，通过申请则退房，不通过则需原因

**普 通 用 户**

1、普通用户登陆界面进入需先注册账户，可在注销界面注销账户

2、订房需填写住房日期与入住天数，再根据提供房间信息选择房间号，再填写姓名电话

3、如若需要修改订房信息，可通过修改界面修改

4、退房会向管理员发出申请，通过后可退房

**游 客**

游客只能查看酒店房间的部分信息，权限度较低

# 总结

## 心得与体会

小组在本科生C语言课程设计中选择了酒店管理系统作为题目，通过这个设计小组成员们深刻地理解了C语言的链表、指针等技术，并学会了如何灵活运用C语言函数进行编程。

在本次课程设计过程中，小组首先进行了需求分析和功能设计。通过对酒店管理系统的业务流程的深入了解，我们明确了该系统需要实现的功能模块，包括添加房间信息、查看房间信息、修改房间信息和处理客户申请等。接着，小组开始使用C语言进行编码。考虑到酒店管理系统需要对大量数据进行存储和处理，我们采用了链表技术来实现数据的存储和管理。同时，我们也利用指针技术提高了程序的运行效率和空间利用率。

在编写代码的过程中，小组遇到了无数个问题，但是通过查阅资料和不断调试，最终都得以解决。在这个过程中，成员都深刻体会到了编程的艰辛和乐趣。尤其是在实现复杂功能时，我们必须反复思考和优化代码，才能保证系统的稳定性和准确性。

此外，在本次课程设计中，我们还学会了如何使用C语言函数进行模块化编程。通过将不同功能的代码封装成函数，就可以在程序中更加方便地调用和复用这些代码，不仅提高了编程效率，也降低了代码维护的难度。

总的来说，本科生C语言课程设计是一次非常宝贵的学习经历。通过这个过程，我们不仅掌握了C语言的基础知识和编程技巧，还深入理解了软件开发的流程和方法。可以相信，在今后的学习和工作中，这些经验和技能都会派上用场。

**最后，想谈谈我们小组对一个优秀的系统需要具备的核心特点与开发原则：**

**功能齐全**

一个优秀的酒店房间管理系统必须要有完备的功能，包括房间信息管理、用户管理、增删与改查、统计与报表等模块。这些模块相互之间应该能够协同工作，使得整个系统更加高效。

**界面友好**

一个好的图书管理系统需要有良好的用户界面设计，让用户能够在使用时轻松快捷地完成操作。对于普通用户来说，简单易懂的界面可以提高用户的满意度。

**数据安全**

由于图书管理系统存储了大量的个人信息和敏感数据，因此必须要保障数据的安全性。系统应该采用合适的加密算法和安全机制，确保数据不被非法获取、篡改或泄露。

**定期迭代**

作为一个长期运营的软件产品，图书管理系统必须要不断更新迭代。我们需要不断根据用户反馈和市场需求进行优化，加入新的功能模块，提高系统效率。

**强调代码规范**

一个好的管理系统需要有清晰、简洁、可读性强的代码结构。为此，我们在开发过程中始终遵循了一些代码规范，如命名规范、注释规范等。

**不断学习**

作为开发者，我认为不断学习是非常重要的。我们需要时刻关注新技术、新思路，不断学习并尝试应用到实践中。这样才能保持我们在开发中的竞争力。

## 优点与不足

## **优 点**

本程序自开发以来经过不断地修改与维护，无论是在代码还是在功能上体现出了优秀的健壮性。开发小组内部的协商与合作更是成为了推进项目快速落地、问题及时解决的关键。总之，本次项目经历的小组优势无论在程序处理上还是在人际处理上都体现得淋漓尽致，为我们自己点赞。

**不 足**

本程序的设计与实现中仍存在与实际应用脱轨的问题，无法真正的模拟现实的使用场景，在数据的算法采用上也出现了影响性能的痛点，而且没有结合前后端与数据库对数据进行储存与管理等，都是我们需要在迭代中优化并完善的问题。

## 优化与设想

本程序虽然已经成功通过C语言实现，但是仍然存在许多有待改进的地方，以下是对酒店房间管理系统优化方向的设想：

**数据库设计优化：**一般来说，酒店管理系统需要处理大量的数据，如订房信息、客户信息等。因此，在设计数据库时，需要合理地选择数据类型、建立索引等，以提高系统的查询和处理速度。

**接口设计优化：**接口是程序和外部环境之间的桥梁，一个好的接口设计能够提高系统的易用性和稳定性。在酒店管理系统中，我们可以通过优化用户界面、增加快捷键、添加错误提示等方式来改进接口设计。

**内存管理优化：**内存泄漏是C语言程序最常见的问题之一。为了避免内存泄漏和其他内存相关的问题，我们可以使用动态内存分配函数（如malloc和free）并合理地管理内存资源。同时，酒店管理系统需要存储大量的数据，因此我们可以采用压缩、索引等方式来优化存储空间，减少存储成本。

**多线程优化：**多线程是提高程序并发性的重要手段，能够有效地提高酒店管理系统的响应速度。我们可以将不同功能的代码放到不同的线程中运行，从而避免因某个操作耗时而导致整个系统的卡顿。

**安全性优化：**酒店管理系统涉及到客户的敏感信息，如个人身份证号码、银行卡号等，因此安全性是非常重要的。我们可以采取加密、防抄袭等措施来保障系统的安全性。

**错误处理：**在酒店管理系统中，出现错误是很常见的。为了让用户更加友好地使用系统，我们可以增加错误处理机制，如错误提示、日志记录等。

**功能扩展：**在系统开发完成后，我们应该持续的对系统进行功能扩展以适应不同的业务需求。例如，添加房间清洁状态、打印订单等新功能。

**代码复用：**酒店管理系统中有很多相似的代码块，我们可以将这些代码块封装成函数并重复利用，从而提高代码重用率和开发效率。

**前端设计优化：**前端界面是用户与系统交互的主要方式。我们可以通过优化前端设计、增加动态效果等方式来提高用户体验，从而提升客户满意度。

**自动化测试：**随着系统的不断升级和迭代，测试工作也变得越来越繁琐。我们可以采用自动化测试工具，并编写自动化测试脚本，以提高测试效率和测试覆盖率。

**性能监控：**对于酒店管理系统而言，性能是至关重要的。在对酒店管理系统进行优化前，我们需要先进行性能测试，以了解系统的瓶颈和性能瓶颈所在。通过性能测试，可以找出影响系统性能的因素，然后可以通过引入性能监控工具，实时监控系统运行情况，并及时发现和解决性能瓶颈问题。

**致 谢**

本次C语言课程设计论文是我们小组经过了一个多月的摸索与尝试后完成的。我们小组成员间互相取长补短，紧密团结在以组长蒋卓然为核心的领导周围，他严肃的科学态度，严谨的治学精神，精益求精的工作作风，深深感染和激励着小组成员，是小组能够碰到bug却不放弃，代码不懂依然坚持的不竭动力源泉。借此机会谨向组长蒋卓然致以诚挚的谢意和崇高的敬意。

在此，还要感谢一齐拼搏奋斗了一年多的446宿舍全体同志，正是由于互相的帮忙和理解，我们在以宿舍为单位的小组才能克服一个又一个的困难和疑惑，直至本次C语言课设顺利完成。

在论文即将完成之际，激动之情荡漾在我们每个小组成员的心头，从开始进入课题到论文的顺利完成，有多少可敬的师长、同学、朋友给予了我们无私的帮助，在此再次向所有给我们提供帮助的同仁致以最诚挚的谢意!

最后的最后，由衷感谢亲切的空调和畅快的校园网，是它们的辛勤工作给我们的论文完成保驾护航！祝愿身边遇到的每一位：前程似锦，万事胜意，一生纯善坦荡，不坠青云之志!

**参 考 文 献**

[1]史宇轩,徐洪丽,丁颐宁.C语言指针变量研究[J].科技资讯,2023,21(08):1-4.

[2]李丽萍,教巍巍.C语言程序设计中链表的教学方法[J].辽宁工业大学学报(社会科学版),2021,23(05):121-123.

[3]夏芸.《C语言程序设计》中结构体实验教学的探讨[J].电脑知识与技术,2015,11(09):163-164.

[4]谭浩强.2017.《C程序设计（第五版）》[J].北京：清华大学出版社.370pp.

[5]姜青山,洪心兰.C语言程序设计的健壮性与安全性研究[J].工矿自动化,2007:128-129.

**附 录**

#include"stdio.h"

#include"conio.h"

#include"stdlib.h"

#include"string.h"

#include"time.h"

#include"windows.h"

#define LEN 10

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*函数声明区\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void gotoxy(int x, int y); //光标定位函数声明

int manage(struct room \*head,struct ID \*p,struct ID \*head1); //管理员端主函数声明

void menu(struct ID \*head1); //管理员端主菜单函数声明

struct room \*read(); //将文件中的房间信息读取到链表中函数声明

void write(struct room \* head); //将文件中的房间信息写进链表中函数声明

struct room \*add(struct room \*head); //增添房间信息函数声明

void look(struct room \*head,struct ID \*head1); //浏览房间信息函数声明

void change(struct room \*head); //修改房间信息主函数声明

void cMenu(); //修改房间信息菜单函数声明

struct room \*amend(struct room \*head); //修改遍历函数声明

void cNum(struct room \*head); //修改房间号函数声明

void cType(struct room \*head); //修改房型函数声明

void cSite(struct room \*head); //修改房间位置函数声明

void cCheckIn(struct room \*head); //修改房间入住信息函数声明

void cCost(struct room \*head); //修改房间价格函数声明

void cName(struct room \*head); //修改住客姓名函数声明

void dispose(struct room \*head); //处理住客申请函数声明

struct room \*del(struct room \*head,struct ID \*head1); //删除房间信息函数声明

void finds(struct room \*head,struct ID \*head1); //查询房间信息函数声明

void fMenu(); //查询主菜单函数声明

void fNum(struct room \*head,struct ID \*head1); //按身份证号码进行查询函数声明

void fType(struct room \*head,struct ID \*head1); //按房型进行查询函数声明

void fSite(struct room \*head,struct ID \*head1); //按位置查询函数声明

void fStatus(struct room \*head,struct ID \*head1); //按状态查询函数声明

void fInputTime(struct room \*head,struct ID \*head1); //按加入时间查询函数声明

void fDays(struct room \*head,struct ID \*head1); //按加入天数查询函数声明

void fCost(struct room \*head,struct ID \*head1); //按每天费用查询函数声明

void fName(struct room \*head,struct ID \*head1); //按姓名查询函数声明

void fTel(struct room \*head,struct ID \*head1); //按电话号码查询函数声明

void signFind(struct ID \*head); //密码找回函数声明

struct ID \*logout(struct ID \*head); //密码注销函数声明

struct ID \*accountName(struct ID \*head); //用户账号注册函数声明

struct ID \*sign(struct ID \*head); //用户登录函数声明

struct ID \*loginRead(); //用户数据读入函数声明

void loginWrite(struct ID \*head); //将用户数据写入文件中函数声明

void loginMenu(); //登录界面菜单函数声明

struct ID \*login(); //用户登录系统主函数

void changePassword(struct ID \*head); //密码修改函数声明

char \*randStr(char str[], int n); //验证码生成函数声明

int user(struct room \*head,struct ID \*p,struct ID \*head1); //用户端主函数声明

void umenu(struct ID \*p); //用户端主菜单函数声明

void reserve(struct room \*head,struct ID \*p,struct ID \*head1); //用户端预定酒店函数声明

void out(struct room \*head,struct ID \*p,struct ID \*head1); //用户端退酒店函数声明

void userLook(struct room \*head,struct ID \*p,struct ID \*head1); //用户房间查看函数声明

void userChange(struct room \*head,struct ID \*p,struct ID \*head1); //用户房间修改函数声明

void userChangeMenu(); //用户房间修改菜单函数声明

struct ID \*userAmend(struct room \*head,struct ID \*p,struct ID \*head1); //用户查看预定房间函数声明

void cInputTime(struct room \*head,struct ID \*p,struct ID \*head1); //用户修改预定时间函数声明

void cInputName(struct room \*head,struct ID \*p,struct ID \*head1); //用户修改预定姓名函数声明

void cUserTel(struct room \*head,struct ID \*p,struct ID \*head1); //用户修改预定电话函数声明

int scanfPlus(char \*st,int m,int n,int x); //改进版输入函数声明

void visitor(); //游客端函数声明

void viewMenu(); //游客端菜单函数声明

void viewLook(); //游客端查看函数声明

void load(); //进度条加载函数声明

void HideCursor(); //隐藏光标函数声明

void form(int x,int n); //菜单表格函数声明

void form2(int n); //房间完整信息表格函数声明

void form3(struct room \*p1,int i); //房间完整信息表格中的数据函数声明

int form4(int n,struct room \*head); //房间部分信息表格中的数据函数声明

struct ID \*tranday(struct ID \*p); //天数转化函数声明

void status(struct room \*p1,struct ID \*x1); //状态更改函数声明

BOOL SetConsoleColor(WORD wAttributes); //设置字符的前景色和后景色函数声明

void reset(struct room \*head); //重置信息函数声明

int arrows(int x,char min,char max); //方向键指向函数声明

void blue(); //蓝色字体函数声明

void red(); //红色字体函数声明

int cColor(int y,int x,int judge); //箭头移动信息同步变化函数声明

void inversion(int \*i,int y,int \*x,int a,int b,char \*s); //箭头方向倒置函数声明

void remind(struct ID \*p,struct ID \*head1); //提醒函数声明

void ShowCursor(); //显示关标函数声明

/\*\*\*\*\*\*\*\*\*\*\*创建房间信息的链表\*\*\*\*\*\*\*\*\*\*\*/

struct room

{

char num[10]; //房间号

char type[11]; //房型

char site[10]; //位置

char status[10]; //状态

char check[10]; //入住

char checkOut[10]; //退房

char inputTime[14]; //加入时间

int year; //加入的年

int mon; //月

int day; //日

int days; //天数

char cost[10]; //费用

char name[10]; //姓名

char tel[12]; //电话

char idnum[20]; //身份证号

struct room \*next; //下一个结点地址

};

/\*\*\*\*\*\*\*\*\*\*\*\*创建用户信息的链表\*\*\*\*\*\*\*\*\*\*/

struct ID

{

char power[10]; //权限(管理员or普通用户)

char account[20]; //账号

char key[20]; //密码

char keyProtect[20]; //密保

char check[10]; //入住

char checkOut[10]; //退房

char name[10]; //姓名

char idnum[20]; //身份证号

char num[10]; //房间号

char sex[3]; //性别

char inputName[10]; //入住时间

char inputTel[12]; //入住电话

int year; //入住时的年

int mon; //入住时的月

int day; //入住时的日

int years; //离开时的年

int month; //离开时的月

int days; //离开时的日

int numDay; //预定天数

int daymax; //离开日期的天数

int daymin; //入住日期的天数

int cost; //房间每日的价格

char information[100]; //信息

struct ID \* next; //下一个结点地址

};

/\*\*\*\*\*\*\*\*\*\*\*\*主函数\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int main(void)

{

system("color F1");

// load();

struct ID \*p,\*head1;

struct room \*head = NULL;

gotoxy(80,14);

head = read(); //房间信息读入链表中

head1 = loginRead();

p = login(); //登录后返回该用户信息块

while(1)

{

if (strcmp(p->power,"管理员") == 0) //判断用户身份并选择进入相应端口

{

if(manage(head,p,head1) == 0) //判断用户否是退出登录

p = login();

}

else

if(user(head,p,head1) == 0)

p = login();

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*设置字符的前景色和背景色\*\*\*\*\*\*\*\*\*\*/

BOOL SetConsoleColor(WORD wAttributes)

{

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

if (hConsole == INVALID\_HANDLE\_VALUE)

return FALSE;

return SetConsoleTextAttribute(hConsole,wAttributes);

}

/\*\*\*\*\*\*\*\*\*\*光标定位\*\*\*\*\*\*\*\*\*\*/

void gotoxy(int x, int y)

{

COORD coord = {x,y}; //在控制台屏幕上的坐标

SetConsoleCursorPosition(GetStdHandle(STD\_OUTPUT\_HANDLE), coord); //获取句柄(光标停留位置的信息)

}

/\*这是一个在控制台屏幕上移动光标位置的函数。它接受两个参数，x和y，代表要移动到的行列坐标。首先将这些坐标存储在一个COORD结构中，然后使用Windows API函数SetConsoleCursorPosition()将光标位置设置为这些坐标。GetStdHandle(STD\_OUTPUT\_HANDLE)函数用于获取控制台输出缓冲区的句柄，从而能够操作控制台屏幕上的光标位置。通过调用此函数可以在控制台窗口上输出指定位置的字符或字符串。\*/

/\*\*\*\*\*管理端主函数\*\*\*\*\*\*/

int manage(struct room \*head,struct ID \*p,struct ID \*head1)

{

int x,i;

char t;

system("cls");

gotoxy(80,14);

if (strcmp(p->sex,"男") == 0)

printf("尊敬的%s管理员先生欢迎您登录酒店网上订房系统\n",p->name);

else

printf("尊敬的%s管理员女士欢迎您登录酒店网上订房系统\n",p->name);

gotoxy(80,15);

system("pause");

head = read();

A:system("cls");

write(head);

head = read();

head1 = loginRead();

menu(head1);

x = 8;

i = 0;

while (1)

{

t = arrows(x,'0','7');

switch(t)

{

case 72:

if (i > 0)

{

x = cColor(89,x,0);

if (x == 8)

puts("1.添加房间信息");

else if (x == 10)

puts("2.查看房间信息");

else if (x == 12)

puts("3.修改房间信息");

else if (x == 14)

puts("4.处理客户申请");

else if (x == 16)

puts("5.删除房间信息");

else if (x == 18)

puts("6.查找房间信息");

else

puts("7.退出登录");

i--;

}

else

inversion(&i,89,&x,7,22,"0.退出系统");

break;

case 80:

if (i < 7)

{

x = cColor(89,x,1);

if (x == 10)

puts("2.查看房间信息");

else if (x == 12)

puts("3.修改房间信息");

else if (x == 14)

puts("4.处理客户申请");

else if (x == 16)

puts("5.删除房间信息");

else if (x == 18)

puts("6.查找房间信息");

else if (x == 20)

puts("7.退出登录");

else

puts("0.退出系统");

i++;

}

else

inversion(&i,89,&x,0,8,"1.添加房间信息");

break;

case 13: system("cls");

if (x == 8)

head = add(head);

else if (x == 10)

look(head,head1);

else if (x == 12)

change(head);

else if (x == 14)

dispose(head);

else if (x == 16)

head = del(head,head1);

else if (x == 18)

finds(head,head1);

else if (x == 20)

return 0;

else

exit(0);

goto A;

case 49: system("cls");ShowCursor();head = add(head); goto A;

case 50: system("cls");ShowCursor();look(head,head1); goto A;

case 51: system("cls");ShowCursor();change(head); goto A;

case 52: system("cls");ShowCursor();dispose(head); goto A;

case 53: system("cls");ShowCursor();head = del(head,head1); goto A;

case 54: system("cls");ShowCursor();finds(head,head1);goto A;

case 55: return 0;

case 48: exit(0);

}

}

}

/\*此函数主要实现了一个酒店管理系统的功能。其中包括添加房间信息、查看房间信息、修改房间信息、处理客户申请、删除房间信息以及查找房间信息等功能。同时还有管理员的登录和退出系统等操作。该系统可以通过终端界面进行操作，具有较好的交互性和用户体验。\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*管理端主菜单\*\*\*\*\*\*\*\*\*\*\*\*/

void menu(struct ID \*head1)

{

struct ID \*x1;

int i = 0;

x1 = head1;

while (x1 != NULL) //遍历链表检索申请

{

if (strcmp(x1->checkOut,"是") == 0)

++i;

x1 = x1->next;

}

if (i>0)

printf("你有%d个待处理的退房申请\n\n\a",i);

//system("pause");

form(5,8);

gotoxy(88,6);

puts("446酒店房间管理系统");

gotoxy(89,10);

puts("2.查看房间信息");

gotoxy(89,12);

puts("3.修改房间信息");

gotoxy(89,14);

puts("4.处理客户申请");

gotoxy(89,16);

puts("5.删除房间信息");

gotoxy(89,18);

puts("6.查找房间信息");

gotoxy(89,20);

puts("7.退出登录");

gotoxy(89,22);

puts("0.退出系统");

red();

gotoxy(89,8);

puts("1.添加房间信息");

}

/\*此函数用于显示一个酒店房间管理系统的菜单。该函数接受一个名为head1的ID类型结构体指针作为参数。

这个函数首先遍历链表以检索待处理的退房申请，并计算出待处理的申请数量i。然后在屏幕上打印出菜单选项，包括添加、查看、修改、处理客户申请、删除、查找房间信息、退出登录和退出系统等选项。

此外，在菜单显示时，函数还调用了form函数和gotoxy函数，用于绘制菜单框架和定位文本位置。在菜单中，“446酒店房间管理系统”使用了红色字体突出显示。\*/

/\*\*\*\*\*\*\*\*\*\*\*\*房间信息从小到大读取\*\*\*\*\*\*\*\*\*/

struct room \*read()

{

struct room \*head = NULL,\*p1,\*p2,\*p;

FILE \*fp;

if ((fp=fopen("房间信息","rb"))==NULL)

return head;

B:p=(struct room \*)malloc(sizeof(struct room)); //分配空间

fread(p,sizeof(struct room),1,fp); //将文件中的房间信息读取到链表中

if (feof(fp)) //判断是否读到文件结尾

{

fclose(fp);

return head;

}

if (head==NULL) //由于空链表的特殊性单独读取第一个房间信息

{

head=p;

p->next=NULL;

goto B;

}

p1=head;

if (p1->next==NULL) //判断链表之中是否只有一个头结点

{

if (strcmp(p->num,p1->num)<=0) //比较该房号与头结点房号的大小判断此房间信息写在头结点之前还是之后

{

head=p;

p->next=p1;

}

else

{

p1->next=p;

p->next=NULL;

}

goto B;

}

p2=p1;

while(p2!=NULL)

{

if (strcmp(p->num,p2->num)<0)

{

if (p2==head)

{

head=p;

p->next=p2;

goto B;

}

else

{

p1->next=p;

p->next=p2;

goto B;

}

}

else

{

p1=p2;

p2=p2->next;

}

}

p1->next=p;

p->next=NULL;

goto B;

}

/\*这段代码是一个用于读取二进制文件中房间信息的函数，它返回一个指向链表头部的指针。该链表中的结点按照房间号从小到大排列。

函数首先打开名为"房间信息"的文件，如果无法打开文件，则返回空链表。然后，函数通过循环读取文件中的房间信息，并将这些信息插入到链表中，以确保链表中结点的有序性。

具体而言，函数使用了三个指针：p1、p2和p。其中，p1指向当前节点的前一个节点，p2指向当前节点的后一个节点，p指向新读取出来的节点。在循环过程中，指针p1从链表头开始遍历所有节点，直到找到一个节点的房间号比新读取出来的节点的房间号大，或者遍历到链表末尾。如果找到了这样一个节点，则将新节点插入到p1和p2之间，否则将新节点插入到链表末尾。最后，函数返回链表头部的指针。

需要注意的是，此代码使用了goto语句，这种方式不太常见，但是在一些特定的条件下可以简化程序的控制流程。\*/

/\*\*\*\*\*\*\*\*\*\*\*房间信息写入\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void write(struct room \* head)

{

FILE \*fp;

struct room \*p1;

p1 = head;

fp = fopen("房间信息","wb");

while (p1 != NULL) //遍历链表中的信息写入文件中

{

fwrite(p1,sizeof(struct room),1,fp);

p1 = p1->next;

}

fclose(fp);

}

/\*此函数用于将链表中的房间信息写入文件中。该函数接受一个名为head的room类型结构体指针作为参数。

在该函数中，首先定义了一个名为fp的文件指针，并以二进制写方式打开名为“房间信息”的文件，如果该文件不存在则创建它。

然后定义了一个名为p1的room类型结构体指针，初始化为head。接下来，在while循环中，遍历链表，将每个房间信息以二进制形式写入文件中。fwrite函数用于写入数据，其参数依次为待写入数据的地址、数据结构体的大小、待写入的数量和文件指针。最后关闭文件。

该函数主要用于将链表中的房间信息保存到文件中，方便下次程序运行时使用。在程序中，可以通过调用此函数来实现将当前房间信息保存下来的功能。\*/

/\*\*\*\*\*\*\*\*浏览房间信息\*\*\*\*\*\*\*\*/

void look(struct room \*head,struct ID \*head1)

{

int i,n = 0;

int s = time(0);

int d = s/86400 - 10956;

struct ID \*x1;

struct room \*p2;

if (head == NULL)

{

MessageBox(NULL,"无任何房间信息！","错误",MB\_ICONHAND);

return;

}

gotoxy(80,1);

printf("浏览房间信息");

p2 = head;

while (p2 != NULL)

{

x1 = head1;

while (x1 != NULL)

{

if (x1->daymin <= d && x1->daymax > d && strcmp(x1->num,p2->num) == 0)

{

strcpy(p2->status,"已被预订");//strcpy字符串复制

strcpy(p2->name,x1->inputName);

strcpy(p2->tel,x1->inputTel);

p2->year = x1->year;

p2->mon = x1->mon;

p2->day = x1->day;

p2->days = x1->numDay;

break;

}

x1 = x1->next;

}

n++;

p2 = p2->next;

}

form2(n);

p2 = head;

i = 0;

while (p2 != NULL)

{

form3(p2,i);

p2 = p2->next;

i++;

}

gotoxy(13,10+2\*i);

reset(head);

system("pause");

}

/\*此函数用于浏览房间信息。该函数接受两个参数：一个是名为head的room类型结构体指针，表示链表中存储的房间信息；另一个是名为head1的ID类型结构体指针，表示链表中存储的客户预订信息。

在该函数中，首先定义了一些变量，如i、n、s和d等。其中，s表示当前时间距离1970年1月1日之间的秒数，d表示从1970年1月1日到现在的天数。

然后定义了两个结构体指针x1和p2，并初始化为head1和head。接下来，在while循环中遍历链表中的每个房间信息。在内部循环中，遍历链表中的每个客户预订信息，如果找到一个客户预订信息与当前房间信息匹配，则将该房间状态更改为“已被预订”，并将客户姓名、电话、入住日期、预定天数等信息更新到房间信息中。最后退出内层循环。

接下来，根据房间信息个数n，调用form2函数绘制相应的窗口框架，然后再次遍历链表中的每个房间信息，调用form3函数在窗口中打印出该房间的相关信息。最后重置控制台的颜色，并暂停程序的运行，等待用户输入。该函数主要用于浏览房间信息和客户预订信息，并将它们显示在屏幕上，以便用户查看。\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*修改房间信息\*\*\*\*\*\*\*\*\*\*\*\*/

void change(struct room \*head)

{

int x,i;

char t;

A:system("cls");

write(head);

x = 8;

i = 0;

cMenu();

while (1)

{

t = arrows(x,'0','5');

switch(t)

{

case 72:

if (i>0)

{

x = cColor(90,x,0);

if (x == 8)

puts("1.房间号");

else if (x == 10)

puts("2.房型");

else if (x == 12)

puts("3.位置");

else if (x == 14)

puts("4.入住");

else

puts("5.费用");

i--;

}

else

inversion(&i,90,&x,5,18,"0.返回上一级界面");

break;

case 80:

if (i<5)

{

x = cColor(90,x,1);

if (x == 10)

puts("2.房型");

else if (x == 12)

puts("3.位置");

else if (x == 14)

puts("4.入住");

else if (x == 16)

puts("5.费用");

else

puts("0.返回上一级界面");

i++;

}

else

inversion(&i,90,&x,0,8,"1.房间号");

break;

case 13: system("cls");

if (x == 8)

cNum(head);

else if (x == 10)

cType(head);

else if (x == 12)

cSite(head);

else if (x == 14)

cCheckIn(head);

else if (x == 16)

cCost(head);

else

return;

goto A;

case 49: system("cls");ShowCursor();cNum(head);goto A;

case 50: system("cls");ShowCursor();cType(head);goto A;

case 51: system("cls");ShowCursor();cSite(head);goto A;

case 52: system("cls");ShowCursor();cCheckIn(head);goto A;

case 53: system("cls");ShowCursor();cCost(head);goto A;

case 48: return;

}

}

}

/\*这是一个C语言函数，名为change，用于修改房间信息。该函数接受一个名为head的room类型结构体指针作为参数。

在该函数中，首先调用write函数将链表中的房间信息写入文件中。然后定义了一些变量，如x、i和t等。其中，x表示光标所在位置的纵坐标，i表示当前选项的序号，t表示用户按下的键盘字符。

接下来，调用cMenu函数绘制菜单，并进入while循环。在循环中，根据用户输入的上下箭头，移动光标并高亮显示相应的选项。如果用户按下回车键，则根据光标所在的位置，调用相应的函数进行房间信息的修改。如果用户按下数字键0，则返回上一级界面。

在修改房间信息的过程中，如果需要输入新的值，则清空屏幕并调用ShowCursor函数显示控制台光标，然后调用相应的函数，等待用户输入新的值。完成输入后，再次调用goto A返回到主界面。

该函数主要用于在程序运行过程中修改房间信息，方便管理员对房间信息进行管理和维护。\*/

/\*\*\*\*\*\*房间信息修改菜单\*\*\*\*\*\*\*\*/

void cMenu()

{

form(5,6);

gotoxy(87,6);

puts("酒店房间信息更改");

gotoxy(90,10);

puts("2.房型");

gotoxy(90,12);

puts("3.位置");

gotoxy(90,14);

puts("4.入住");

gotoxy(90,16);

puts("5.费用");

gotoxy(90,18);

puts("0.返回上一级界面");

red();

gotoxy(90,8);

puts("1.房间号");

}

/\*此函数实现的是一个酒店房间信息修改的菜单界面，其中:

- form(5,6) 是将光标移动到(5,6)的位置，可以理解为定位。

- gotoxy() 是将光标移动到指定的位置，puts() 是输出字符串，用于在屏幕上显示文字信息。

- red() 函数是改变文字颜色为红色，在这里可能表示该选项比较重要或者需要特别注意。

该菜单提供了五个选项，分别是修改房型、位置、入住、费用以及返回上一级界面。其中房间号作为唯一标识符，不在此处进行修改。\*/

/\*\*\*\*\*\*\*\*\*\*房间信息修改通式\*\*\*\*\*\*\*\*/

struct room \*amend(struct room \*head)

{

struct room \*p1;

char amendNum[20];

A:printf("请输入要修改房间信息的房间号:");

if (scanfPlus(amendNum,1,9,0) == 1)

return NULL;

p1 = head;

while(p1 != NULL) //遍历链表直到遍历到要修改的房间信息的位置

{

if (strcmp(p1->num,amendNum) == 0)

{

printf("该房间信息如下:\n\n");

gotoxy(63,5);

puts("┌───────────────┬────────────────┬───────────────┬───────────────┬───────────────┐");

gotoxy(63,6);

puts("│ │ │ │ │ │");

gotoxy(63,7);

puts("├───────────────┼────────────────┼───────────────┼───────────────┼───────────────┤");

gotoxy(63,8);

puts("│ │ │ │ │ │");

gotoxy(63,9);

puts("└───────────────┴────────────────┴───────────────┴───────────────┴───────────────┘");

gotoxy(69,6);

printf("房间号");

gotoxy(86,6);

printf("房型");

gotoxy(103,6);

printf("位置");

gotoxy(117,6);

printf("入住");

gotoxy(135,6);

printf("费用(天)");

gotoxy(70,8);

printf("%s",p1->num);

gotoxy(85,8);

printf("%s",p1->type);

gotoxy(103,8);

printf("%s",p1->site);

gotoxy(117,8);

printf("%s",p1->check);

gotoxy(135,8);

printf("%s",p1->cost);

return p1;

}

p1 = p1->next;

}

MessageBox(NULL,"未查到此房间号请重新输入!","错误",MB\_ICONHAND);

system("cls");

goto A;

}

/\*这段代码是一个房间信息修改的通用函数，其中：

- 定义了一个结构体 room，表示一个房间的基本信息。

- amend() 函数接收一个指向链表头结点的指针 head，并返回一个指向修改后的链表头结点的指针。

- 通过输入要修改房间信息的房间号 amendNum，可以定位到需要修改的房间信息。

- scanfPlus() 函数用于限制输入字符数，确保输入的房间号在1到9个字符之间，且不能包含空格。

- while 循环遍历整个链表，找到与输入的房间号相匹配的房间信息。

- 如果找到了需要修改的房间信息，就可以进行相应的修改操作，如修改房型、位置、入住时间、费用等。

- 最终返回修改后的链表头结点的指针。\*/

/\*\*\*\*修改房间号\*\*\*\*\*\*/

void cNum(struct room \*head)

{

struct room \*p1,\*p2;

if ((p1 = amend(head)) == NULL)

return;

gotoxy(13,10);

A:printf("请输入修改后的房间号:");

if (scanfPlus(p1->num,1,9,0) == 1)

return;

p2 = head;

while (p2 != NULL)

{

if (strcmp(p1->num,p2->num) == 0 && p1 != p2) //判断房间号是否重复

{

MessageBox(NULL,"此房间号已存在请重新输入","错误",MB\_ICONHAND);

system("cls");

goto A;

}

p2 = p2->next;

}

MessageBox(NULL,"修改成功！","提示",MB\_ICONASTERISK);

}

/\*此函数用于修改房间号，其中：

- 定义了一个指向 room 结构体的指针 p1 和 p2，用于遍历链表和存储需要修改的房间信息。

- 调用 amend() 函数获取要修改的房间信息的地址，并将其赋值给 p1。

- 通过输入修改后的房间号，可以更新需要修改的房间信息中的房间号。

- scanfPlus() 函数用于限制输入字符数，确保输入的房间号在1到9个字符之间，且不能包含空格。

- 在修改房间号之前，需要检查新的房间号是否已经存在于链表中。为此，需要再次遍历整个链表，如果找到与修改后的房间号相同的房间信息，则表示该房间号已被占用，需要重新输入。

- 如果发现重复房间号，则会显示一个错误提示消息框，并要求用户重新输入新的房间号。

- 最终如果修改成功，会显示一个提示消息框，告知用户修改已成功完成。\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*修改房型\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void cType(struct room \*head)

{

struct room \*p1;

if ((p1 = amend(head)) == NULL)

return;

gotoxy(13,10);

printf("请输入修改后的房型:");

if (scanfPlus(p1->type,1,19,0) == 1)

return;

MessageBox(NULL,"修改成功！","提示",MB\_ICONASTERISK);

}

/\*此函数功能是修改房间的类型。具体实现过程如下：

1. 首先定义一个指向结构体room的指针p1，并调用amend函数对p1进行赋值（该函数未给出代码，不了解其具体实现）。

2. 如果p1为空指针，则直接返回。

3. 使用gotoxy函数将光标移动到屏幕相应位置处，并提示用户输入修改后的房型信息。

4. 调用scanfPlus函数读取用户输入的字符串，并将其存储到p1所指向的结构体变量中的type成员中。

5. 如果输入成功，直接返回并弹出“修改成功”的提示框。

总体来说，这段代码比较简单，主要实现了用户交互和数据修改的功能。但在实际使用中还需要考虑一些细节问题，比如输入字符串的长度限制、数据是否合法等等。\*/

/\*\*\*\*\*修改房间位置\*\*\*\*\*\*/

void cSite(struct room \*head)

{

struct room \*p1;

if ((p1 = amend(head)) == NULL)

return;

gotoxy(13,10);

printf("请输入修改后的位置:");

if (scanfPlus(p1->site,1,9,0) == 1)

return;

MessageBox(NULL,"修改成功！","提示",MB\_ICONASTERISK);

}

/\*此函数功能是修改某个房间的位置信息。具体实现过程如下：

1. 首先定义一个指向结构体room的指针p1，并调用amend函数对p1进行赋值（该函数未给出代码，不了解其具体实现）。

2. 如果p1为空指针，则直接返回。

3. 使用gotoxy函数将光标移动到屏幕相应位置处，并提示用户输入修改后的位置信息。

4. 调用scanfPlus函数读取用户输入的字符串，并将其存储到p1所指向的结构体变量中的site成员中。

5. 如果输入成功，直接返回并弹出“修改成功”的提示框。

总体来说，这段代码也比较简单，主要实现了用户交互和数据修改的功能。但在实际使用中还需要考虑一些细节问题，比如输入字符串的长度限制、数据是否合法等等。同时，这段代码与前面那段修改房型的代码非常类似，可以进一步封装为一个通用的修改函数以提高代码复用性。\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*修改房间入住情况\*\*\*\*\*\*\*\*\*\*\*/

void cCheckIn(struct room \*head)

{

struct room \*p1;

if ((p1 = amend(head)) == NULL)

return;

gotoxy(13,10);

A:printf("请输入修改后的入住情况:");

if (scanfPlus(p1->check,1,9,0) == 1)

return;

if (strcmp(p1->check,"已入住") != 0 && strcmp(p1->check,"未入住") != 0)

{

MessageBox(NULL,"非法输入请重新输入!","错误",MB\_ICONHAND);

system("cls");

goto A;

}

MessageBox(NULL,"修改成功！","提示",MB\_ICONASTERISK);

}

/\*此函数的功能是修改某个房间的入住情况。具体实现过程如下：

1. 首先定义一个指向结构体room的指针p1，并调用amend函数对p1进行赋值（该函数未给出代码，不了解其具体实现）。

2. 如果p1为空指针，则直接返回。

3. 使用gotoxy函数将光标移动到屏幕相应位置处，并提示用户输入修改后的入住情况信息。

4. 调用scanfPlus函数读取用户输入的字符串，并将其存储到p1所指向的结构体变量中的check成员中。

5. 如果输入成功，直接返回并弹出“修改成功”的提示框。

6. 但如果用户输入的字符串既不是“已入住”，也不是“未入住”，则弹出一个提示框告诉用户“非法输入请重新输入!”。然后清屏并回到步骤3，等待用户重新输入。

总体来说，这段代码比前面两段稍微复杂一些，因为需要判断用户输入的字符串是否合法，否则会陷入循环。同时，还存在一些问题需要进一步优化，比如当用户连续多次输入非法字符串时，程序是否会进入死循环等。\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*修改一天的费用\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void cCost(struct room \*head)

{

struct room \*p1;

if ((p1 = amend(head)) == NULL)

return;

gotoxy(13,10);

printf("请输入修改后的费用(每天):");

if (scanfPlus(p1->cost,1,15,0) == 1)

return;

MessageBox(NULL,"修改成功！","提示",MB\_ICONASTERISK);

}

/\*此函数功能是修改某个房间每天的费用。具体实现过程如下：

1. 首先定义一个指向结构体room的指针p1，并调用amend函数对p1进行赋值。

2. 如果p1为空指针，则直接返回。

3. 使用gotoxy函数将光标移动到屏幕相应位置处，并提示用户输入修改后的每天的费用信息。

4. 调用scanfPlus函数读取用户输入的字符串，并将其存储到p1所指向的结构体变量中的cost成员中。

5. 如果输入成功，直接返回并弹出“修改成功”的提示框。

总体来说，这段代码也比较简单，主要实现了用户交互和数据修改的功能。但在实际使用中还需要考虑一些细节问题，比如输入字符串的长度限制、数据是否合法等等。同时，这段代码与前面那两段非常类似，也可以进一步封装为一个通用的修改函数以提高代码复用性。\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*管理端对申请进行处理\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void dispose(struct room \*head)

{

struct ID \*head1,\*x1;

struct room \*p1;

char judge[3],name[20];

head1 = loginRead();

x1 = head1;

while (x1 != NULL) //遍历客户信息检索申请

{

if (strcmp(x1->checkOut,"是") == 0)

{

p1 = head;

while (p1 != NULL) //遍历到该客户所退的房间

{

if (strcmp(p1->num,x1->num) == 0)

break;

p1 = p1->next;

}

strcpy(p1->status,"已预订");

strcpy(name,x1->name);

strcpy(p1->name,x1->inputName);

strcpy(p1->tel,x1->inputTel);

p1->year = x1->year;

p1->mon = x1->mon;

p1->day = x1->day;

p1->days = x1->numDay;

A:form2(1);

form3(p1,0);

gotoxy(13,14);

if(MessageBox(NULL,"是否同意该客户的退房申请","询问",MB\_ICONASTERISK|MB\_OKCANCEL) != IDCANCEL)

{

strcpy(x1->checkOut,"已退房");

strcpy(x1->num,"无");

// x1->daymax = 0;

p1->days = 0;

p1->year = 1;

strcpy(p1->tel,"无");

strcpy(p1->idnum,"无");

strcpy(p1->status,"空闲");

strcpy(p1->check,"未入住");

strcpy(p1->name,"无");

MessageBox(NULL,"处理成功！","提示",MB\_ICONASTERISK);

loginWrite(head1);

return;

}

strcpy(x1->checkOut,"退房失败");

gotoxy(13,15);

printf("请输入拒绝其退房的原因:");

if (scanfPlus(x1->information,1,99,0) == 1)

return;

MessageBox(NULL,"处理成功！","提示",MB\_ICONASTERISK);

loginWrite(head1);

return;

}

x1 = x1->next;

}

MessageBox(NULL,"无待处理的申请!","错误",MB\_ICONHAND);

}

/\*此函数的功能是处理客户的退房申请。具体实现过程如下：

1. 首先定义一个指向结构体ID的指针head1，并调用loginRead函数将登录信息从文件中读取到链表中。

2. 使用while循环遍历客户信息链表head1，查找是否有待处理的退房申请。

3. 如果发现某个客户已经申请退房，则使用另外一个while循环遍历房间信息链表head，查找该客户所住的房间。

4. 修改该房间的状态、入住信息等相关信息。

5. 在屏幕上显示修改后的房间信息，并询问用户是否同意该客户的退房申请。

6. 如果用户同意该退房申请，则将该客户的相关信息清零，并将其标记为已退房。同时清空该房间的入住天数、电话号码、身份证号码等信息，并将其状态设置为“空闲”、入住情况设置为“未入住”、客户姓名设置为“无”。最后弹出“处理成功”的提示框并退出函数。

7. 如果用户不同意该退房申请，则在该客户信息中记录下拒绝原因，并将其标记为“退房失败”。然后弹出“处理成功”的提示框并退出函数。

8. 如果没有待处理的退房申请，则弹出“无待处理的申请”提示框并退出函数。

总体来说，这段代码比较复杂，需要涉及到客户信息、房间信息、用户交互等多个方面。同时，对于输入数据的合法性判断也不够严格，存在一定的安全隐患。所以在实际使用中还需要进行进一步的优化和改进。\*/

/\*\*\*\*\*\*\*\*删除房间信息\*\*\*\*\*\*\*\*\*\*\*/

struct room \*del(struct room \*head,struct ID \*head1)

{

struct room \*p2,\*p3;

struct ID \*x1;

char delnum[20];

if (head == NULL)

{

MessageBox(NULL,"无任何房间信息!","提示",MB\_ICONASTERISK);

return head;

}

printf("请输入要删除房间信息的房间号:");

if (scanfPlus(delnum,1,9,0) == 1)

return head;

x1 = head1;

while (x1 != NULL)

{

if (strcmp(x1->num,delnum) == 0)

{

MessageBox(NULL,"该房间有人预定不能删除","警告", MB\_ICONEXCLAMATION );

return head;

}

x1 = x1->next;

}

p2 = head;

while (p2 != NULL)

{

if (strcmp(delnum,p2->num) == 0)

{

A:system("cls");

form2(1);

form3(p2,0);

gotoxy(13,10);

if(MessageBox(NULL,"是否删除该房间信息","询问",MB\_ICONASTERISK|MB\_OKCANCEL)!=IDCANCEL)

{

if (p2 == head)

{

head = p2->next;

free(p2);

MessageBox(NULL,"删除成功!","提示",MB\_ICONASTERISK);

return head;

}

else

{

p3->next = p2->next;

free(p2);

MessageBox(NULL,"删除成功!","提示",MB\_ICONASTERISK);

return head;

}

}

MessageBox(NULL,"已取消删除！","提示",MB\_ICONASTERISK);

return head;

}

p3 = p2;

p2 = p2->next;

}

MessageBox(NULL,"查无此房间信息!","提示",MB\_ICONASTERISK);

return head;

}

/\*这是一个删除房间信息的函数。函数包含两个指针，一个指向房间信息表头，另一个指向已预定房间的链表头部。

函数首先会检查房间信息表是否为空。如果为空，则会弹出提示框告知用户无任何房间信息，并返回原始表头。

然后函数会要求用户输入待删除的房间号码。如果用户输入的不符合要求，函数会继续等待输入。

接下来，函数会遍历已预定的房间列表，检查用户希望删除的房间是否被预订。如果该房间已被预订，则弹出提示窗口告知用户不能删除该房间，函数将返回原始表头。

在房间信息表中查找到了需要删除的房间后，函数会弹出提示框询问用户是否删除该房间信息。如果用户选择取消删除，则函数将返回原始表头。

如果用户决定删除该房间信息，则该函数会将该房间从链表中删除，并释放内存空间。如果该房间是第一个节点，则修改表头指针；否则，函数会将前一个房间节点与该房间节点的后一个节点连接起来。

最后，函数会弹出提示框告知用户删除成功，并返回修改后的房间信息表头。如果没有找到需要删除的房间，则弹出提示框告知用户查无此房间信息，并返回原始表头。\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*根据房间信息进行查找\*\*\*\*\*\*\*\*\*\*\*\*\*/

void finds(struct room \*head,struct ID \*head1)

{

int x,i;

char t;

A:system("cls");

x = 6;

i = 0;

fMenu();

while (1)

{

t = arrows(x,'0','9');

switch(t)

{

case 72:

if (i>0)

{

x = cColor(92,x,0);

if (x == 6)

puts("1.房间号");

else if (x == 8)

puts("2.房型");

else if (x == 10)

puts("3.位置");

else if (x == 12)

puts("4.房间状态");

else if (x == 14)

puts("5.入住时间");

else if (x == 16)

puts("6.天数");

else if (x == 18)

puts("7.费用");

else if (x == 20)

puts("8.入住客户姓名");

else if (x == 22)

puts("9.电话");

i--;

}

else

inversion(&i,92,&x,9,24,"0.返回上一级界面");

break;

case 80:

if (i<9)

{

x = cColor(92,x,1);

if (x == 8)

puts("2.房型");

else if (x == 10)

puts("3.位置");

else if (x == 12)

puts("4.房间状态");

else if (x == 14)

puts("5.入住时间");

else if (x == 16)

puts("6.天数");

else if (x == 18)

puts("7.费用");

else if (x == 20)

puts("8.入住客户姓名");

else if (x == 22)

puts("9.电话");

else

puts("0.返回上一级界面");

i++;

}

else

inversion(&i,92,&x,0,6,"1.房间号");

break;

case 13: system("cls");

if (x == 6)

fNum(head,head1);

else if (x == 8)

fType(head,head1);

else if (x == 10)

fSite(head,head1);

else if (x == 12)

fStatus(head,head1);

else if (x == 14)

fInputTime(head,head1);

else if (x == 16)

fDays(head,head1);

else if (x == 18)

fCost(head,head1);

else if (x == 20)

fName(head,head1);

else if (x == 22)

fTel(head,head1);

else

return;

goto A;

case 49: system("cls");ShowCursor();fNum(head,head1);goto A;

case 50: system("cls");ShowCursor();fType(head,head1);goto A;

case 51: system("cls");ShowCursor();fSite(head,head1);goto A;

case 52: system("cls");ShowCursor();fStatus(head,head1);goto A;

case 53: system("cls");ShowCursor();fInputTime(head,head1);goto A;

case 54: system("cls");ShowCursor();fDays(head,head1);goto A;

case 55: system("cls");ShowCursor();fCost(head,head1);goto A;

case 56: system("cls");ShowCursor();fName(head,head1);goto A;

case 57: system("cls");ShowCursor();fTel(head,head1);goto A;

case 48: return;

}

}

}

/\*这是一个根据房间信息进行查找的函数，通过用户在菜单中选择不同的选项实现对房间信息的查找。

该函数首先会清除屏幕，然后显示一个菜单供用户进行选择。接着，函数通过控制台输入读取用户的选择，根据不同的选择执行相应的查找操作。

当用户选择某个查找选项时，该函数会清除屏幕，并进入到相应的查找函数中，在函数结束后跳转回查找主界面。如果用户选择了返回上一级界面的选项，则函数调用返回。\*/

/\*\*\*\*\*\*\*\*\*\*\*根据不同局部房间信息查找的菜单\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void fMenu()

{

form(3,10);

gotoxy(88,4);

puts("查询酒店房间信息");

gotoxy(92,8);

puts("2.房型");

gotoxy(92,10);

puts("3.位置");

gotoxy(92,12);

puts("4.房间状态");

gotoxy(92,14);

puts("5.入住时间");

gotoxy(92,16);

puts("6.天数");

gotoxy(92,18);

puts("7.费用");

gotoxy(92,20);

puts("8.入住客户姓名");

gotoxy(92,22);

puts("9.电话");

gotoxy(92,24);

puts("0.返回上一级菜单");

gotoxy(92,6);

red();

puts("1.房间号");

}

/\*这是一个用于显示根据不同局部房间信息进行查找的菜单的函数。

该函数通过控制台输出在屏幕上显示菜单。菜单中包括各种不同的选项，例如：房型、位置、房间状态、入住时间、天数、费用、入住客户姓名和电话等。每个选项都有对应的数字和功能描述，用户可以根据需求选择相应的数字进入到相应的查找函数中。

菜单的底部还有一个返回上一级菜单的选项，使得用户可以方便地返回到上一级菜单。\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*根据房间号进行查找\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void fNum(struct room \*head,struct ID \*head1)

{

struct room \*p1;

struct ID \*x1;

char rNum[10];

int flag = 0,i,n = 0;

printf("请输入待查找房间的房间号:");

if (scanfPlus(rNum,1,9,0) == 1)

return;

p1 = head;

while(p1 != NULL)

{

if (strstr(p1->num,rNum) != 0)

n++;

p1 = p1->next;

}

if (n == 0)

{

MessageBox(NULL,"未查到此房间信息!","提示",MB\_ICONASTERISK);

return;

}

do

{

if (flag == 1)

form2(i);

p1 = head;

i = 0;

while (p1 != NULL)

{

if (strstr(p1->num,rNum) != 0)

{

x1 = head1;

while (x1 != NULL)

{

if (strcmp(p1->num,x1->num) == 0)

{

status(p1,x1);

form3(p1,i);

i++;

}

x1 = x1->next;

}

if (strcmp(p1->status,"空闲") == 0)

{

form3(p1,i);

i++;

}

}

p1 = p1->next;

}

flag++;

}

while (flag == 1);

gotoxy(13,10+2\*i);

system("pause");

}

/\*\*\*\*\*\*\*\*\*\*\*\*根据房型进行查找\*\*\*\*\*\*\*\*\*\*\*\*\*/

void fType(struct room \*head,struct ID \*head1)

{

struct room \*p1;

struct ID \*x1;

char rType[10];

int flag = 0,i,n = 0;

printf("请输入待查找房间的房型:");

if (scanfPlus(rType,1,9,0) == 1)

return;

p1 = head;

while(p1 != NULL)

{

if (strstr(p1->type,rType) != 0)

n++;

p1 = p1->next;

}

if (n == 0)

{

MessageBox(NULL,"未查到此房间信息!","提示",MB\_ICONASTERISK);

return;

}

do

{

if (flag == 1)

form2(i);

p1 = head;

i = 0;

while (p1 != NULL)

{

if (strstr(p1->type,rType) != 0)

{

x1 = head1;

while (x1 != NULL)

{

if (strcmp(p1->num,x1->num) == 0)

{

status(p1,x1);

form3(p1,i);

i++;

}

x1 = x1->next;

}

if (strcmp(p1->status,"空闲") == 0)

{

form3(p1,i);

i++;

}

}

p1 = p1->next;

}

flag++;

}

while (flag == 1);

gotoxy(13,10+2\*i);

system("pause");

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*根据房间位置进行查找\*\*\*\*\*\*\*\*\*\*\*\*\*/

void fSite(struct room \*head,struct ID \*head1)

{

struct room \*p1;

struct ID \*x1;

char rSite[10];

int flag = 0,i,n = 0;

printf("请输入待查找房间的位置:");

if (scanfPlus(rSite,1,9,0) == 1)

return;

p1 = head;

while(p1 != NULL)

{

if (strstr(p1->site,rSite) != 0)

n++;

p1 = p1->next;

}

if (n == 0)

{

MessageBox(NULL,"未查到此房间信息!","提示",MB\_ICONASTERISK);

return;

}

do

{

if (flag == 1)

form2(i);

p1 = head;

i = 0;

while (p1 != NULL)

{

if (strstr(p1->site,rSite) != 0)

{

x1 = head1;

while (x1 != NULL)

{

if (strcmp(p1->num,x1->num) == 0)

{

status(p1,x1);

form3(p1,i);

i++;

}

x1 = x1->next;

}

if (strcmp(p1->status,"空闲") == 0)

{

form3(p1,i);

i++;

}

}

p1 = p1->next;

}

flag++;

}

while (flag == 1);

gotoxy(13,10+2\*i);

system("pause");

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*根据房间状态进行查找\*\*\*\*\*\*\*\*\*\*\*\*/

void fStatus(struct room \*head,struct ID \*head1)

{

struct room \*p1;

struct ID \*x1;

char rStatus[10];

int flag = 0,flag2 = 0,i,n = 0;

printf("请输入待查找房间的状态:");

if (scanfPlus(rStatus,1,9,0) == 1)

return;

p1 = head;

while (p1 != NULL)

{

x1 = head1;

while (x1 != NULL)

{

if (strcmp(p1->num,x1->num) == 0)

strcpy(p1->status,"已预订");

x1 = x1->next;

}

p1 = p1->next;

}

p1 = head;

while(p1 != NULL)

{

if (strstr(p1->status,rStatus) != 0)

n++;

p1 = p1->next;

}

if (n == 0)

{

MessageBox(NULL,"未查到此房间信息!","提示",MB\_ICONASTERISK);

return;

}

do

{

if (flag == 1)

form2(i);

p1 = head;

i = 0;

while (p1 != NULL)

{

if (strstr(p1->status,rStatus) != 0)

{

x1 = head1;

while (x1 != NULL)

{

if (strcmp(p1->num,x1->num) == 0)

{

status(p1,x1);

form3(p1,i);

i++;

flag2 = 1;

}

x1 = x1->next;

}

if (flag2 == 0)

{

form3(p1,i);

i++;

}

flag2 = 0;

}

p1 = p1->next;

}

flag++;

}

while (flag == 1);

gotoxy(13,10+2\*i);

system("pause");

}

/\*\*\*\*\*\*\*\*\*\*\*\*根据入住时间进行查找\*\*\*\*\*\*\*\*\*\*\*/

void fInputTime(struct room \*head,struct ID \*head1)

{

struct room \*p1;

struct ID \*x1;

int i,n=0;

char tran[5];

int year,mon,day;

printf("请输入待查房间客户的入住时间:");

gotoxy(34,0);

printf("年");

gotoxy(41,0);

printf("月");

gotoxy(46,0);

printf("日");

gotoxy(30,0);

if (scanfPlus(tran,1,4,0) == 1)

return;

year = atoi(tran);

gotoxy(38,0);

if (scanfPlus(tran,1,2,0) == 1)

return;

mon = atoi(tran);

gotoxy(43,0);

if (scanfPlus(tran,1,2,0) == 1)

return;

day = atoi(tran);

x1 = head1;

while (x1 != NULL)

{

if (x1->year == year && x1->mon == mon && x1->day == day)

n++;

x1 = x1->next;

}

if (n == 0)

{

MessageBox(NULL,"未查到此房间信息!","提示",MB\_ICONASTERISK);

return;

}

form2(n);

p1 = head;

i = 0;

while (p1 != NULL)

{

x1 = head1;

while (x1 != NULL)

{

if (strcmp(p1->num,x1->num) == 0)

{

if (x1->year == year && x1->mon == mon && x1->day == day)

{

status(p1,x1);

form3(p1,i);

i++;

}

}

x1 = x1->next;

}

p1 = p1->next;

}

gotoxy(13,10+2\*i);

system("pause");

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*根据入住天数进行查找\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void fDays(struct room \*head,struct ID \*head1)

{

struct room \*p1;

struct ID \*x1;

char tran[10];

int rDays;

int i,n=0;

printf("请输入待查房间客户入住的天数:");

if (scanfPlus(tran,1,9,0) == 1)

return;

rDays = atoi(tran);

x1 = head1;

while (x1 != NULL)

{

if (x1->numDay == rDays)

n++;

x1 = x1->next;

}

if (n == 0)

{

MessageBox(NULL,"未查到此房间信息!","提示",MB\_ICONASTERISK);

return;

}

form2(n);

p1 = head;

i = 0;

while (p1 != NULL)

{

x1 = head1;

while (x1 != NULL)

{

if (strcmp(p1->num,x1->num) == 0)//字符串判断

{

if (x1->numDay == rDays)

{

status(p1,x1);

form3(p1,i);

i++;

}

}

x1 = x1->next;

}

p1 = p1->next;

}

gotoxy(13,10+2\*i);

system("pause");

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*根据住一天的房间费用进行查找\*\*\*\*\*\*\*\*\*\*/

void fCost(struct room \*head,struct ID \*head1)

{

struct room \*p1;

struct ID \*x1;

char rCost[10];

int flag = 0,i,n = 0;

printf("请输入待查找房间每天的费用:");

if (scanfPlus(rCost,1,9,0) == 1)

return;

p1 = head;

while(p1 != NULL)

{

if (strstr(p1->cost,rCost) != 0)

n++;

p1 = p1->next;

}

if (n == 0)

{

MessageBox(NULL,"未查到此房间信息!","提示",MB\_ICONASTERISK);

return;

}

do

{

if (flag == 1)

form2(i);

p1 = head;

i = 0;

while (p1 != NULL)

{

if (strstr(p1->cost,rCost) != 0)

{

x1 = head1;

while (x1 != NULL)

{

if (strcmp(p1->num,x1->num) == 0)

{

status(p1,x1);

form3(p1,i);

i++;

}

x1 = x1->next;

}

if (strcmp(p1->status,"空闲") == 0)

{

form3(p1,i);

i++;

}

}

p1 = p1->next;

}

flag++;

}

while (flag == 1);

gotoxy(13,10+2\*i);

system("pause");

}

/\*\*\*\*\*\*\*\*\*\*\*根据入住客户姓名进行查找\*\*\*\*\*\*\*\*\*\*\*\*/

void fName(struct room \*head,struct ID \*head1)

{

struct room \*p1;

struct ID \*x1;

char rName[20];

int i,n = 0;

printf("请输入要查找的房间信息的入住客户名:");

if (scanfPlus(rName,1,9,0) == 1)

return;

x1 = head1;

while (x1 != NULL)

{

if (strstr(x1->inputName,rName) != 0)

n++;

x1 = x1->next;

}

if (n == 0)

{

MessageBox(NULL,"未查到此房间信息!","提示",MB\_ICONASTERISK);

return;

}

form2(n);

p1 = head;

i = 0;

while (p1 != NULL)

{

x1 = head1;

while (x1 != NULL)

{

if (strcmp(p1->num,x1->num) == 0)

{

if (strstr(x1->inputName,rName) != 0)

{

status(p1,x1);

form3(p1,i);

i++;

}

}

x1 = x1->next;

}

p1 = p1->next;

}

gotoxy(13,10+2\*i);

system("pause");

}

/\*\*\*\*\*\*\*\*\*\*根据电话号码进行查找\*\*\*\*\*\*\*\*\*/

void fTel(struct room \*head,struct ID \*head1)

{

struct room \*p1;

struct ID \*x1;

char rTel[12];

int i,n = 0;

printf("请输入要查找的房间信息的电话:");

if (scanfPlus(rTel,1,11,3) == 1)

return;

x1 = head1;

while (x1 != NULL)

{

if (strstr(x1->inputTel,rTel) != 0)

n++;

x1 = x1->next;

}

if (n == 0)

{

MessageBox(NULL,"未查到此房间信息!","提示",MB\_ICONASTERISK);

return;

}

form2(n);

p1 = head;

i = 0;

while (p1 != NULL)

{

x1 = head1;

while (x1 != NULL)

{

if (strcmp(p1->num,x1->num) == 0)

{

if (strstr(x1->inputTel,rTel) != 0)

{

status(p1,x1);

form3(p1,i);

i++;

}

}

x1 = x1->next;

}

p1 = p1->next;

}

gotoxy(13,10+2\*i);

system("pause");

}

/\*\*\*\*\*\*\*\*\*\*\*\*登录主函数\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

struct ID \*login()

{

struct ID \*head = NULL,\*p;

head = loginRead();

int x,i;

char t;

A:system("cls");

loginWrite(head);

loginMenu();

x = 9;

i = 0;

while (1)

{

t = arrows(x,'0','6');

switch(t)

{

case 72:

if (i > 0)

{

x = cColor(93,x,0);

if (x == 9)

puts("1.登录");

else if (x == 11)

puts("2.注册");

else if (x == 13)

puts("3.注销");

else if (x == 15)

puts("4.密码找回");

else if (x == 17)

puts("5.修改密码");

else if (x == 19)

puts("6.游客登录");

i--;

}

else

{

inversion(&i,93,&x,6,21,"0.退出");

}

break;

case 80:

if (i < 6)

{

x = cColor(93,x,1);

if (x == 11)

puts("2.注册");

else if (x == 13)

puts("3.注销");

else if (x == 15)

puts("4.密码找回");

else if (x == 17)

puts("5.修改密码");

else if (x == 19)

puts("6.游客登录");

else

puts("0.退出");

i++;

}

else

{

inversion(&i,93,&x,0,9,"1.登录");

}

break;

case 13: system("cls");

if (x == 9)

{

if ((p = sign(head)))

return p; //登录后返回该账号的基本信息

}

if (x == 11)

head = accountName(head);

if (x == 13)

head = logout(head);

if (x == 15)

signFind(head);

if (x == 17)

changePassword(head);

if (x == 19)

visitor();

if (x == 21)

exit(0);

goto A;

case 49:

system("cls");

ShowCursor();

if ((p = sign(head)))

return p;

goto A;

case 50: system("cls");ShowCursor();head = accountName(head); goto A;

case 51: system("cls");ShowCursor();head = logout(head); goto A;

case 52: system("cls");ShowCursor();signFind(head); goto A;

case 53: system("cls");ShowCursor();changePassword(head); goto A;

case 54: system("cls");ShowCursor();visitor(); goto A;

case 48: exit(0);

}

}

}

/\*\*\*\*\*\*登录界面菜单\*\*\*\*\*\*\*\*/

void loginMenu()

{

system("cls");

form(6,7);

gotoxy(92,7);

puts("446五星酒店");

gotoxy(93,11);

puts("2.注册");

gotoxy(93,13);

puts("3.注销");

gotoxy(93,15);

puts("4.密码找回");

gotoxy(93,17);

puts("5.修改密码");

gotoxy(93,19);

puts("6.游客登录");

gotoxy(93,21);

puts("0.退出");

gotoxy(70,23);

puts("Tip:在录入数据的过程中按ESC键可直接取消录入返回上一级");

gotoxy(93,9);

red();

puts("1.登录");

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*将文件中的账户信息读出\*\*\*\*\*\*\*\*\*\*\*\*/

struct ID \*loginRead()

{

FILE \*fp;

char ch;

struct ID \*head = NULL,\*p1,\*p2; //将头指针初始化为空

if ((fp = fopen("用户","rb")) == NULL)

return head;

A:p1 = (struct ID \*)malloc(sizeof(struct ID));

fread(p1,sizeof(struct ID),1,fp);

if(feof(fp)) //判断文件是否为空

{

fclose(fp);

return head;

}

if (head == NULL) //因空链表特殊性，头结点需单独读取

{

head = p1;

p1->next = NULL;

p2 = p1;

goto A;

}

p2->next = p1; //遍历链表依次从文件中读取数据到链表

p1->next = NULL;

p2 = p1;

goto A;

}

/\*此函数返回一个结构体指针类型的结果，用于保存读取出的链表头指针。

首先打开一个名为“用户”的二进制文件，如果文件为空则直接返回空指针；否则，通过循环遍历读取文件中的每个结构体，将其转化为一个节点，并依次连接成链表，最后返回链表头指针。

定义一个文件指针fp和两个结构体指针变量p1、p2，同时将头指针初始化为空。

打开一个名为“用户”的二进制文件，以只读（"rb"）方式打开。如果打开失败，则直接返回空指针。

定义一个标签A，用于后面的跳转。使用动态内存分配函数malloc()申请一个struct ID大小的空间，然后使用fread()从文件中读取一个struct ID大小的块并将其存储到p1指向的内存空间中。

如果文件已经结束，即读取到文件末尾，则关闭文件并返回链表头指针。

如果链表为空，则将当前节点设置为头节点，并将其指向下一个节点的指针next设置为空（因为当前节点就是链表中唯一的节点），同时使用指针变量p2记录当前节点。

如果链表不为空，则将当前节点连接到链表尾部，并更新p2指向当前节点，然后跳转回标签A处继续读取文件内容，直到文件结束。\*/

/\*用户登录函数\*/

struct ID \*sign(struct ID \*head)

{

int i = 2;

struct ID \*p1 = NULL,\*p2;

char accountt[20];

char key1[20];

char testGetCode[7];

char code[7];

if (head == NULL)

{

MessageBox(NULL,"未注册任何账号请注册后重试!","提示",MB\_ICONASTERISK);

return NULL;

}

A:form(9,2);

gotoxy(94,10);

puts("登录");

gotoxy(82,12);

puts("账号:");

gotoxy(82,14);

puts("密码:");

gotoxy(88,12);

if (p1 == NULL)

{

if (scanfPlus(accountt,1,18,0) == 1)

return NULL;

}

else

puts(accountt);

p1 = head;

while(1)

{

if (strcmp(accountt,p1->account) == 0)

break;

p1 = p1->next;

if (p1 == NULL)

{

MessageBox(NULL,"无此账号请重新输入!","错误", MB\_ICONHAND);

system("cls");

goto A;

}

}

gotoxy(88,14);

if (scanfPlus(key1,1,18,1) == 1)

return NULL;

C:system("cls");

form(9,2);

gotoxy(94,10);

printf("验证");

gotoxy(82,12);

printf("验证码:%s\n",randStr(testGetCode,4));

gotoxy(82,14);

printf("请输入验证码(区分大小写):");

gotoxy(107,14);

if (scanfPlus(code,1,4,0) == 1)

return NULL;

if (strcmp(testGetCode,code) != 0)

{

if (i>0)

{

system("cls");

gotoxy(80,14);

printf("验证码输入错误你还有%d次机会请重新输入!",i);

i--;

gotoxy(80,15);

system("pause");

}

else

{

MessageBox(NULL,"验证码输入错误你已经没有机会了!即将返回上一级界面","错误", MB\_ICONHAND);

system("cls");

return NULL;

}

goto C;

}

if (strcmp(key1,p1->key) != 0)

{

if (i>0)

{

system("cls");

gotoxy(80,14);

printf("密码输入错误你还有%d次机会请重新输入！",i);

gotoxy(80,15);

system("pause");

system("cls");

i--;

}

else

{

MessageBox(NULL,"密码输入错误你已经没有机会了!即将返回上一级界面","错误", MB\_ICONHAND);

system("cls");

return NULL;

}

goto A;

}

return p1;

}

/\*该代码实现了一个用户登录函数。具体运行步骤：

1. 首先判断链表头指针是否为空，如果为空，则提示“未注册任何账号请注册后重试!”，并返回NULL。
2. 在控制台输出登录界面，要求用户输入账号和密码。
3. 使用scanfPlus函数获取用户输入的账号和密码，其中账号长度为1-18个字符，密码长度也为1-18个字符。如果用户在此过程中输入非法字符或者输入超过规定长度，则会返回NULL。
4. 查找输入的账号是否存在于链表中，如果不存在则提示“无此账号请重新输入!”，并跳转到标签A处让用户重新输入账号和密码。
5. 生成一个长度为4的随机验证码，并输出到控制台上供用户输入。
6. 使用scanfPlus函数获取用户输入的验证码，如果用户在此过程中输入非法字符或者输入超过规定长度，则会返回NULL。
7. 比较用户输入的验证码与生成的验证码是否相同，如果不相同，则提示“验证码输入错误你还有n次机会请重新输入!”，n为剩余的尝试次数，每次输入错误都会减少一次，若此时已经没有尝试次数，则提示“验证码输入错误你已经没有机会了!即将返回上一级界面”，并返回NULL。然后回到标签C处重新进行验证码验证。
8. 如果验证码验证通过，再比较用户输入的密码与链表中对应账号的密码是否相同，如果不相同则提示“密码输入错误你还有n次机会请重新输入!”，n为剩余的尝试次数，每次输入错误都会减少一次，若此时已经没有尝试次数，则提示“密码输入错误你已经没有机会了!即将返回上一级界面”，并返回NULL。然后回到标签A处让用户重新输入账号和密码。
9. 如果账号和密码都验证通过，则返回链表中对应账号的结构体指针p1。\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*将文件中的账户信息读出\*\*\*\*\*\*\*\*\*\*\*\*/

struct ID \*loginRead()

{

FILE \*fp;

char ch;

struct ID \*head = NULL,\*p1,\*p2;

if ((fp = fopen("用户","rb")) == NULL)

return head;

A:p1 = (struct ID \*)malloc(sizeof(struct ID));

fread(p1,sizeof(struct ID),1,fp);

if(feof(fp))

{

fclose(fp);

return head;

}

if (head == NULL)

{

head = p1;

p1->next = NULL;

p2 = p1;

goto A;

}

p2->next = p1;

p1->next = NULL;

p2 = p1;

goto A;

}

/\*这段代码实现了从文件中读取用户信息到一个链表中的功能。

具体步骤如下：

1. 打开名为“用户”的二进制文件，如果打开失败则返回空指针。
2. 利用malloc函数动态分配内存给p1，将文件中的struct ID结构体数据读入p1中。
3. 判断文件是否已经读取完毕，如果是则关闭文件并返回头指针head。
4. 如果链表为空，则将p1赋值给head作为链表的头结点。同时将p1和p2都指向链表末尾，以便后续添加节点。
5. 如果链表不为空，则将p1添加到链表末尾，并将p2指向p1，以便下一轮读取时能够连贯地添加节点。
6. 通过跳转到标记A的方式实现循环读取直到文件结束。
7. 返回读取完毕后的头指针head。\*/

/\*\*\*\*\*\*\*\*\*\*\*\*将账户信息写入文件\*\*\*\*\*\*\*\*\*/

void loginWrite(struct ID \*head)

{

FILE \*fp;

struct ID \*p1;

fp = fopen("用户","wb");

if (head == NULL)

return;

p1 = head;

while (p1 != NULL)

{

fwrite(p1,sizeof(struct ID),1,fp);

p1 = p1->next;

}

fclose(fp);

}

/\*这段代码实现了将账户信息写入文件的功能，主要通过循环遍历链表，将链表中存储的结构体数据写入到二进制文件中。

具体步骤如下：

1. 创建一个名为“用户”的二进制文件，如果创建失败则退出函数。
2. 判断链表是否为空，如果是则直接返回。
3. 将头指针p1指向链表的第一个节点。
4. 遍历整个链表，每次将当前节点p1的struct ID结构体数据写入文件中。
5. 将p1指向下一个节点，以便下一轮循环执行。
6. 关闭文件。\*/

/\*\*\*\*\*\*登录界面菜单\*\*\*\*\*\*\*\*/

void loginMenu()

{

system("cls");

form(6,7);

gotoxy(92,7);

puts("446五星酒店");

gotoxy(93,11);

puts("2.注册");

gotoxy(93,13);

puts("3.注销");

gotoxy(93,15);

puts("4.密码找回");

gotoxy(93,17);

puts("5.修改密码");

gotoxy(93,19);

puts("6.游客登录");

gotoxy(93,21);

puts("0.退出");

gotoxy(70,23);

puts("Tip:在录入数据的过程中按ESC键可直接取消录入返回上一级");

gotoxy(93,9);

red();

puts("1.登录");

}

/\*这段代码实现了一个登录菜单的功能，主要是为用户提供各种登录操作选项，并通过视觉效果提高用户交互体验。

具体步骤如下：

1. 清屏并调用form函数绘制一个菜单框架。
2. 在菜单栏上方输出酒店名称“446五星酒店”。
3. 分别在菜单中输出6个选项：注册、注销、密码找回、修改密码、游客登录和退出。
4. 在屏幕底部输出提示信息：“Tip:在录入数据的过程中按ESC键可直接取消录入返回上一级”。
5. 将选项“1.登录”设为红色字体，突出显示。\*/

/\*\*\*\*\*\*\*\*\*\*\*\*登录主函数\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

struct ID \*login()

{

struct ID \*head = NULL,\*p;

head = loginRead();

int x,i;

char t;

A:system("cls");

loginWrite(head);

loginMenu();

x = 9;

i = 0;

while (1)

{

t = arrows(x,'0','6');

switch(t)

{

case 72:

if (i > 0)

{

x = cColor(93,x,0);

if (x == 9)

puts("1.登录");

else if (x == 11)

puts("2.注册");

else if (x == 13)

puts("3.注销");

else if (x == 15)

puts("4.密码找回");

else if (x == 17)

puts("5.修改密码");

else if (x == 19)

puts("6.游客登录");

i--;

}

else

{

inversion(&i,93,&x,6,21,"0.退出");

}

break;

case 80:

if (i < 6)

{

x = cColor(93,x,1);

if (x == 11)

puts("2.注册");

else if (x == 13)

puts("3.注销");

else if (x == 15)

puts("4.密码找回");

else if (x == 17)

puts("5.修改密码");

else if (x == 19)

puts("6.游客登录");

else

puts("0.退出");

i++;

}

else

{

inversion(&i,93,&x,0,9,"1.登录");

}

break;

case 13: system("cls");

if (x == 9)

{

if ((p = sign(head)))

return p; //登录后返回该账号的基本信息

}

if (x == 11)

head = accountName(head);

if (x == 13)

head = logout(head);

if (x == 15)

signFind(head);

if (x == 17)

changePassword(head);

if (x == 19)

visitor();

if (x == 21)

exit(0);

goto A;

case 49:

system("cls");

ShowCursor();

if ((p = sign(head)))

return p;

goto A;

case 50: system("cls");ShowCursor();head = accountName(head); goto A;

case 51: system("cls");ShowCursor();head = logout(head); goto A;

case 52: system("cls");ShowCursor();signFind(head); goto A;

case 53: system("cls");ShowCursor();changePassword(head); goto A;

case 54: system("cls");ShowCursor();visitor(); goto A;

case 48: exit(0);

}

}

}

/\*这段代码是一个登录功能的主函数。用户可以使用键盘上的上下箭头键选择菜单项，回车键确认选择。具体步骤如下：

1. 首先调用了loginRead()函数读取账号信息，然后进入一个循环，显示登录菜单并等待用户输入选择。
2. 当用户按下上下箭头键时，程序会改变光标所在的位置，并根据光标所在位置打印出相应的菜单项。当用户按下回车键时，程序会执行对应的操作。
3. 各个菜单项的含义：1.登录：用户输入账号和密码进行登录，如果登录成功则返回该账号的基本信息。2.注册：用户注册新账号。3.注销：用户注销已登录的账号。4.密码找回：用户找回忘记的密码。5.修改密码：用户修改已登录账号的密码。6.游客登录：用户以游客身份登录。其中，每个菜单项对应一些子函数，例如登录对应sign()函数，注册对应accountName()函数等等。
4. 当用户选择退出时，程序调用exit()函数结束程序。
5. 如果发生任何错误或用户需要再次选择菜单项，则会跳转回开始处重新显示菜单。\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*修改密码\*\*\*\*\*\*\*\*\*\*/

void changePassword(struct ID \*head)

{

int i = 2;

char accountt[20];

char key1[20];

struct ID \*p1;

printf("请输入账号:");

if (scanfPlus(accountt,8,18,0) == 1)

return;

if (head == NULL)

{

MessageBox(NULL,"无任何账号信息","提示", MB\_ICONASTERISK);

return;

}

p1 = head;

while (p1 != NULL)

{

if (strcmp(accountt,p1->account) == 0)

{

printf("请输入该账号密码:");

A:if (scanfPlus(key1,8,18,1) == 1)

return;

if (strcmp(key1,p1->key) != 0)

{

printf("输入错误!你还有%d次机会请再次输入:",i);

if (i == 0)

{

MessageBox(NULL,"密码输入错误你已经没有机会了!即将返回上一级菜单","错误", MB\_ICONHAND);

return;

}

i--;

goto A;

}

printf("请输入新密码(密码为8~18位):");

B:if (scanfPlus(p1->key,8,18,1) == 1)

return;

printf("请再次输入新密码:");

if (scanfPlus(key1,8,18,1) == 1)

return;

if (strcmp(p1->key,key1) != 0)

{

MessageBox(NULL,"两次输入密码不一致请重新输入！","错误", MB\_ICONHAND);

system("cls");

goto B;

}

return;

}

p1 = p1->next;

}

MessageBox(NULL,"未检索到此账号请检查是否输入正确!","错误", MB\_ICONHAND);

}

/\*这段代码是一个修改账号密码的函数changePassword()。该函数需要传入一个链表头指针head，表示所有账号的信息。

1. 函数会提示用户输入要修改密码的账号和当前密码。如果账号不存在，则会弹出错误提示框；如果当前密码错误，则会提醒用户还剩下几次机会。如果连续三次输入密码错误，则会弹出错误提示框并返回上一级菜单。
2. 如果输入正确，则会提示用户输入新密码，并再次确认。如果两次输入不一致，则会弹出错误提示框并要求重新输入。
3. 在读取用户输入时，使用了scanfPlus()函数对用户的输入进行了检查。如果输入不符合规定，则会提示用户重新输入。如果输入正确，则直接将链表中对应账号的密码修改为新密码。\*/

/\*验证码生成函数声明\*/

char \*randStr(char str[], int n)

{

int i,random;

char strArray[59] = "abcdefghijkmnpqrstuvwxyzABCDEFGHIJKLMNPQRSTUVWXYZ123456789";

srand(time(0));

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

{

random = rand()%58;

str[i] = strArray[random];

}

str[n] = '\0';

return str;

}

/\*这段代码是一个生成随机字符串的函数randStr()。该函数需要传入一个字符数组str和一个整数n，表示需要生成的随机字符串长度。

1. 创建了一个包含了所有可供选择的字符的字符串strArray
2. 使用srand()函数设置随机数种子，使得每次程序运行时生成的随机数序列都不同。
3. 函数进入一个循环，每次从strArray中随机取出一个字符，并将其存储到str[i]中，直到存储了n个字符为止。
4. 函数给str添加一个字符串结束符\0，并返回str数组的首地址。\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*用户端\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int user(struct room \*head,struct ID \*p,struct ID \*head1)

{

struct room \*p1;

int x,i = 0;

char t;

system("cls");

gotoxy(80,14);

if (strcmp(p->sex,"男") == 0)

printf("尊敬的%s先生欢迎您光临446\n",p->name);

else

printf("尊敬的%s女士欢迎您光临446\n",p->name);

gotoxy(80,15);

system("pause");

system("cls");

p1 = head;

while (p1 != NULL)

{

if (strcmp(p1->num,p->num) == 0)

{

remind(p,head1);

break;

}

p1 = p1->next;

}

A:system("cls");

//write(head);

umenu(p);

head = read();

head1 = loginRead();

x = 12;

while (1)

{

t = arrows(x,'0','5');

switch(t)

{

case 72:

if (i > 0)

{

x = cColor(91,x,0);

if (x == 12)

puts("1.订房");

else if (x == 14)

puts("2.退房");

else if (x == 16)

puts("3.查看自己的订房信息");

else if (x == 18)

puts("4.修改自己的订房信息");

else

puts("5.退出登录");

i--;

}

else

{

inversion(&i,91,&x,5,22,"0.退出系统");

}

break;

case 80:

if (i < 5)

{

x = cColor(91,x,1);

if (x == 14)

puts("2.退房");

else if (x == 16)

puts("3.查看自己的订房信息");

else if (x == 18)

puts("4.修改自己的订房信息");

else if (x == 20)

puts("5.退出登录");

else

puts("0.退出系统");

i++;

}

else

{

inversion(&i,91,&x,0,12,"1.订房");

}

break;

case 13: system("cls");

if (x == 12)

reserve(head,p,head1);

else if (x == 14)

{

out(head,p,head1);

loginWrite(head1);

}

else if (x == 16)

userLook(head,p,head1);

else if (x == 18)

userChange(head,p,head1);

else if (x == 20)

return 0;

else

exit(0);

goto A;

case 49: system("cls");reserve(head,p,head1); goto A;

case 50: system("cls");out(head,p,head1);loginWrite(head1); goto A;

case 51: system("cls");userLook(head,p,head1); goto A;

case 52: system("cls");userChange(head,p,head1); goto A;

case 53: return 0;

case 48: exit(0);

}

}

}

/\*这段代码是一个用户端的函数user()，实现了用户使用酒店管理系统的各项功能。该函数需要传入三个参数，分别是房间信息的链表头指针head、登录用户信息的指针p和所有用户信息的链表头指针head1。具体步骤如下：

1. 进行一些欢迎提示，并等待用户按下任意键开始操作。
2. 函数在循环中调用了arrows()函数获取用户输入的方向键，并根据方向键的不同来改变菜单的选中项。
3. 当用户按下回车键时，函数根据选中项的不同调用相应的子函数：如果选中了“订房”，则调用reserve()函数；如果选中了“退房”，调用out()函数；如果选中了“查看自己的订房信息”，则调用userLook()函数；如果选中了“修改自己的订房信息”，则调用userChange()函数；如果选中了“退出登录”，则直接返回0；如果选中了“退出系统”，则直接退出程序。
4. 在每一次子函数调用结束后，函数都会清屏并重新显示用户菜单，等待用户进行下一步操作。\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*用户端菜单\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void umenu(struct ID \*p)

{

struct ID \*head,\*p1;

char name[10];

head = loginRead();

p1 = head;

while (p1 != NULL)

{

if (strcmp(p->account,p1->account) == 0)

break;

p1 = p1->next;

}

if (strcmp(p1->checkOut,"已退房") == 0)

{

/\*gotoxy(80,14);

printf("你提交的退房申请已通过！欢迎你再次下榻本酒店\n\n");\*/

MessageBox(NULL,"你提交的退房申请已通过!欢迎你再次下榻本酒店","提示", MB\_ICONASTERISK);

strcpy(p1->checkOut,"否");

loginWrite(head);

/\*gotoxy(80,15);

system("pause");\*/

system("cls");

}

if (strcmp(p1->checkOut,"退房失败") == 0)

{

gotoxy(80,14);

puts("你的退房申请被驳回了原因如下:");

gotoxy(80,15);

puts(p1->information);

gotoxy(80,19);

printf("如有必要请联系管理员解决相关问题管理员电话为:xxxxxxxxxxx\n\n");

strcpy(p1->checkOut,"否");

loginWrite(head);

gotoxy(80,20);

system("pause");

system("cls");

}

form(9,6);

gotoxy(92,10);

puts("446酒店");

gotoxy(91,14);

puts("2.退房");

gotoxy(91,16);

puts("3.查看自己的订房信息");

gotoxy(91,18);

puts("4.修改自己的订房信息");

gotoxy(91,20);

puts("5.退出登录");

gotoxy(91,22);

puts("0.退出系统");

red();

gotoxy(91,12);

puts("1.订房");

}

/\*这段代码是一个用于显示用户端菜单的函数umenu()。该函数需要传入一个指向登录用户信息的指针p。具体步骤如下：

1. 首先读取所有用户信息，然后根据当前用户的退房状态，在屏幕上输出相应的提示信息。
2. 如果用户提交的退房申请已经被通过，则会弹出提示框；如果申请被驳回，则会在屏幕上显示原因和管理员电话，并等待用户按下任意键继续操作。
3. 函数输出用户菜单，让用户选择需要进行的操作。具体来说，菜单项包括：订房、退房、查看自己的订房信息、修改自己的订房信息、退出登录和退出系统。
4. 在菜单上方还有酒店名称的标识。\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*用户预订房间\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void reserve(struct room \*head,struct ID \*p,struct ID \*head1)

{

struct room \*p1,\*p2;

struct ID \*x1;

char tran[10];

char rNum[10];

int flag = 1,n=0,i,d;

int times[2][13] =

{

{0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31},

{0, 31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31}

};

if (head == NULL)

{

MessageBox(NULL,"本酒店没录入房间信息请稍后重试!","提示", MB\_ICONASTERISK);

return;

}

x1 = head1;

while (x1 != NULL)

{

if (strcmp(x1->account,p->account) == 0)

break;

x1 = x1->next;

}

p1 = head;

while (p1 != NULL)

{

if (strcmp(p1->num,x1->num) == 0)

{

MessageBox(NULL,"你已经预定了酒店房间请勿重复预定!","提示", MB\_ICONASTERISK);

return;

}

n++;

p1 = p1->next;

}

p1 = head;

while (p1 != NULL)

{

if (strcmp(p1->status,"已被预订") == 0)

strcpy(p1->status,"空闲");

p1 = p1->next;

}

puts("请输入你要入住的日期:");

gotoxy(25,0);

printf("年");

gotoxy(29,0);

printf("月");

gotoxy(33,0);

printf("日");

gotoxy(21,0);

if (scanfPlus(tran,4,4,3) == 1)

return;

p->years = p->year = atoi(tran);

gotoxy(27,0);

if (scanfPlus(tran,1,2,3) == 1)

return;

p->month = p->mon = atoi(tran);

gotoxy(31,0);

if (scanfPlus(tran,1,2,3) == 1)

return;

d = p->day = atoi(tran);

B:printf("请输入入住天数:");

if (scanfPlus(tran,1,15,3) == 1)

return;

p->days = p->numDay = atoi(tran);

if (p->days == 0)

{

MessageBox(NULL,"非法输入请重新输入！","错误", MB\_ICONHAND);

system("cls");

goto B;

}

while (1)

{

i = p->years % 4 == 0 && p->years % 100 != 0 || p->years % 400 == 0;

if (p->days + d > times[i][p->month])

{

p->days = p->days + d - times[i][p->month];

if (p->month != 12) p->month++;

else

{

p->years++;

p->month = 1;

}

d = 0;

}

if (d == 0)

break;

else

{

p->days = p->days + d;

break;

}

}

p = tranday(p);

p1 = head;

while (p1 != NULL)

{

x1 = head1;

while (x1 != NULL)

{

if (strcmp(x1->num,p1->num) == 0)

{

if (p->daymin < x1->daymin && p->daymax > x1->daymin)

strcpy(p1->status,"已被预订");

else if (p->daymin > x1->daymin && p->daymax < x1->daymax)

strcpy(p1->status,"已被预订");

else if (p->daymin < x1->daymax && p->daymax >x1->daymax)

strcpy(p1->status,"已被预订");

else if (p->daymin == x1->daymin)

strcpy(p1->status,"已被预订");

}

x1 = x1->next;

}

p1 = p1->next;

}

A:i = form4(n,head);

gotoxy(63,10+2\*i);

printf("请输入你想要的订的房间的房间号:");

if (scanfPlus(rNum,1,9,0) == 1)

return;

p1 = head;

while(p1 != NULL)

{

if (strcmp(p1->num,rNum) == 0 )

{

if (strcmp(p1->status,"空闲") != 0)

{

flag = 2;

break;

}

else

{

flag = 0;

break;

}

}

p1 = p1->next;

}

if (flag == 1)

{

MessageBox(NULL,"未查到此房间信息请重新输入!","提示", MB\_ICONASTERISK);

system("cls");

goto A;

}

if (flag == 2)

{

MessageBox(NULL,"该房间已被预定请重新输入!","提示", MB\_ICONASTERISK);

system("cls");

goto A;

}

x1 = head1;

while (x1 != NULL)

{

if (strcmp(x1->idnum,p->idnum) == 0)

{

strcpy(x1->num,p->num);

x1->year = p->year;

x1->years = p->years;

x1->mon = p->mon;

x1->month = p->month;

x1->day = p->day;

x1->days = p->days;

x1->daymin = p->daymin;

x1->daymax = p->daymax;

x1->numDay = p->numDay;

break;

}

x1 = x1->next;

}

gotoxy(63,12+2\*i);

puts("请填写以下基本信息:");

gotoxy(63,13+2\*i);

printf("姓名:");

if (scanfPlus(x1->inputName,1,9,0) == 1)

return;

gotoxy(63,14+2\*i);

printf("电话:");

if (scanfPlus(x1->inputTel,11,11,3) == 1)

return;

MessageBox(NULL,"订房成功!","提示", MB\_ICONASTERISK);

strcpy(x1->num,rNum);

write(head);

loginWrite(head1);

}

/\*这段代码是一个用于用户预订房间的函数reserve()，该函数需要传入三个参数，分别为房间信息的链表头指针head、登录用户信息的指针p和所有用户信息的链表头指针head1。

具体步骤如下：

1. 判断酒店是否有录入房间信息，如果没有，则会弹出提示框并直接返回。
2. 遍历用户信息链表，查找当前用户是否已经预定了酒店房间，如果已经预定，则会弹出提示框并直接返回。
3. 要求用户输入入住日期、入住天数以及预订房间的房间号，依次进行输入，并将其保存到登录用户信息中。同时，函数计算出用户离店日期，并更新登录用户信息中的相关字段。
4. 遍历房间信息链表，并利用登录用户信息中的预订信息来更新每个房间的状态。具体来说，对于每个房间，检查它在预订日期范围内是否已被预订，如果已被预订，则将房间状态标记为“已被预订”。
5. 函数输出基本信息的填写提示，并要求用户输入姓名和电话。将填写好的预订信息（包括房间号、入住日期、离店日期、预订天数、姓名、电话等）保存到用户信息链表中，并弹出预订成功的提示框。\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*用户退房间\*\*\*\*\*\*\*\*\*\*\*\*\*/

void out(struct room \*head,struct ID \*p,struct ID \*head1)

{

struct room \*p1,\*p2;

struct ID \*x1;

char judge[3];

int cost;

int s = time(0);

int d = s/86400 - 10957;

if (head == NULL)

{

MessageBox(NULL,"管理员还未录入任何房间信息!","提示", MB\_ICONASTERISK);

return;

}

x1 = head1;

while (x1 != NULL)

{

if (strcmp(x1->idnum,p->idnum) == 0)

break;

x1 = x1->next;

}

if (strcmp(x1->checkOut,"是") == 0)

{

MessageBox(NULL,"你已申请退房请等待管理员审核!","提示", MB\_ICONASTERISK);

return;

}

p1 = head;

while (p1 != NULL)

{

if (strcmp(p1->num,x1->num) == 0)

{

strcpy(p1->name,x1->inputName);

strcpy(p1->tel,x1->inputTel);

strcpy(p1->status,"已被预订");

p1->year = x1->year;

p1->mon = x1->mon;

p1->day = x1->day;

p1->days = x1->numDay;

A:printf("请核对你的房间信息是否正确\n");

form2(1);

form3(p1,0);

gotoxy(13,13);

s = time(0);

d = s/86400 - 10956;

cost = atoi(p1->cost);

x1->cost = (d - x1->daymin)\*cost;

if (x1->daymax < d)

printf("你已经超过了你的预定天数，你的住宿费用为%d元",x1->cost);

else if (x1->daymax > d)

printf("你的住宿时间还未到，现在退房住宿费用为%d元",x1->cost);

else printf("你的住宿费用为%d元",x1->cost);

gotoxy(13,14);

if(MessageBox(NULL,"是否退房","询问",MB\_ICONASTERISK|MB\_OKCANCEL)!=IDCANCEL)

{

strcpy(x1->checkOut,"是");

MessageBox(NULL,"申请已提交请等待管理员审核!","提示", MB\_ICONASTERISK);

return;

}

MessageBox(NULL,"已取消退房申请!","提示", MB\_ICONASTERISK);

return;

}

p1 = p1->next;

}

MessageBox(NULL,"你还未预定任何房间!","提示", MB\_ICONASTERISK);

}

/\*此段代码是一个用于用户退房间的函数。该函数需要传入三个参数，分别为房间信息的链表头指针head、登录用户信息的指针p和所有用户信息的链表头指针head1。

具体步骤如下：

1. 判断房间信息是否已经录入，如果没有，则弹出提示框并直接返回。
2. 在用户信息链表中查找当前用户，如果其申请了退房，则弹出提示框并直接返回。
3. 如果当前用户已经预定了酒店房间，函数会遍历房间信息链表，找到与该用户所预定房间相匹配的房间，并将房间信息更新为该用户的相关信息（如姓名、电话等）。
4. 函数输出房间信息填写提示，并要求用户核对房间信息是否正确。同时，函数计算出用户住宿费用，并更新用户信息中的相关字段。
5. 函数要求用户确认是否退房，如果确认，则将用户信息中的“checkOut”字段标记为“是”，表示用户已申请退房并提交审核。\*/

/\*用户查看房间\*/

void userLook(struct room \*head,struct ID \*p,struct ID \*head1)

{

struct ID \*x1;

struct room \*p1;

p1 = head;

x1 = head1;

while (x1 != NULL)

{

if (strcmp(x1->account,p->account) == 0)

break;

x1 = x1->next;

}

p1 = head;

while (p1 != NULL)

{

if (strcmp(x1->num,p1->num) == 0)

break;

p1 = p1->next;

}

if (p1 == NULL)

{

MessageBox(NULL,"你还未预定任何房间,请预定后再查看!","警告", MB\_ICONEXCLAMATION );

return;

}

status(p1,x1);

form2(1);

form3(p1,0);

gotoxy(13,12);

system("pause");

}

/\*此段代码是一个用于用户查看房间的函数。该函数需要传入三个参数，分别为房间信息的链表头指针head、登录用户信息的指针p和所有用户信息的链表头指针head1。

具体步骤如下：

1. 在用户信息链表中查找当前用户所预定的房间号，然后遍历房间信息链表，找到与该房间相匹配的房间信息。
2. 函数输出房间信息，并要求用户按任意键继续。\*/

/\*\*\*\*\*\*用户信息更改\*\*\*\*\*\*\*/

void userChange(struct room \*head,struct ID \*p,struct ID \*head1)

{

int x = 11,i = 0;

char t;

A:system("cls");

loginWrite(head1);

head1 = loginRead();

userChangeMenu();

while (1)

{

t = arrows(x,'0','3');

switch(t)

{

case 72:

if (i>0)

{

x = cColor(90,x,0);

if (x == 11)

puts("1.入住时间和天数");

else if (x == 13)

puts("2.入住客户姓名");

else

puts("3.电话");

i--;

}

else

inversion(&i,90,&x,3,17,"0.返回上一级界面");

break;

case 80:

if (i<3)

{

x = cColor(90,x,1);

if (x == 13)

puts("2.入住客户姓名");

else if (x == 15)

puts("3.电话");

else

puts("0.返回上一级界面");

i++;

}

else

inversion(&i,90,&x,0,11,"1.入住时间和天数");

break;

case 13: system("cls");

if (x == 11)

cInputTime(head,p,head1);

else if (x == 13)

cInputName(head,p,head1);

else if (x == 15)

cUserTel(head,p,head1);

else

return;

goto A;

case 49: system("cls");ShowCursor();cInputTime(head,p,head1); goto A;

case 50: system("cls");ShowCursor();cInputName(head,p,head1);goto A;

case 51: system("cls");ShowCursor();cUserTel(head,p,head1);goto A;

case 48: return;

}

}

}

/\*此段代码是一个用于用户信息更改的函数。该函数需要传入三个参数，分别为房间信息的链表头指针head、登录用户信息的指针p和所有用户信息的链表头指针head1。具体步骤如下：

1. 先输出用户信息更改菜单，并等待用户选择操作。
2. 如果用户选择“入住时间和天数”选项，则调用cInputTime()函数进行信息输入。
3. 如果用户选择“入住客户姓名”选项，则调用cInputName()函数进行信息输入。
4. 如果用户选择“电话”选项，则调用cUserTel()函数进行信息输入。
5. 如果用户选择“返回上一级界面”，则直接返回到上一级菜单。
6. 函数实现采用了使用箭头选择以及颜色反转的方式来进行交互。\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*用户信息更改菜单\*\*\*\*\*\*\*\*/

void userChangeMenu()

{

form(8,4);

gotoxy(87,9);

puts("酒店客户信息更改");

gotoxy(90,13);

puts("2.入住客户姓名");

gotoxy(90,15);

puts("3.电话");

gotoxy(90,17);

puts("0.返回上一级界面");

gotoxy(90,11);

red();

puts("1.入住时间和天数");

}

/\*此函数实现了一个用户信息更改菜单，用于提供给用户进行信息更改的操作选项。具体步骤如下：

1. 函数调用form()函数创建一个空白窗口，用于输出菜单。
2. 函数使用gotoxy()函数将光标移动到菜单中间位置，并输出菜单标题“酒店客户信息更改”。
3. 函数使用gotoxy()函数将光标移动到菜单指定位置，并输出菜单选项，分别为“1.入住时间和天数”、“2.入住客户姓名”、“3.电话”和“0.返回上一级界面”。
4. 函数使用red()函数设置输出文字颜色，将“1.入住时间和天数”选项反色显示，以便提示用户当前选项。\*/

/\*\*\*\*\*\*用户查看预定房间\*\*\*\*\*\*/

struct ID \*userAmend(struct room \*head,struct ID \*p,struct ID \*head1)

{

struct room \*p1;

struct ID \*x1;

int flag = 0;

x1 = head1;

while (x1 != NULL)

{

if (strcmp(x1->idnum,p->idnum) == 0)

break;

x1 = x1->next;

flag = 1;

}

if (flag == 1)

{

MessageBox(NULL,"你还未预定任何房间!","错误",MB\_ICONHAND);

return NULL;

}

p1 = head;

while (p1 != NULL)

{

if (strcmp(p1->num,x1->num) == 0)

{

status(p1,x1);

break;

}

p1 = p1->next;

}

printf("你的订房信息如下");

form2(1);

form3(p1,0);

return x1;

}

/\*此函数主要作用是让用户查看自己预定的房间信息。以下是代码解释：

* 输入参数：三个参数，分别为指向房间链表头的指针head、指向ID链表当前用户节点的指针p和指向ID链表头的指针head1。
* 输出参数：返回值为指向ID链表当前用户节点的指针x1。

具体步骤如下：

1. 遍历ID链表，查找当前用户是否预定了房间。如果已经预定，则跳出循环，并将flag标记为0；否则flag标记为1，表示未预定。
2. 如果flag为1，即未预定任何房间，则弹出提示框"你还未预定任何房间!"，并结束函数，返回NULL。
3. 如果flag为0，在房间链表中查找该用户所预定的房间。遍历整个房间链表，逐个比较每个房间的房间号是否与用户所预定的房间号相同。如果找到了相应的房间，则调用函数status更新该房间的状态；退出循环，进入下一步。
4. 输出该用户所预定的房间信息，包括房间号、房间类型、价格等。\*/

/\*\*\*\*\*\*\*\*\*\*\*用户入住时间修改\*\*\*\*\*\*\*\*\*\*\*/

void cInputTime(struct room \*head,struct ID \*p,struct ID \*head1)

{

struct ID \*x1,\*x2;

struct room \*p1;

char tran[5];

int year,mon,day;

int years,month,days,numDay;

int daymin = 0,daymax = 0;

int i;

int leap;

int times[2][13] =

{

{0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31},

{0, 31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31}

};

if ((x1 = userAmend(head,p,head1)) == NULL)

return;

gotoxy(13,11);

printf("请输入修改后的入住时间:");

gotoxy(41,11);

printf("年");

gotoxy(46,11);

printf("月");

gotoxy(50,11);

printf("日");

gotoxy(37,11);

if (scanfPlus(tran,4,4,3) == 1)

return;

years = year = atoi(tran);

gotoxy(43,11);

if (scanfPlus(tran,1,2,3) == 1)

return;

month = mon = atoi(tran);

gotoxy(48,11);

if (scanfPlus(tran,1,2,3) == 1)

return;

days = day = atoi(tran);

A:printf("请输入修改后天天数:");

if (scanfPlus(tran,1,9,3) == 1)

return;

numDay = atoi(tran);

if (numDay == 0)

{

MessageBox(NULL,"非法输入请重新输入！","错误", MB\_ICONHAND);

system("cls");

goto A;

}

while (1)

{

i = years % 4 == 0 && years % 100 != 0 || years % 400 == 0;

if (days + numDay > times[i][p->month])

{

days = days + numDay - times[i][p->month];

if (month != 12) month++;

else

{

years++;

month = 1;

}

numDay = 0;

}

if (numDay == 0)

break;

else

{

days = days + numDay;

break;

}

}

for (i = 2000; i < p->year; i++)

{

if (i % 4 == 0 && i % 100 != 0 || i % 400 == 0)

daymin += 366;

else

daymin += 365;

}

leap = year % 4 == 0 && year % 100 != 0 || year % 400 == 0;

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

daymin += times[leap][i];

daymin += day;

for (i = 2000; i < years; i++)

{

if (i % 4 == 0 && i % 100 != 0 || i % 400 == 0)

daymax += 366;

else

daymax += 365;

}

leap = years % 4 == 0 && years % 100 != 0 || years % 400 == 0;

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

daymax += times[leap][i];

daymax += days;

p1 = head;

while (p1 != NULL)

{

x2 = head1;

while (x2 != NULL)

{

if (strcmp(x2->num,p1->num) == 0)

{

if (daymin < x2->daymin && daymax > x2->daymin && x2->num != x1->num)

{

MessageBox(NULL,"修改失败该时间段房间已被预定!","错误",MB\_ICONHAND);

return;

}

else if (daymin > x1->daymin && daymax < x2->daymax && x2->num != x1->num)

{

MessageBox(NULL,"修改失败该时间段房间已被预定!","错误",MB\_ICONHAND);

return;

}

else if (daymin < x2->daymax && daymax >x2->daymax && x2->num != x1->num)

{

MessageBox(NULL,"修改失败该时间段房间已被预定!","错误",MB\_ICONHAND);

return;

}

else if (daymin == x2->daymin && x2 != p && x2->num != x1->num)

{

MessageBox(NULL,"修改失败该时间段房间已被预定!","错误",MB\_ICONHAND);

return;

}

}

x2 = x2->next;

}

p1 = p1->next;

}

x1->year = year;

x1->mon = mon;

x1->day = day;

x1->numDay = numDay;

x1->daymin = daymin;

x1->daymax = daymax;

MessageBox(NULL,"修改成功!","提示",MB\_ICONASTERISK);

}

/\*此函数实现的是用户修改入住时间的功能并进行了合理性检查，确保用户所选时间未被其他用户占用。具体解释如下：

* 定义了两个指针变量x1和x2，分别指向ID链表中当前用户和所有用户。
* 用户进行输入操作，包括年月日以及入住天数，更新指向当前用户的节点p。
* 根据用户输入的年月日和入住天数计算出新的待预定时间段daymin和daymax。
* 遍历整个房间链表，对于每个房间，在ID链表中查找是否有其他人在该时间段内预定该房间。如果发现已经被预定，则结束函数并弹出提示框；否则继续遍历房间链表。
* 如果没有被预定，则更新当前用户节点的年月日和入住天数，并重新计算daymin和daymax。
* 弹出提示框"修改成功!"，函数结束。\*/

/\*\*\*\*\*\*\*\*\*\*用户入住姓名修改\*\*\*\*\*\*\*\*\*\*\*/

void cInputName(struct room \*head,struct ID \*p,struct ID \*head1)

{

struct ID \*x1;

if ((x1 = userAmend(head,p,head1)) == NULL)

return;

gotoxy(13,11);

printf("请输入修改后的入住客户姓名:");

if (scanfPlus(x1->inputName,1,9,0) == 0)

return;

MessageBox(NULL,"修改成功!","提示",MB\_ICONASTERISK);

}

/\*此函数实现的是用户修改入住姓名的功能，确保用户输入合法并进行了相应的信息更新操作。具体解释如下：

* 首先调用函数userAmend，根据当前用户的ID查找是否已经预定房间。
* 如果该用户未预定任何房间，则结束函数并弹出提示框"你还未预定任何房间!"；否则继续执行后面的代码。
* 通过调用scanfPlus函数，让用户输入新的客户姓名，并更新指向当前用户节点x1的inputName成员变量。
* 弹出提示框 "修改成功!" ，函数结束。\*/

/\*\*\*\*\*\*\*\*\*\*用户入住电话修改\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void cUserTel(struct room \*head,struct ID \*p,struct ID \*head1)

{

struct ID \*x1;

if ((x1 = userAmend(head,p,head1)) == NULL)

return;

gotoxy(13,11);

printf("请输入修改后的电话:");

if (scanfPlus(x1->inputName,1,9,0) == 0)

return;

MessageBox(NULL,"修改成功!","提示",MB\_ICONASTERISK);

}

/\*此函数实现的是用户修改入住电话号码的功能，确保用户输入合法并进行了相应的信息更新操作。具体解释如下：

* 首先调用函数userAmend，根据当前用户的ID查找是否已经预定房间。
* 如果该用户未预定任何房间，则结束函数并弹出提示框"你还未预定任何房间!"；否则继续执行后面的代码。
* 通过调用scanfPlus函数，让用户输入新的电话号码，并更新指向当前用户节点x1的inputName成员变量。
* 弹出提示框 "修改成功!" ，函数结束。\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*游客端\*\*\*\*\*\*\*\*\*\*\*\*\*/

void visitor()

{

int x,i;

char t;

A:system("cls");

x = 12;

i = 0;

viewMenu();

while (1)

{

t = arrows(x,'0','2');

switch(t)

{

case 72:

if (i>0)

{

x = cColor(89,x,0);

if (x == 12)

puts("1.查看房间信息");

else

puts("2.登录");

i--;

}

else

inversion(&i,89,&x,2,16,"0.退出系统");

break;

case 80:

if (i<2)

{

x = cColor(89,x,1);

if (x == 14)

puts("2.登录");

else

puts("0.退出系统");

i++;

}

else

inversion(&i,89,&x,0,12,"1.查看房间信息");

break;

case 13: system("cls");

if (x == 12)

viewLook();

else if (x == 14)

return;

else

exit(0);

goto A;

case 49: system("cls");ShowCursor();viewLook(); goto A;

case 50: return;

case 48: exit(0);

}

}

}

/\*此函数实现的是一个游客端的菜单功能，可以让用户选择查看房间信息、登录或退出系统。具体解释如下：

* 定义了变量x和i，分别代表当前选中的菜单项和其在菜单列表中的序号。
* 调用viewMenu函数显示游客端的菜单界面。
* 使用arrows函数获取用户按键输入，并根据不同的按键进行相应的操作。
* 如果用户按上下箭头，则根据当前选择的菜单项向上或向下移动，同时更新x和i的值。
* 如果用户按回车键，则根据当前选择的菜单项进行相应的操作：
* 如果选择"查看房间信息"，则调用viewLook函数查看房间信息，并重新进入游客端菜单界面；
* 如果选择"登录"，则结束该函数，返回上一级菜单（例如登录界面）；
* 如果选择"退出系统"，则直接退出整个程序。\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*游客菜单\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void viewMenu()

{

system("cls");

form(9,3);

gotoxy(86,10);

puts("亲！欢迎光临本酒店");

gotoxy(89,14);

puts("2.登录");

gotoxy(89,16);

puts("0.退出系统");

gotoxy(89,12);

red();

puts("1.查看房间信息");

}

/\*此函数实现的是游客端的菜单界面显示功能，让用户可以方便地选择不同的菜单选项。具体解释如下：

* 调用system函数，清空屏幕上的所有内容。
* 使用form函数在屏幕上绘制一个矩形框架，并将光标移动到矩形框架中央。
* 使用gotoxy函数将光标移动到指定位置，输出欢迎语和菜单选项。
* 使用red函数设置字体颜色为红色，并将光标移动到第一行菜单选项上，输出"查看房间信息"选项。
* 函数执行完毕，返回上一级调用函数。\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*游客端房间信息查看\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void viewLook()

{

int i,n = 0;

int s = time(0);

int d = s/86400 - 10957;

struct ID \*head1,\*x1,\*x2;

struct room \*head,\*p1,\*p2;

head = read();

head1 = loginRead();

if (head == NULL)

{

system("cls");

gotoxy(80,14);

printf("无任何房间信息!");

gotoxy(80,15);

system("pause");

return;

}

gotoxy(100,1);

printf("浏览房间信息");

p2 = head;

while (p2 != NULL)

{

x1 = head1;

while (x1 != NULL)

{

if (x1->daymin <= d && x1->daymax > d && strcmp(x1->num,p2->num) == 0)

{

strcpy(p2->status,"已被预订");

break;

}

x1 = x1->next;

}

n++;

p2 = p2->next;

}

i = form4(n,head);

gotoxy(63,10+2\*i);

p2 = head;

while (p2 != NULL)

{

strcpy(p2->status,"空闲");

p2 = p2->next;

}

system("pause");

}

/\*此函数实现的是游客端查看房间信息的功能，包括将所有房间的状态更新为"已被预订"或"空闲"、在屏幕上显示所有房间的详细信息。具体解释如下：

* 定义了变量i、n、s和d，分别代表当前选中的行数、房间总数、当前系统时间和以天为单位计算的系统时间。
* 定义了指针变量head、p1、p2、head1、x1和x2，分别代表房间链表头结点、遍历房间链表时的当前节点、用户ID链表头结点、遍历用户ID链表时的当前节点。
* 调用read函数读取房间信息并将其存储在房间链表中；调用loginRead函数读取用户ID信息并将其存储在用户ID链表中。
* 如果房间链表为空，则直接输出"无任何房间信息!"并等待用户按任意键结束函数。
* 使用gotoxy函数将光标移动到屏幕中央，输出"浏览房间信息"字样。
* 遍历整个房间链表，并在ID链表中查找是否有人预定该房间。如果该房间已被预定，则将该房间的状态更新为"已被预订"；否则状态更新为"空闲"。
* 根据房间链表中的内容，使用form4函数在屏幕上显示所有房间的详细信息。
* 等待用户按任意键结束函数。\*/

/\*\*\*\*\*\*\*\*\*\*\*改进版输入函数\*\*\*\*\*\*\*\*\*\*\*\*/

int scanfPlus(char \*st,int m,int n,int x) //st该字符串变量名，m代表该字符串最短长度，n代表该字符串最长长度,x代表以什么形式显示字符串

{

int i=0;

char ch;

while(1)

{

ch = getch();

if (ch == 27) // 判断输入是否为ESC键

return 1;

if (ch == 13) //判断输入是否为回车键

{

if (i>m-1)

{

st[i] = '\0';

printf("\n");

return 0;

}

continue;

}

if (ch == 8) //判断输入是否为BACKSPACE键

{

if (i>0)

{

if (st[i-1]< 0) //判断输入BACKSPACE键前一个键是否为汉字

{

printf("\b\b \b\b"); //汉字占两个字节

i = i-2;

}

else

{

printf("\b \b");

i--;

}

}

continue;

}

if (ch == -32)

{

getch();

continue;

}

if (ch == 9)

continue;

if (ch < 0 && x == 1) //当x=1时只能输入字符不能输入汉字

continue;

if (x == 3) //当x=3时只能输入数字

if (ch < 47 || ch >58)

continue;

if (i<n)

{

if (x == 0 || x == 3) printf("%c",ch);

else printf("\*"); //当x=1 是用"\*"表示输入

st[i] = ch;

i++;

}

else

continue;

}

}

/\*此函数实现的是一个改进版的输入函数scanfPlus，该函数可以控制用户输入字符串的长度、内容和显示形式，并且可以防止用户输入非法字符。具体解释如下：

* 定义了变量i和ch，分别代表当前字符串的长度和用户输入的字符。
* 使用while循环，不断读取用户输入的字符，根据不同的字符进行相应的操作。
* 如果用户按下ESC键，则返回值1，表示用户取消了输入。
* 如果用户按下回车键，则判断当前字符串的长度是否达到最小值m，如果达到则结束函数并返回值0；否则继续等待用户输入。
* 如果用户按下BACKSPACE键，则将字符串长度减1，并在屏幕上删除最后一个字符（如果该字符是汉字则删除两个字符），然后继续等待用户输入。
* 如果用户按下TAB键，则忽略该输入字符，继续等待用户输入。
* 如果x=1且用户输入的字符为汉字，则忽略该输入字符，继续等待用户输入。
* 如果x=3且用户输入的字符不是数字，则忽略该输入字符，继续等待用户输入。
* 如果当前字符串长度没有达到最大值n，则将该字符添加到字符串末尾，并在屏幕上显示该字符（如果x=0或x=3则直接显示该字符，如果x=1则用"\*"代替该字符）。
* 如果当前字符串长度已经达到最大值n，则忽略该输入字符，继续等待用户输入。
* 函数执行完毕，返回0或1表示用户是否成功输入字符串。\*/

/\*\*\*\*\*\*\*\*\*\*\*\*隐藏光标\*\*\*\*\*\*\*\*\*\*/

void HideCursor()

{

CONSOLE\_CURSOR\_INFO cursor; //Windows.h头文件内置结构体

cursor.bVisible = FALSE; //不显示光标(TRUE为显示、FALSE为不显示)

cursor.dwSize = sizeof(cursor); //光标的高度，控制台一行字符的高度为100，光标的高度为1到100

HANDLE handle = GetStdHandle(STD\_OUTPUT\_HANDLE); //获取标准输出(控制台)句柄

SetConsoleCursorInfo(handle, &cursor); //让标准输出(控制台)隐藏光标

}

/\*此C++函数目的是隐藏控制台窗口内的光标。代码中主要使用了Windows.h头文件中提供的结构体和API函数。

首先定义了一个名为cursor的结构体变量，该结构体是CONSOLE\_CURSOR\_INFO类型，表示控制台窗口光标信息。然后将bVisible成员变量设置为FALSE，表示不显示光标。接着通过GetStdHandle(STD\_OUTPUT\_HANDLE)获取标准输出(控制台)句柄，并将该句柄和cursor结构体作为参数传给SetConsoleCursorInfo函数，从而实现隐藏光标的功能。需要注意的是，该函数只能在Windows操作系统下运行。\*/

/\*\*\*\*\*\*\*\*\*\*\*\*显示光标\*\*\*\*\*\*\*\*\*\*\*/

void ShowCursor()

{

CONSOLE\_CURSOR\_INFO cursor; //Windows.h头文件内置结构体

cursor.bVisible = TRUE; //不显示光标(TURE为显示、FALSE为不显示)

cursor.dwSize = sizeof(cursor); //光标的高度，控制台一行字符的高度为100，光标的高度为1到100

HANDLE handle = GetStdHandle(STD\_OUTPUT\_HANDLE); //获取标准输出(控制台)句柄

SetConsoleCursorInfo(handle, &cursor); //让标准输出(控制台)隐藏光标

}

/\*此C++函数的作用是在控制台窗口中显示光标，方便用户输入和交互操作。具体实现步骤如下：

* 定义了一个Windows.h头文件内置的CONSOLE\_CURSOR\_INFO结构体变量cursor，用于控制光标的显示和大小。
* 将cursor.bVisible设置为TRUE，表示要显示光标。
* 设置cursor.dwSize为sizeof(cursor)，表示光标的高度与CONSOLE\_CURSOR\_INFO结构体的大小相同。
* 通过GetStdHandle(STD\_OUTPUT\_HANDLE)函数获取标准输出（即控制台）句柄。
* 最后通过调用SetConsoleCursorInfo(handle, &cursor)函数将标准输出（控制台）的光标信息设置为所定义的cursor，以此来显示光标。\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*加载进度条\*\*\*\*\*\*\*\*\*\*/

void load()

{

float p; //进度条刻度

float progLen; //程序总耗时

int i, j;

progLen = 15.0;

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

{

system ("cls");

p = (i/progLen)\*LEN; //计算程序当前进度之于进度条的比例

gotoxy(80,10);

for (j=1; j<=(int)(p+0.5); j++) //打印已执行刻度

printf ("■");

for (j=1; j<=LEN-(int)(p+0.5); j++) //打印待执行刻度

{

printf ("□");

HideCursor();

}

printf (" %.1f%%", (i/progLen)\*100); //计算程序当前进度的百分比

Sleep(200);

}

// putchar ('\n');

gotoxy(86,12);

printf("加载完成");

Sleep(150);

// getch ();

return;

}

/\*此函数的作用是在控制台窗口中实现一个简单的进度条，帮助用户更好地了解程序的运行状态和进度。具体实现步骤如下：

* 定义了两个变量p和progLen，其中p表示当前进度条的刻度，progLen表示程序总耗时。
* 在for循环中，通过system("cls")函数清空控制台屏幕，并计算当前程序运行进度之于进度条总长度的比例，将结果赋值给p。
* 通过gotoxy()函数将光标移动到指定位置，循环打印已执行的刻度和待执行的刻度。
* 通过printf()函数打印当前进度的百分比，使用Sleep()函数让程序暂停200毫秒，以便用户能够观察进度条的变化。
* 当程序结束运行后，输出“加载完成”的提示信息，并让程序暂停150毫秒。\*/

/\*\*\*\*\*\*\*\*\*\*\*\*菜单表格打印\*\*\*\*\*\*\*\*\*\*/

void form(int x,int n)

{

int i;

gotoxy(80,x);

puts("┌──────────────────────────────┐ ");

gotoxy(80,x+1);

puts("│ │ ");

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

{

gotoxy(80,x+2+2\*i);

puts("├──────────────────────────────┤ ");

gotoxy(80,x+3+2\*i);

puts("│ │");

}

gotoxy(80,x+2+2\*i);

puts("└──────────────────────────────┘");

}

/\*此函数的作用是在控制台窗口中打印一个简单的、具有分割线的菜单表格，帮助用户更好地了解菜单选项和功能。具体实现步骤如下：

* 在form函数中通过调用gotoxy()函数将光标移动到指定位置，打印菜单表格的上边框。
* 接着通过调用puts()函数打印菜单表格的第一行，并将光标移到下一行以便后面继续添加内容。
* 通过for循环，循环次数为n，每次循环都将光标移动到相应的位置并打印菜单表格中间的分割线和空白行。
* 最后再通过调用gotoxy()和puts()函数将光标移到菜单表格底部，打印菜单表格的下边框。\*/

/\*\*\*\*\*\*\*\*\*房间完整信息表格打印\*\*\*\*\*\*\*\*\*\*\*\*/

void form2(int n)

{

int i;

gotoxy(13,5);

puts("┌───────────────┬────────────────┬───────────────┬───────────────┬───────────────┬───────────────┬───────────────┬───────────────┬───────────────┬───────────────┐");

gotoxy(13,6);

puts("│ │ │ │ │ │ │ │ │ │ │");

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

{

gotoxy(13,7+2\*i);

puts("├───────────────┼────────────────┼───────────────┼───────────────┼───────────────┼───────────────┼───────────────┼───────────────┼───────────────┼───────────────┤");

gotoxy(13,8+2\*i);

puts("│ │ │ │ │ │ │ │ │ │ │");

}

gotoxy(13,7+2\*i);

puts("└───────────────┴────────────────┴───────────────┴───────────────┴───────────────┴───────────────┴───────────────┴───────────────┴───────────────┴───────────────┘");

gotoxy(19,6);

printf("房间号");

gotoxy(36,6);

printf("房型");

gotoxy(53,6);

printf("位置");

gotoxy(67,6);

printf("房间状态");

gotoxy(85,6);

printf("入住");

gotoxy(99,6);

printf("入住时间");

gotoxy(117,6);

printf("天数");

gotoxy(133,6);

printf("费用(天)");

gotoxy(147,6);

printf("入住客户");

gotoxy(165,6);

printf("电话");

}

/\*此函数的作用是在控制台窗口中打印一个具有分割线和标题的房间完整信息表格，帮助用户更好地了解房间的基本信息和入住情况。具体实现步骤如下：

* 通过调用gotoxy()函数将光标移动到指定位置，打印房间完整信息表格的上边框。
* 接着通过调用puts()函数打印表格第一行和第二行，并将光标移到下一行以便后面继续添加内容。
* 通过for循环，循环次数为n，每次循环都将光标移动到相应的位置并打印表格中间的分割线和空白行。
* 最后再通过调用gotoxy()和puts()函数将光标移到房间完整信息表格底部，打印表格下边框。
* 在表格中添加了一些列标题，例如房间号、房型、位置等等，方便用户查看信息。\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*房间完整信息打印\*\*\*\*\*\*\*\*\*\*\*/

void form3(struct room \*p1,int i)

{

gotoxy(19,8+2\*i);

printf("%s",p1->num);

gotoxy(34,8+2\*i);

printf("%s",p1->type);

gotoxy(52,8+2\*i);

printf("%s",p1->site);

gotoxy(68,8+2\*i);

printf("%s",p1->status);

gotoxy(85,8+2\*i);

printf("%s",p1->check);

gotoxy(96,8+2\*i);

if (p1->year == 1)

printf("无");

else

{

printf("%d",p1->year);

fputs("-",stdout);

printf("%d",p1->mon);

fputs("-",stdout);

printf("%d",p1->day);

}

gotoxy(115,8+2\*i);

printf("%d",p1->days);

gotoxy(133,8+2\*i);

printf("%s",p1->cost);

gotoxy(147,8+2\*i);

printf("%s",p1->name);

gotoxy(161,8+2\*i);

printf("%s",p1->tel);

}

/\*此函数的作用是在控制台窗口中打印一个房间的完整信息，帮助用户更全面地了解每个房间的基本情况和入住情况，并方便管理人员进行查询和管理。具体实现步骤如下：

* 通过调用gotoxy()函数将光标移动到指定位置，分别打印房间号、房型、位置、房间状态、入住情况等信息。
* 如果该房间没有被入住，则入住时间显示为“无”，否则将入住时间按照年月日的格式打印出来。
* 打印该房间的入住天数、费用、入住客户和电话等详细信息。\*/

/\*\*\*\*\*\*房间部分信息表格及数据打印\*\*\*\*\*\*/

int form4(int n,struct room \*head)

{

struct room \*p2;

int i;

gotoxy(63,5);

puts("┌───────────────┬────────────────┬───────────────┬───────────────┬───────────────┐");

gotoxy(63,6);

puts("│ │ │ │ │ │");

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

{

gotoxy(63,7+2\*i);

puts("├───────────────┼────────────────┼───────────────┼───────────────┼───────────────┤");

gotoxy(63,8+2\*i);

puts("│ │ │ │ │ │");

}

gotoxy(63,7+2\*i);

puts("└───────────────┴────────────────┴───────────────┴───────────────┴───────────────┘");

gotoxy(69,6);

printf("房间号");

gotoxy(86,6);

printf("房型");

gotoxy(103,6);

printf("位置");

gotoxy(117,6);

printf("房间状态");

gotoxy(135,6);

printf("费用(天)");

p2 = head;

i = 0;

while (p2 != NULL)

{

gotoxy(70,8+2\*i);

printf("%s",p2->num);

gotoxy(85,8+2\*i);

printf("%s",p2->type);

gotoxy(103,8+2\*i);

printf("%s",p2->site);

gotoxy(117,8+2\*i);

printf("%s",p2->status);

gotoxy(135,8+2\*i);

printf("%s",p2->cost);

p2 = p2->next;

i++;

}

return i;

}

/\*此函数的作用是在控制台窗口中打印一个具有分割线和标题的房间部分信息表格和数据，帮助用户更好地了解房间的基本信息。具体实现步骤如下：

* 通过调用gotoxy()函数将光标移动到指定位置，打印房间部分信息表格的上边框。
* 接着通过调用puts()函数打印表格第一行和第二行，并将光标移到下一行以便后面继续添加内容。
* 通过for循环，循环次数为n，每次循环都将光标移动到相应的位置并打印表格中间的分割线和空白行。
* 最后再通过调用gotoxy()和puts()函数将光标移到房间部分信息表格底部，打印表格下边框。
* 在表格中添加了一些列标题，例如房间号、房型、位置等等，方便用户查看信息。
* 将链表中存储的房间信息逐个输出至表格中，方便用户查看每个房间的基本情况。
* 返回值为i，即链表中房间的数量。\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*将日期转化为天数\*\*\*\*\*\*\*\*\*\*/

struct ID \*tranday(struct ID \*p)

{

int i;

int leap;

int times[2][13] =

{

{0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31},

{0, 31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31}

};

p->daymin = 0;

p->daymax = 0;

for (i = 2000; i < p->year; i++)

{

if (i % 4 == 0 && i % 100 != 0 || i % 400 == 0)

p->daymin += 366;

else

p->daymin += 365;

}

leap = p->year % 4 == 0 && p->year % 100 != 0 || p->year % 400 == 0;

for (i = 1; i < p->mon; i++)

p->daymin += times[leap][i];

p->daymin += p->day;

for (i = 2000; i < p->years; i++)

{

if (i % 4 == 0 && i % 100 != 0 || i % 400 == 0)

p->daymax += 366;

else

p->daymax += 365;

}

leap = p->years % 4 == 0 && p->years % 100 != 0 || p->years % 400 == 0;

for (i = 1; i < p->month; i++)

p->daymax += times[leap][i];

p->daymax += p->days;

return p;

}

/\*此函数可以将输入的日期转化为天数，方便进行日期计算，例如求出两个日期之间的间隔天数等等。具体实现步骤如下：

* 定义一个二维数组times，其中第一行为平年每个月的天数，第二行为闰年每个月的天数。
* 初始化p->daymin和p->daymax为0，分别表示起始日期和结束日期的天数。
* 通过for循环，循环次数为2000年到输入日期所在年份之间的年数，每次循环都判断该年是否为闰年，并累加相应的天数。
* 再通过for循环，循环次数为1到输入日期所在月份之前的月份数，每次循环都累加相应的天数。
* 最后再加上输入日期所在月份的天数和日期，即可得到该日期的总天数（daymin）。
* 同理，对于结束日期也进行上述操作，最后得到结束日期的总天数(daymax)。
* 返回值为结构体指针p，其中p->daymin和p->daymax属性存储了转化后的天数。\*/

/\*\*\*\*\*\*\*\*\*\*\*房间信息覆盖\*\*\*\*\*\*\*\*\*\*\*\*\*/

void status(struct room \*p1,struct ID \*x1)

{

strcpy(p1->status,"已被预订");

strcpy(p1->name,x1->inputName);

strcpy(p1->tel,x1->inputTel);

p1->year = x1->year;

p1->mon = x1->mon;

p1->day = x1->day;

p1->days = x1->numDay;

}

/\*此函数可以将用户输入的预订信息覆盖到指定房间中，标记该房间已被预定，并更新相应的房间信息，方便管理人员进行信息维护和查询。具体实现步骤如下：

* 通过调用strcpy()函数将用户输入的姓名和电话号码分别覆盖到房间信息结构体p1的name和tel属性上。
* 将用户输入的年、月、日和预定天数分别赋值给房间信息结构体p1的year、mon、day和days属性上。
* 将房间信息结构体p1的status属性修改为“已被预订”，标记该房间已被预定。
* 经过以上操作，该房间的基本信息和入住情况已经得到了更新，方便管理人员进行查看和管理。\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*重置房间信息\*\*\*\*\*\*\*\*\*\*\*\*/

void reset(struct room \*head)

{

struct room \*p2;

p2 = head;

while (p2 != NULL)

{

strcpy(p2->status,"空闲");

strcpy(p2->name,"无");

strcpy(p2->tel,"无");

strcpy(p2->name,"无");

p2->year = 1;

p2->days = 0;

p2 = p2->next;

}

}

/\*此函数作用是将所有房间的信息重置为默认值，即状态修改为空闲，客户姓名和电话号码修改为“无”，入住时间和天数修改为默认值。具体实现步骤如下：

* 定义一个房间信息指针p2，并将其初始化为链表头。
* 通过while循环，循环遍历整个房间信息链表，每次循环都对当前节点的各个属性进行重置操作。
* 将房间信息结构体p2的status属性修改为“空闲”，标记该房间状态为空闲。
* 将房间信息结构体p2的name和tel属性修改为“无”，表示该房间目前没有客户入住。
* 将房间信息结构体p2的year属性修改为1，表示该房间目前没有客户入住。
* 将房间信息结构体p2的days属性修改为0，表示该房间目前没有客户入住。
* 通过p2->next将指针p2指向下一个节点，继续执行循环，直到遍历完整个链表中的房间信息节点。

经过以上操作，所有房间的基本信息和入住情况已经被重置为默认值，方便管理人员进行信息维护和查询。\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*打印箭头\*\*\*\*\*\*\*\*\*/

int arrows(int x,char min,char max)

{

char t;

HideCursor();

gotoxy(84,x);

blue();

printf("->");

while (t = getch())

{

if (t >= min && t <= max || t == 72 || t == 80 || t == 13)

break;

}

system("color F1");

gotoxy(84,x);

printf(" ");

return t;

}

/\*此函数的作用是在控制台窗口中打印一个箭头，并且根据用户的输入控制箭头的移动，方便用户进行选择和操作。具体实现步骤如下：

* 定义变量t用于存储用户的输入。
* 通过调用HideCursor()函数隐藏光标，避免干扰打印输出。
* 通过调用gotoxy()函数将光标移动到指定位置，即箭头所在位置。
* 通过调用blue()函数将字体颜色设置为蓝色，以区分箭头和其他内容。
* 调用printf()函数打印箭头“->”。
* 通过while循环监听用户的输入，只有在输入字符在[min,max]范围内，或者是上、下、回车键（对应ASCII码分别为72、80、13）时，才结束监听并跳出循环。
* 调用system()函数将字体颜色恢复为白底黑字。
* 通过调用gotoxy()函数将光标移回箭头所在位置。
* 调用printf()函数将箭头覆盖为两个空格，以便重新输出新的箭头。
* 返回值为用户输入的字符t。\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*随箭头变化数据同步变化\*\*\*\*\*\*\*\*\*\*\*/

int cColor(int y,int x,int judge)

{

if (judge == 0)

x -= 2;

else

x += 2;

gotoxy(y,x);

red();

return x;

}

/\*此函数的作用是在控制台窗口中通过调整箭头的位置来同步改变其他数据的值，并且设置相应的字体颜色以增加可读性。具体实现步骤如下：

* 定义变量y、x和judge，其中y表示箭头所在行号，x表示箭头当前所在列号，judge表示箭头向左还是向右移动（0为向左移动，1为向右移动）。
* 判断judge的值，如果为0，则将x减去2，即箭头向左移动两个位置；否则将x加上2，即箭头向右移动两个位置。
* 通过调用gotoxy()函数将光标移动到指定位置，即箭头移动后所在的位置。
* 通过调用red()函数将字体颜色设置为红色，以区分被选中的数据和其他内容。
* 返回值为箭头移动后所在的列号x，以便对其他数据进行同步更改。\*/

/\*\*\*\*\*\*\*\*\*箭头方向倒置\*\*\*\*\*\*\*\*\*\*\*\*/

void inversion(int \*i,int y,int \*x,int a,int b,char \*s)

{

\*i = a;

\*x = b;

gotoxy(y,\*x);

red();

puts(s);

}

/\*此函数作用是将箭头的方向倒置，并且在新位置打印指定字符数组s，方便用户进行选择和操作。具体实现步骤如下：

* 定义变量i、y、x、a、b和\*s，其中i表示当前选中的数据索引，y表示箭头所在行号，x表示箭头所在列号，a和b表示箭头移动后所在的新行列号，\*s表示需要打印的字符数组。
* 将i赋值为a，将x赋值为b，即更新当前选中的数据索引和箭头的位置。
* 通过调用gotoxy()函数将光标移动到指定位置，即箭头移动后所在的位置。
* 通过调用red()函数将字体颜色设置为红色，以区分被选中的数据和其他内容。
* 调用puts()函数打印指定的字符数组s。\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*提醒用户退房\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void remind(struct ID \*p,struct ID \*head1)

{

struct ID \*x1;

int s = time(0);

int d = s/86400 - 10956;

x1 = head1;

while (x1 != NULL)

{

if (strcmp(x1->idnum,p->idnum) == 0)

break;

x1 = x1->next;

}

if (d > x1->daymax)

MessageBox(NULL,"你预定的退房时间到了请办理退房手续以免造成不必要的损失或者尝试修改预定天数!","提示",MB\_ICONASTERISK);

}

/\*此函数的作用是在用户预订的退房日期到来前提醒用户办理退房手续，以免造成不必要的损失。具体实现步骤如下：

* 定义结构体指针p和head1，其中p表示当前用户输入的预订信息，head1表示所有预订信息的链表头。
* 通过调用time()函数获取当前时间戳，并将其转化为当天的总天数d，方便计算到达退房日期还剩余的天数。
* 定义结构体指针x1，并将其初始化为链表头。
* 通过while循环，遍历整个预订信息链表，每次循环都判断当前节点的身份证号是否与用户输入的身份证号相同，如果相同就跳出循环。
* 判断d是否大于该用户预订的退房日期对应的总天数daymax，如果大于则说明已经到了退房日，需要提醒用户办理退房手续。
* 调用MessageBox()函数向用户发送提醒消息，提醒用户办理退房手续或者修改预定天数。\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*字体颜色变为蓝色\*\*\*\*\*\*\*\*\*\*\*\*\*/

void blue()

{

SetConsoleColor(FOREGROUND\_BLUE | BACKGROUND\_BLUE | BACKGROUND\_GREEN | BACKGROUND\_RED | BACKGROUND\_INTENSITY);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*字体颜色变为红色\*\*\*\*\*\*\*\*\*\*\*\*\*/

void red()

{

SetConsoleColor(FOREGROUND\_RED | BACKGROUND\_BLUE | BACKGROUND\_GREEN | BACKGROUND\_RED | BACKGROUND\_INTENSITY );

}

/\*这两个函数的作用是将控制台窗口中输出内容的字体颜色分别设置为蓝色和红色，来增强软件界面的交互性和易读性。具体实现步骤如下：

* 通过调用SetConsoleColor()函数并传入对应的参数，来设置控制台窗口中输出内容的字体颜色。
* 在blue()函数中，将字体颜色设置为FOREGROUND\_BLUE（蓝色）。
* 在red()函数中，将字体颜色设置为FOREGROUND\_RED（红色）。
* 在设置字体颜色时，同时也将背景颜色设置为蓝、绿、红三种颜色组合成的混合色，并且设置了字体的强度为最高。\*/