|  |
| --- |
| **内容提要：**  利用C语言结构体数组，单链表等数据结构及图形化界面设计思想，实现一个图形化的酒店管理系统。用数组和单链表分别存储酒店客房信息和会员信息，系统基本完成了酒店客房信息管理、会员信息管理、顾客入住和退房三个模块的工作。 |
| **关键词：**结构体数组，单链表，信息管理 |
| **参考书目：C语言**课程设计案例精编 |
| 1. **引言**   酒店管理是一项复杂、繁琐但又细致的工作，信息量很大，一般不允许出现错误，因而在操作上经常造成许多不便之处，这会耗费大量的时间和精力，降低了工作效率，而且对酒店的经济效益和服务质量产生了极大的影响。因此我设计了这个酒店客房管理系统。该管理系统的开发，本着规范化、科学化、系统化的原则，并考虑到酒店在经营上的实际情况，采用客户/服务器结构，该系统具有检索迅速、查询方便、可靠性高、安全保密性好、用户界面美观、容易操作等优点。应用此管理系统后，酒店管理人员可掌握每位客人的具体情况，而且在对客人信息进行操作时，节省了一定的时间，很大程度上提高了酒店管理的效率和服务质量，为酒店的发展提供了捷径。   1. 概述   本篇使用了文本模式下图形界面的开发技巧，对C语言指针、结构体数组和单链表进行操作，可以使用快捷键选择菜单项，完成基本的酒店客房信息管理、会员信息管理、酒店客房的登录入住和退宿处理工作。   1. 需求分析   3.1 客房信息管理   1. 添加记录，从文件读取或者从键盘输入 2. 查询记录，按客房编号查找或者按名称查找 3. 更新记录，包括修改记录、删除记录和排序记录 4. 统计客房相关的信息   3.2 会员记录管理   1. 添加记录，从文件读取或者从键盘输入 2. 查询记录，按编客查找或者按姓名查找 3. 更新记录，包括修改记录、删除记录和排序记录 4. 统计会员相关的信息   3.3 登记入住和退房   1. 通过价格筛选适合的客房信息，并记录住客的信息 2. 通过客房编号查找客房，并退房   酒店客房管理系统    统计记录模块  入住与退房  更新记录模块  查询记录模块  添加记录模块  系统化界面模块  统计与酒店客房及会员有关信息  顾客退房  登记入住  排序记录  删除记录  修改记录  按名称查询  从键盘输入  按编号查询  系统界面设计  菜单界面控制  从文件读入  图1 系统功能模块图   1. 界面设计 2. 首先调用drawmain函数来显示主界面，主界面涉及菜单栏、显示编辑区和状态栏     菜单栏  显示编辑区  图2 系统界面设计  系统可以读写的方式打开酒店客房文件和会员文件，文件默认路径为E：\room和E:\member，若文件不存在，则新建相关文件。当打开某文件操作成功后，则从文件中一次读出一条记录，写入添加到新建的数组或者单链表中，然后进入主循环操作，等待用户按键输入，并进行按键判断。   1. 若用户按键为H、J、K中的任意键，则调用菜单控制函数menuctrl，进行菜单项的显示和控制。不同于图形模式下的线条和画框操作，文本模式下的图形界面主要利用在指定位置输出特殊字符来实现，其中指定位置可通过gotoxy()函数来实现，特殊字符可通过cprintf()函数指定字符的ASCII码来获得。图书管理系统共有Room、Member和R&R三个菜单项，用户可按光标上移或下移键在某菜单的子菜单之间循环移动，也可以使用光标的左移或右移键在三个菜单项之间循环移动。当光标移动到某个子菜单项上时，用户此时可以使用Enter键来选取相关菜单选项。     图3 菜单界面设计   1. 系统根据用户选择的菜单项结果，调用相应的函数完成相应的功能。在菜单控制模块中，它主要完成子菜单的显示、光标在子菜单之间的上下移动或者菜单之间的左右移动、子菜单项的选取，下面分别介绍这三项功能的具体实现。 2. 显示子菜单项。用户可按H、J、K 功能键来分别调 用 Room、Member、R&R三个菜单的子菜单项，即完成子菜单项的显示。在menuctrl () 函数中，它会根据功能键的键值调用 drawmenu(value , flag) 函数，参数value、flag都为局部变量， 分别用来保存调用某个菜单、某个菜单下的第几个菜单选项。例如，按J后，它的默认值为drawmenu(0, 0),表示绘制 Book 菜单及其6个子菜单选项，并将菜单选择光带条置千第一个菜单选项上。下面简要描述一下 drawmenu(value , flag) 函数的过程。   第一步；先取value 除以3的余数m(因为有3个菜单项， 所以除数选择3) , 根据m的值来绘制不同的菜单。m的取值为0 、1、2。当m等千0时，表示绘制Room菜单；其余类推。  第二步**：** 然后绘制菜单的边框及菜单选项值。  第三步：取 flag 除以x的余数 t, x的取值视m的取值而定， 如当m=6 时，x=6 ,因为 Book 菜单下有6个选项。  画Member子菜单  光标在Room子菜单下的第一行  光标在Room子菜单下的第二行  光标在Room子菜单下的第三行  光标在Room子菜单下的第四行  光标在Room子菜单下的第五行  光标在Room子菜单下的第六行  光标在Book子菜单下的第一行  光标在Book子菜单下的第一行  0  1  2  3  4  5  flag  画Room子菜单  0  Falg 光标位置参照Room子菜单  Value  1  光标在R&R子菜单下的第一行  光标在R&R子菜单下的第二行  光标在R&R子菜单下的第三行  0  1  2  flag  画R&R子菜单  2  第四步**：**根据t的值，用特殊的符号“&”在原来的位置重新显示菜单选项，以达到光标的指示效果。   1. 移动菜单光带条。当用户按H、J、K中的某个功能键调用了某个菜单后， 可继续按光标左移（键盘上的A，注意大小写）、右移（D）、上移（W）和下移（S）键来实现菜单之间的切换和菜单选项之间的切换。 2. 选取菜单。当用户将光标选择条置于某个菜单选项上时，可按回车键来选取该菜单选项。选取菜单操作的实现比较简单，它主要利用a=(value%3)\* 10+flag%b来计算出选择的菜单选项的编号。不同菜单选项选取后，a的值不同。这样，程序可根据a的值，返回给main()函数不同的标记 ， 在main() 函数中，可根据标记的不同来执行相关功能 。 3. 开发工具及运行环境   VS2019   1. 数据结构设计 2. 酒店客房信息   表1 酒店客房信息的数据结构   |  |  | | --- | --- | | 酒店客房编号 | 字符串（char） | | 客房价格 | 整型（int） | | 酒店空闲标记 | 整型（int） | | 入住顾客姓名 | 字符串（char） | | 顾客身份证号 | 字符串（char） | | 入住日期 | 结构体 | | 入住天数 | 整型（int） | | 顾客预付款 | 整型（int） |   typedef struct room  {  char num[15];  int price;  int empty\_flag;  char client\_name[15];  char client\_ic[20];  struct date client\_date;  int client\_totaldays;  int client\_ap  };  结构ROOM用于存储酒店客房相关的基本信息。其各字段值含义如下：   * num[15]：保存酒店客房编号 * Price：保存酒店客房价格 * Empty\_flag：保存客房空闲标记，1表示空闲，0表示未空闲 * Client\_name[15]：保存顾客姓名 * client\_ic[20]：保存顾客身份证号 * Client\_date：保存顾客入住日期 * Client\_numdays：保存顾客入住天数 * Client\_ap：保存预付款   其中，client\_date为struct date类型的结构变量，struct date为系统机构体，用于存储系统日期。该结构体有三个成员，分别是year、mon和day，表示年、月和日。   1. 会员信息   表2 会员信息的数据结构   |  |  | | --- | --- | | 编号 | 字符串（char） | | 姓名 | 字符串（char） | | 性别 | 字符串（char） | | 年龄 | 整型（int） | | 身份证号 | 字符串（char） | | 联系电话 | 字符串（char） |   typedef struct member  {  char num[15]; //会员编号  char name[15]; //姓名  char sex[4]; //性别  int age; //年龄  char ic[20]; //身份证号  char tele[15]; //联系方式  };   * 结构体member用于存储会员相关的基本信息，它将作为单链表的数据域。其各字段值含义如下： * Num[15]：保存会员编号 * Name[15]：保存会员姓名 * Sex[4]：保存会员性别，只能填入‘男’或者‘女’ * Age：保存会员年龄 * Ic[20]：保存会员身份证号 * Tele[15]：保存会员的联系电话  1. 单链表member\_node结构体   typedef struct member\_node {  struct member data;  struct member\_node\* next;  }\*Member\_Link, Member\_Node;//\*Member\_Link为Member\_Node类型结构体的指针H  /\*与界面控制相关的函数的声明\*/  在单链表member\_node结构体中，data为member结构体类型的数据，作为单链表结构体中的数据域，next为单链表中的指针域，用来存储直接后继结点的地址。Member\_Node为member\_node类型的结构体变量，\*Member\_Link为member\_node类型的指针变量。   1. 对象设计   系统代码要三个文件，头文件additional.h当中是数据结构和一些宏的定义，还有系统实现所用到的函数。Contral.cpp文件的代码负责界面的设计和键盘按键控制。main.cpp文件实现对存放在数组和链表中的数据的操作    图4 系统文件布局  实现系统的数据结构和方法：  typedef struct date  {  int year;  int mon;  int day;  };  typedef struct room  {  char num[15]; //客房编号  int price; //价格  int empty\_flag; //空闲标记  char client\_name[15]; //顾客姓名  char client\_ic[20]; //顾客身份证号  struct date client\_date;//顾客入住日期  int client\_totaldays; //顾客入住天数  int client\_ap; //顾客预付款  }ROOM;  typedef struct member  {  char num[15]; //会员编号  char name[15]; //姓名  char sex[4]; //性别  int age; //年龄  char ic[20]; //身份证号  char tele[15]; //联系方式  };  typedef struct member\_node {  struct member data;  struct member\_node\* next;  }\*Member\_Link, Member\_Node;//\*Member\_Link为Member\_Node类型结构体的指针H  /\*与界面控制相关的函数的声明\*/  void insline();  void insline1();  void gotoxy(int x, int y);  int wherex();  int wherey();  void mainmenu();  void drawmain();  int menuctrl();  /\*与数据结构逻辑相关的函数声明\*/  void modeset(int w, int h);  int AddRoom(ROOM tp[], int n);//添加客房记录  void AddMember(Member\_Link l);//添加会员记录  int DelRoom(ROOM tp[], int n);//删除客房记录  void DelMember(Member\_Link l);//删除会员记录  void SaveRoom(ROOM tp[], int n);//退出系统时保存客房记录  void SaveMember(Member\_Link l);//退出系统时保存会员记录  void QueryRoom(ROOM tp[], int n);//查询客房记录  void QueryMember(Member\_Link l);//查询会员记录  void ModifyRoom(ROOM tp[], int n);//修改客房记录  void ModifyMember(Member\_Link l);//修改会员记录  void SortRoom(ROOM tp[], int n);//排序客房记录  void SortMember(Member\_Link l);//排序会员记录  void CountRoom(ROOM tp[], int n);//统计客房记录  void CountMember(Member\_Link l);//统计会员记录  void RegisterRoom(ROOM tp[], int n, Member\_Link ll);//登记入住  void ReturnRoom(ROOM tp[], int n, Member\_Link ll);//退房   1. 算法设计的思想、流程图、分析 2. 添加记录模块   酒店客房管理系统中，客房记录使用结构数组存储，会员记录使用单链表存储，这些记录的添加，可以从文本形式存储的数据文件中读入，也可以从键盘逐个记录。当从数据文件中读入记录时，它就是在以记录为单位存储的数据文件中，调用fread()文件读取函数，将记录逐条复制到结构数组和单链表中。并且这个操作在main函数中执行，即当酒店客房管理系统进入显示菜单界面时，该操作已经执行了。若该文件中没有数据时，系统会提示记录为空，此时，用户可以通过选择Room菜单或者Member菜单下的添加记录模块选项，调用AddRoom()或AddMember()函数，进行记录的输入，即可完成在数组中添加新元素和单链表中添加新节点的操作。    null  data  data  data  null  data  图5 向会员记录链表中添加节点   1. 记录查找定位   用户景星酒店客房管理时，对某个记录处理前，需要按照条件找到这条记录，需要一个函数完成数组结点定位的功能。查找定位功能由LocateRoom()函数与LocateMember()函数实现，其中LocateRoom()用于定位数组中符合要求的记录，并返回保存该记录的数组元素下标，LocateMember()函数用于定位会员链表中符合要求的结点，并返回指向该节点的指针。需要遍历数组或者链表。  data  data  data  data  \*head Data.num  P Data.name  图6 查询记录   1. 更新记录模块 2. 修改记录   酒店客房修改记录操作由ModifyRoom()函数实现。在修改酒店客房记录操作中，系统会先按输入的编号查询到该记录，然后提示用户修改编号之外的相关字段值。   1. 删除记录   酒店客房删除记录操作由DelRoom()函数实现。在删除操作后，系统会按照用户要求找到该酒店客房记录的节点，然后从结构体数组中删除该记录。同样，会员记录删除函数DelMember()的实现与之类似，只不过采用单链表实现。第一步，输入要修改的编号或名称，输入后调用定位函数Locate()或LocateReader()在单链表中逐个对结点数据域中的编号字段的值进行比较，直到找到该编号的记录，返回指向该记录的指针；第二步，若找到该记录，将该记录所在的前驱节点的指针指向目标结点的后继结点。具体过程如图二所示，图中p为指向需删除结点的指针变量，其中，q为p所指节点的前驱节点的指针变量，删除结点p执行的操作为：q->next=p->next或者q->next=q->next->next.    data  data  data  data  q P  图7单链表中删除结点   1. 记录排序   采用冒泡排序对酒店客房记录按客房价格字段进行升序排序，采用直接排序对会员记录按会员编号进行升序排序。  采用冒泡排序法对客房记录进行升序排序由SortRoom()函数实现，它的基本思想为：将两个相邻的两个数组元素的客房价格字段值进行比较，若左边的值大于右边的值，则将这两个元素的值进行交换；若左边的值小于等于右边的值，则这两个的位置不变。右边的值继续好下一个值作比较，重复此动作，直到比较到最后一个值。   |  |  |  |  |  | | --- | --- | --- | --- | --- | | 4 | 6 | 7 | 3 | 5 |   图8 冒泡排序  过程演示  未排序的数组   |  |  |  |  |  | | --- | --- | --- | --- | --- | | 4 | 6 | 3 | 5 | 7 |   第一趟排序 最大的数7排在了最后边   |  |  |  |  |  | | --- | --- | --- | --- | --- | | 4 | 3 | 5 | 6 | 7 |   第二趟排序 第二大的数6排在了倒数第二个   |  |  |  |  |  | | --- | --- | --- | --- | --- | | 3 | 4 | 5 | 6 | 7 |   第三趟排序 已经完成排序，但是冒牌排序还没结束，第四趟只比较前三个元素，相同的第五趟排序只比较前两个元素。  选择直接选择法对会员记录进行排序，由SortMember()函数完成。直接选择排序的基本思想：从欲排序的n个元素中，以线性查找的方式找出最小的元素和第一个元素交换，再从生下的(n-1)个元素中，找出最小的元素和第二个元素进行交换，依次类推，直到所有元素均已完成排序  外层循环决定每次排序的开始位置，以及需要交换结点之间  指针关系的改变。  内层循环负责在单链表中找到当前关键字最小的结点。  重复第一步和第二步，直到待排序链表取出的结点的指针域  为NULL，即此节点为链表的尾部结点后，排序完成。    8  7  9  2  4  10    Head 第一次循环遍历链表找到最小的元素与链表的第一个元素交换位置    8  7  9  10  4  2    图9 直接选择排序过程演示  第二次循环遍历链表找到最小的元素与链表的第二个元素交换位置   1. 统计记录模块   统计记录模块主要完成了对酒店客房及会员有关的信息的统计。例如，统计各档次的空房数量够，入住天数超期的房间号，统计会员的总数量，男性会员和女性会员的数量，统计目前会员年龄最大的会员名等。统计入住天数需要获取入住时系统的当前时间，另外不同年份还需要判断是否是闰年，年份year能被400整除，并且不能被100整除，或能被4整除，并且能被400整除，那么该年份为闰年。入住天数统计：  入住日期与当前日期不在同一年，总入住天数=入住日期至当年最后一天的相隔天数+相隔年数的总天数+当前日期与当年的第一天相隔天数  入住日期与当前日期在同一年，总入住天数=入住日期至当月最后一天的相隔天数+相隔月数的总天数+当前日期与当月的第一天相隔天数   1. 酒店客房登记入住与退房模块   登记入住由RegisterRoom()函数实现。首先，提示用户输入顾客能够承受的客房价格上限，如存在小于此价格上限且客房为空闲的客房，则显示所有的满足条件的记录，并为顾客分配相应的客房；然后，确认顾客是否为会员，若不是会员，即普通顾客，系统会要求输入相应顾客基本信息：对于会员，顾客只需要输入正确的会员编号，系统就会将会员信息复制到顾客客房记录中的顾客信息字段。另外，对于普通顾客和会员都需要输入的信息包括入住天数、客房预付款。其中客房预付款≥客房价格×入住天数+100，即客房预付款为总客房费用另加100元押金。最后更新相关客房字段，如空闲状态tp[suffix].empty\_flag=0,入住日期等。  酒店退房由ReturnRoom()函数实现。与酒店客房登记入住类似，首先提示用户输入欲退房的编号，系统查询该客房编号是否已经存在且处于非空闲状态，若满足这两条件，则允许执行退房操作。然后，提示用户确认顾客是否为会员，若是会员，还需要根据提供的会员编号进行确认。若是会员，则将享受客房价格的9折优惠，若不是会员，则进行全价结账。最后，更新相关客房字段，如空闲状态置为空、客房中的顾客信息置为空等。   1. 主main()函数执行流程图   开始  打开客房文件和会员文件，分别调入数组和表中存储  进入while(1)主循环  调用bioskey()返回用户的按键值  是否为H、J、K  否  是  调用Menuctrl()，进行菜单显示和控制  返回结果是否为EXIT  否  是  调用文件保存函数，进行写数据文件操作  结束  图10 main函数运行流程图   1. 运行结果分析 2. 添加记录   当用户选择Room或Member下的Add选项并按Enter键后，即可进行记录添加工作。其输入记录过程如图所示。输入编号为004的图书，当再次输入4编号时，系统的提示信息结果如图11-2所示    图11-1 添加room记录操作    图11-2 添加相同编号的客房结果    图11-3 添加会员记录   1. 查找记录   当用户选择Room或Member下的Query选项后并按Enter键后，即可进入查询界面。如图12所示。    图12-1 按客房编号查询记录    12-2 按照会员编号查询记录   1. 修改记录   当用户选择Room或Member下的Modify选项并按Enter键后，即可进行记录修改工作，如图13所示，用户已经成功修改了一条编号为004的客房记录。    图12-1 修改客房记录    图13-2 修改会员记录   1. 删除客房记录   当用户选择Room或Member下的Delete选项并按Enter键后，即可进行记录删除工作，如图14，用户已成功删除一条编号为004的客房记录。    图14 删除客房记录   1. 排序记录   当用户选择Room或Member下的Sort选项并按Enter键后，即可进行记录排序操作。客房记录采用冒泡法进行排序如图15-1所示。用直接选择排序方法为客房记录按价格排序，图15-2为会员记录按编号排序后的结果    图15-1 按价格排序客房记录    图15-2 按编号排序会员记录   1. 统计用户   当用户选择Room或Member下的Count选项并按Enter键后，即可进行记录统计操作。图16-1为图书统计结果，图16-2为读者信息统计结果。    图16-1 统计客房记录    图16-2 统计会员记录   1. 登记入住   当用户选择R&R菜单下的Register选项后并按Enter键后，即可进行借图书操作。图17-1所表示为非会员顾客FE成功入住107房间，预期住1填，预付款450。图17-2为客户入住后客房信息的查询结果，结果表示为编号为107的客房已有客户入住。    图17-1 登记入住    图17-2 客房记录查询   1. 退房   当用户选择R&R菜单下的Return选项后并按Enter键后，即可进行退房操作。图18-1所表示为客房编号为107的顾客成功退房。图18-2为退房后客房信息的查询结果    图18-1 107顾客成功退房    图18-2 退房后的客房信息   1. 保存记录   如图19所示，当用户选择R&R菜单下的EXIT选项后并按Enter键后，会提示用户是否退出系统，当用户选择“Y”或“y”后，系统会自动将客房记录和会员记录分别存入E:\book和E:\reader文件中，最后执行系统  退出工作。    图19 退出酒店管理系统   1. 总结   本篇重点介绍了各功能模块的设计原理、文本模式下图形界面的设计、菜单的灵活控制，综合利用结构体数组和单链表存储数据的方法，以及冒泡排序与直接选择排序的思想。通过本篇学习，掌握到了以下知识点;   1. 文本窗口打下的设定、窗口的颜色设置、窗口文本的清除和输入输出等 2. 对结构体数组和单链表的各种基本操作 3. 对文件的打开、关闭、读取、写入操作。 4. 参考文献   <https://blog.csdn.net/LittleWhiteLv/article/details/80370260>读取鼠标光标位置  <https://www.runoob.com/cprogramming/c-function-fwrite.html> 库函数-fwrite()  <https://www.bejson.com/othertools/keycodes/>获取键盘按键值  <https://blog.csdn.net/zwb8848happy/article/details/7284650C>语言清屏函数  <https://blog.csdn.net/konghouy/article/details/80530937> feof()函数原理和用法  **附录**  **源代码**  additonal.h  #pragma once  #pragma warning(disable:4996)  #include<stdio.h>  #include <Windows.h>  #include<graphics.h>  # include "conio.h" //屏幕操作函数库  #include <time.h>  #include<string.h>  //与按键有关的宏定义  #define LEFT 65 //光标左移  #define RIGHT 68 //光标右移  #define DOWN 83 //光标下移  #define UP 87 //光标上移  #define ESC 27 //Esc键：取消菜单打开操作  #define H 72 // F1键：打开文件菜单  #define J 74//F2键：打开编辑菜单  #define K 75//F3键：打开帮助菜单  #define ENTER 13 //回车键：换行  //与菜单选项有关的宏定义  #define ADD\_ROOM 100  #define QUERY\_ROOM 101  #define MODIFY\_ROOM 102  #define DEL\_ROOM 103  #define SORT\_ROOM 104  #define COUNT\_ROOM 105  #define ADD\_MEMBER 200  #define QUERY\_MEMBER 201  #define MODIFY\_MEMBER 202  #define DEL\_MEMBER 203  #define SORT\_MEMBER 204  #define COUNT\_MEMBER 205  #define REGISTER\_ROOM 300  #define RETURN\_ROOM 301  #define EXIT 302  #define N 200 //定义酒店客房的最大数量  /\*定义与Room记录格式化输出有关的宏定义\*/  #define HEADER1 "| Number | Price | E\_F | C\_Name | C\_IC | Date | CNum | C\_AP | \n"  #define FORMAT1 "| %-5s | %5d | %3d | %-6s | %18s | %4d-%2d-%2d | %4d | %4d | \n"  #define DATA1 lll->num,lll->price,lll->empty\_flag,lll->client\_name,lll->client\_ic,\  lll->client\_date.year,lll->client\_date.mon,lll->client\_date.day,lll->client\_totaldays,lll->client\_ap  #define HEADER2 "|Number| Name | Sex | Age | IC | Tele | \n"  #define FORMAT2 "| %-4s | %-4s | %-3s | %3d | %18s | %11s | \n"  #define DATA2 lll->data.num,lll->data.name,lll->data.sex,lll->data.age,lll->data.ic,lll->data.tele  typedef struct date  {  int year;  int mon;  int day;  };  typedef struct room  {  char num[15]; //客房编号  int price; //价格  int empty\_flag; //空闲标记  char client\_name[15]; //顾客姓名  char client\_ic[20]; //顾客身份证号  struct date client\_date;//顾客入住日期  int client\_totaldays; //顾客入住天数  int client\_ap; //顾客预付款  }ROOM;  typedef struct member  {  char num[15]; //会员编号  char name[15]; //姓名  char sex[4]; //性别  int age; //年龄  char ic[20]; //身份证号  char tele[15]; //联系方式  };  typedef struct member\_node {  struct member data;  struct member\_node\* next;  }\*Member\_Link, Member\_Node;//\*Member\_Link为Member\_Node类型结构体的指针H  /\*与界面控制相关的函数的声明\*/  void insline();  void insline1();  void gotoxy(int x, int y);  int wherex();  int wherey();  void mainmenu();  void drawmain();  int menuctrl();  /\*与数据结构逻辑相关的函数声明\*/  void modeset(int w, int h);  int AddRoom(ROOM tp[], int n);  void AddMember(Member\_Link l);  int DelRoom(ROOM tp[], int n);  void DelMember(Member\_Link l);  void SaveRoom(ROOM tp[], int n);  void SaveMember(Member\_Link l);  void QueryRoom(ROOM tp[], int n);  void QueryMember(Member\_Link l);  void ModifyRoom(ROOM tp[], int n);  void ModifyMember(Member\_Link l);  void SortRoom(ROOM tp[], int n);  void SortMember(Member\_Link l);  void CountRoom(ROOM tp[], int n);  void CountMember(Member\_Link l);  void RegisterRoom(ROOM tp[], int n, Member\_Link ll);  void ReturnRoom(ROOM tp[], int n, Member\_Link ll);  static int saveflag = 0; //是否需要存盘的全局变量  /\*用于保存系统的当前时间\*/  static int currentYear;  static int currentMonth;  static int currentDay;  contral.c  #include"additional.h"  void insline() {  int i;  for (i = 0; i < 87; i++) {  printf("%c", '\*');  }  }  void insline1() {  int i;  for (i = 1; i < 86; i++) {  printf("%c", ' ');  }  }  void gotoxy(int x, int y)  {  int xx = 0x0b;  HANDLE hOutput;  COORD loc;  loc.X = x;  loc.Y = y;  hOutput = GetStdHandle(STD\_OUTPUT\_HANDLE);  SetConsoleCursorPosition(hOutput, loc);  return;  }  int wherex()  {  CONSOLE\_SCREEN\_BUFFER\_INFO pBuffer;  GetConsoleScreenBufferInfo(GetStdHandle(STD\_OUTPUT\_HANDLE), &pBuffer);  return (pBuffer.dwCursorPosition.X + 1);  }  int wherey()  {  CONSOLE\_SCREEN\_BUFFER\_INFO pBuffer;  GetConsoleScreenBufferInfo(GetStdHandle(STD\_OUTPUT\_HANDLE), &pBuffer);  return (pBuffer.dwCursorPosition.Y + 1);  }  void modeset(int w, int h)  {  HANDLE hOut = GetStdHandle(STD\_OUTPUT\_HANDLE);  COORD size = { w,h };  SetConsoleScreenBufferSize(hOut, size);  SMALL\_RECT rc = { 1,1,w,h };  SetConsoleWindowInfo(hOut, true, &rc);  system("cls");  return;  }  void drawmain2()  {  MultiByteToWideChar(CP\_UTF8, NULL, NULL, NULL, NULL, NULL);  initgraph(700, 600);  SetWindowText(GetHWnd(), \_T("酒店客房管理系统"));  COLORREF color;//定义一个颜色类型的变量  color = RGB(189, 190, 0);  settextcolor(color);//设置字体颜色  settextstyle(20, 7, \_T("Times New Roman"));  color = RGB(179, 23, 220);  setlinecolor(color);  line(1, 1, 1, 600);  line(700, 1, 700, 600);  line(1, 1, 700, 1);  line(1, 50, 700, 50);  line(1, 550, 700, 550);  line(1, 599, 700, 599);  outtextxy(150, 20, \_T("ROOM"));  outtextxy(300, 20, \_T("MENBER"));  outtextxy(450, 20, \_T("R&R"));  outtextxy(275, 150, \_T("press H -> ROOM\n"));  outtextxy(275, 225, \_T("press J -> MENBER\n"));  outtextxy(275, 300, \_T("press K -> R&R\n"));  outtextxy(0, 560, \_T("Hotel Managment System"));  outtextxy(600, 560, \_T("Version 2.0"));  getch();  closegraph();  }  void mainmenu() /\*绘制系统主界面\*/  {  int i;  /\*打印菜单内容\*/  for (i = 0; i <= 30; i++)  {  gotoxy(1, i);  cprintf("%c", '#');  gotoxy(89, i);  cprintf("%c", '#');  }  gotoxy(2, 0); insline();  gotoxy(2, 2); insline();  gotoxy(2, 28); insline();  gotoxy(2, 30); insline();  gotoxy(12, 1); printf("%c %c ROOM %c %c", '|', '<', '>', '|');  gotoxy(37, 1); printf("%c %c MENBER %c %c", '|', '<', '>', '|');  gotoxy(64, 1); printf("%c %c R&R %c %c", '|', '<', '>', '|');  gotoxy(5, 29); printf("Hotel Managment System");  gotoxy(70, 29); printf("Version 2.0");  gotoxy(65, 32);  }  void drawmain()  {  system("color d");  /\*打印伪窗口竖线\*/  mainmenu();  gotoxy(35, 10); printf("press H -> Room");  gotoxy(35, 12); printf("press J -> Menber");  gotoxy(35, 14); printf("press K -> R&R");  }  //绘制菜单  void drawmenu(int m, int n) {  int i;  if (m % 3 == 0)  {  system("cls");  /\*打印边框\*/  for (i = 0; i < 10; i++)  {  gotoxy(12, 2 + i); cprintf("%c", '\*');  gotoxy(25, 2 + i); cprintf("%c", '\*');  }  gotoxy(12, 11); cprintf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  /\*打印菜单内容\*/  mainmenu();  gotoxy(14, 3); cprintf("Add ");  gotoxy(14, 4); cprintf("Query ");  gotoxy(14, 5); cprintf("Modify ");  gotoxy(14, 6); cprintf("Delete ");  gotoxy(14, 7); cprintf("Sort ");  gotoxy(14, 8); cprintf("Count ");  //gotoxy(14, 9); cprintf("Details");  /\*打印选项\*/  switch (n % 6) {  case 0:gotoxy(14, 3); cprintf("Add &"); break;  case 1:gotoxy(14, 4); cprintf("Query &"); break;  case 2:gotoxy(14, 5); cprintf("Modify &"); break;  case 3:gotoxy(14, 6); cprintf("Delete &"); break;  case 4:gotoxy(14, 7); cprintf("Sort &"); break;  case 5:gotoxy(14, 8); cprintf("Count &"); break;  //case 6:gotoxy(14,9); cprintf("Details &");break;  }  }  if (m % 3 == 1)  {  system("cls");  /\*打印边框\*/  for (i = 1; i < 10; i++)  {  gotoxy(37, 2 + i); cprintf("%c", '\*');  gotoxy(52, 2 + i); cprintf("%c", '\*');  }  gotoxy(39, 11); cprintf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*");//  /\*打印菜单内容\*/  mainmenu();  gotoxy(39, 3); cprintf("Add ");  gotoxy(39, 4); cprintf("Query ");  gotoxy(39, 5); cprintf("Modify ");  gotoxy(39, 6); cprintf("Delete ");  gotoxy(39, 7); cprintf("Sort ");  gotoxy(39, 8); cprintf("Count ");  //gotoxy(39, 9); cprintf("Details");  /\*打印选项\*/  switch (n % 6) {  case 0:gotoxy(39, 3); cprintf("Add &"); break;  case 1:gotoxy(39, 4); cprintf("Query &"); break;  case 2:gotoxy(39, 5); cprintf("Modify &"); break;  case 3:gotoxy(39, 6); cprintf("Delete &"); break;  case 4:gotoxy(39, 7); cprintf("Sort &"); break;  case 5:gotoxy(39, 8); cprintf("Count &"); break;  //case 6:gotoxy(39,9); cprintf("Details &");break;  }  }  if (m % 3 == 2)  {  system("cls");  /\*打印边框\*/  for (i = 1; i < 10; i++)//列  {  gotoxy(64, 2 + i); cprintf("%c", '\*');  gotoxy(77, 2 + i); cprintf("%c", '\*');  }  gotoxy(64, 11); cprintf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  /\*打印菜单内容\*/  mainmenu();  gotoxy(66, 3); cprintf("Register ");  gotoxy(66, 5); cprintf("Return ");  gotoxy(66, 7); cprintf("Exit ");  /\*打印选项\*/  switch (n % 3) {  case 0:gotoxy(66, 3); cprintf("Register &"); break;  case 1:gotoxy(66, 5); cprintf("Return &"); break;  case 2:gotoxy(66, 7); cprintf("Exit &"); break;  }  }  }  int menuctrl() /\*菜单控制\*/  {  int A, B, value,flag = 36, a, b;  while (1)  {  /\*获取键值\*/  while (1)  {  if (\_kbhit())  {  A = getch();  A = getch();  if (A == 'H' || A == 'J' || A == 'K') break;  }  }  if (A == 'H')  {  drawmenu(0, flag);  value = 300;  }  if (A == 'J') {  drawmenu(1, flag);  value = 301;  }  if (A == 'K') {  drawmenu(2, flag);  value = 302;  }  while (1)  {  /\*获取键值\*/  while (1)  {  if (\_kbhit()) {  B = getch();  if (B == ESC || B == ENTER) {  break;  }  B = getch();  if (B == LEFT || B == RIGHT || B == UP || B == DOWN) break;  }  }  /\*返回上一级菜单\*/  if (B == ESC)  {  system("cls");  drawmain();  break;  }  if (flag == 0) flag = 36;  if (value == 0) value = 300;  /\*返回功能键值\*/  if (B == ENTER)  {  if (value % 3 == 0) b = 6;  if (value % 3 == 1) b = 6;  if (value % 3 == 2) b = 3;  a = (value % 3) \* 10 + flag % b;  if (a == 0) return ADD\_ROOM;  if (a == 1) return QUERY\_ROOM;  if (a == 2) return MODIFY\_ROOM;  if (a == 3) return DEL\_ROOM;  if (a == 4) return SORT\_ROOM;  if (a == 5) return COUNT\_ROOM;  if (a == 10) return ADD\_MEMBER;  if (a == 11) return QUERY\_MEMBER;  if (a == 12) return MODIFY\_MEMBER;  if (a == 13) return DEL\_MEMBER;  if (a == 14) return SORT\_MEMBER;  if (a == 15) return COUNT\_MEMBER;  if (a == 20) return REGISTER\_ROOM;  if (a == 21) return RETURN\_ROOM;  if (a == 22) return EXIT;  }  else if (B == UP) { /\*切换选项\*/  drawmenu(value, --flag);  }  else if (B == DOWN) {  drawmenu(value, ++flag);  }  else if (B == LEFT) {/\*切换菜单\*/  drawmenu(--value, flag);  }  else {  drawmenu(++value, flag);  }  }  }  }  main.c  # include "additional.h"  #pragma warning(disable:4996)  int main()  {  ROOM GR[N]; //定义客房结构数组  Member\_Link ll; //定义会员链表指针  FILE\* fp1, \* fp2; //fp1为指向客房文件指针，fp2为指向会员的文件指针  Member\_Node\* p2, \* r2; //定义会员记录指针变量  int A, B, i;  char ch,a; //保存（y,Y,n,N）  int count1 = 0, count2 = 0; //分别保存客房文件或者会员文件中的记录条数  //modeset(80, 35); //显示系统主界面  drawmain();  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*打开客房文件ROOM，将其调入数组中存储\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  fp1 = fopen("E:\\room", "at+");//以追加方式打开一个二进制文件，可读可写，若此文件不存在，会创建此文件  if (fp1 == NULL)  {  system("cls"); mainmenu(); gotoxy(2, 3);  printf("\n=====>can not open file1!\n");  exit(0);  }  while (!feof(fp1))  {  if (fread(&GR[count1], sizeof(ROOM), 1, fp1) == 1)//一次从文件中读出一条客房记录  {  count1++;  }  }fclose(fp1); //关闭文件  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*打开客房文件MEMBER，将其调入数组中存储\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  ll = (Member\_Node\*)malloc(sizeof(Member\_Node));  if (!ll)  {  system("cls"); mainmenu(); gotoxy(2, 3);  printf("\n allocate memory failure");//如果没有申请到打印提示信息  return 0;//返回主界面  }  ll->next = NULL;  r2 = ll;  fp2 = fopen("E:\\member", "at+");//以追加方式打开一个二进制文件，可读可写，若此文件不存在，会创建此文件  if (fp2 == NULL)  {  system("cls"); mainmenu(); gotoxy(2, 3);  printf("\n=====>can not open file2!\n");  exit(0);  }  while (!feof(fp2)) /\*feof()函数检测流上的文件结束符，若文件结束，则返回非零的值，否则返回零\*/  {  p2 = (Member\_Node\*)malloc(sizeof(Member\_Node));  if (!p2)  {  system("cls"); mainmenu(); gotoxy(2, 3);  printf("memory malloc failure!\n");//没有申请成功  exit(0); //退出  }  if (fread(p2, sizeof(Member\_Node), 1, fp2) == 1)//一次从文件中读取一条图书记录  {  p2->next = NULL;  r2->next = p2;  r2 = p2;//r指针向后移一个位置  count2++;  }  }  fclose(fp2);//关闭文件  p2 = r2;  while (1)  {  B = menuctrl();  switch (B)  {  case ADD\_ROOM:count1 = AddRoom(GR, count1); break;//增加图书记录  case QUERY\_ROOM:QueryRoom(GR, count1); break;//删除图书记录  case MODIFY\_ROOM:ModifyRoom(GR,count1); break;//修改图数记录  case DEL\_ROOM:count1=DelRoom(GR,count1); break;//删除图书记录  case SORT\_ROOM:SortRoom(GR,count1); break;//排序图书记录  case COUNT\_ROOM:CountRoom(GR,count1); break;//统计图书记录  case ADD\_MEMBER:AddMember(ll); break;//增加读者记录  case QUERY\_MEMBER:QueryMember(ll); break;//查询读者记录  case MODIFY\_MEMBER:ModifyMember(ll); break;//修改读者记录  case DEL\_MEMBER:DelMember(ll); break;//删除读者记录  case SORT\_MEMBER:SortMember(ll); break;//排序读者记录  case COUNT\_MEMBER:CountMember(ll); break;//统计读者记录  case REGISTER\_ROOM:RegisterRoom(GR,count1,ll); break;//借书  case RETURN\_ROOM:ReturnRoom(GR, count1,ll); break;//还书  case EXIT: //退出系统  {  system("cls"); mainmenu(); gotoxy(3, 3);  cprintf("====>Are you really exit the Room Management System?(y/n):");  scanf("%c", &ch);  if (ch == 'y' || ch == 'Y')  {  SaveRoom(GR,count1); SaveMember(ll); exit(0);  }  }  }  system("cls");  drawmain();  }  }  void stringinput(char\* t, int lens, const char\* notice,int flag)  {  char n[225];  int x=0, y=0,i=0;  do {  if(x&&i)  {  gotoxy(x + 2, y-2);insline1();  gotoxy(x + 2, y-1);insline1();  i--;  gotoxy(x + 2, y-2);  }  printf(notice);  scanf("%s", n);  if(flag==2&&strlen(n)!=lens)  {  x = wherex();  y = wherey();  gotoxy(x + 2,y-1); printf("please input the 11 digits Telephone correctly\n");i++;getchar();getchar();  continue;  }  if(flag==3&&strlen(n)!=lens)  {  x = wherex();  y = wherey();  gotoxy(x + 2,y-1); printf("please input the 18 digits ID num correctly\n");i++;getchar();getchar();  continue;  }  if(flag==4&&strlen(n)!=lens&&strcmp(n,"0"))  {  x = wherex();  y = wherey();  gotoxy(x + 2,y-1); printf("please input the 3 digits num correctly\n");i++;getchar();getchar();  continue;  }  if(flag==1&&strcmp(n,"男")&&strcmp(n,"女"))  {  x = wherex();  y = wherey();  gotoxy(x + 2, y-1); printf("please input '男' or '女'\n");i++;getchar();getchar();  continue;  }  if (strlen(n) > lens)  {  x = wherex();  y = wherey();  gotoxy(x + 2, y-1); printf("exceed the required length!\n");i++;getchar();getchar();  }/\*进行长度校验，如果超出len长度，从新输入\*/  } while (strlen(n)>lens||((flag==3||flag==2)&&strlen(n)!=lens)||flag==1&&strcmp(n,"男")&&strcmp(n,"女")||flag==4&&strlen(n)!=lens&&strcmp(n,"0"));  strcpy(t, n);/\*将输入的字符串复制到字符串t中\*/  }  int LocateRoom(ROOM tp[],int n, char findmess[], char nameornum[])//图书定位  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  作用：用于定位链表中符合要求的节点，并返回指向该节点的指针  参数：findmess[]保存要查找的具体内容；nameornum[]保存按什么查找；  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  {  int i=0;  if (strcmp(nameornum, "num") == 0)//按图书编号查询  {  while (i<n)  {  if (strcmp(tp[i].num, findmess) == 0)//若找到findmess值的图书编号  return i;  i++;  }  }  else if (strcmp(nameornum, "name") == 0)//按图书名查询  {  while (i<n)  {  if (strcmp(tp[i].client\_name, findmess) == 0)//若找到findmess值的图书名  return i;  i++;  }  }  return -1;//若未找到，返回一个空指针  }  Member\_Node\* LocateMember(Member\_Link l, char findmess[], char nameornum[])//会员定位  {  Member\_Node\* r;  if (strcmp(nameornum, "num") == 0)//按会员编号查询  {  r = l->next;  while (r)  {  if (strcmp(r->data.num, findmess) == 0)//若找到findmess值的会员编号  return r;  r = r->next;  }  }  else if (strcmp(nameornum, "name") == 0)//按会员姓名查询  {  r = l->next;  while (r)  {  if (strcmp(r->data.name, findmess) == 0)//若找到findmess值的会员  return r;  r = r->next;  }  }  return 0;//若未找到，返回一个空指针  }  //添加记录  int AddRoom(ROOM tp[], int n)  {  char ch, flag = 0,num[15];  int i;  system("cls");  while (1)  {  while (1)  {  system("cls");  mainmenu();  gotoxy(3, 4);  stringinput(num, 3, "input room number(press '0'return menu):",4);  flag = 0;  if (strcmp(num, "0") == 0)  {  return n;  }  i = 0;  while (i < n)  {  if (strcmp(tp[i].num, num) == 0)  {  flag = 1;  break;  }  i++;  }  if (flag == 1)  {  gotoxy(3, 5);  getchar();  printf("====>The number %s is existing,please try again(Y/N)?", num);  scanf("%c", &ch);  if (ch == 'Y' || ch == 'y')  continue;  else  return n;  }  else break;  }  /\*给客房记录赋值\*/  strcpy(tp[n].num, num); /\*将字符串num复制到tp[n].num中\*/  gotoxy(3,6); printf("Room Price:"); scanf("%d", &tp[n].price);  tp[n].empty\_flag = 1; /\*客房空闲标记，1表示空闲，0表示未空闲\*/  strcpy(tp[n].client\_name, ""); /\*顾客姓名\*/  strcpy(tp[n].client\_ic,""); /\*顾客身份证号\*/  tp[n].client\_date.year = 0;  tp[n].client\_date.mon = 0;  tp[n].client\_date.day = 0;  tp[n].client\_totaldays = 0;  tp[n].client\_ap = 0;  saveflag = 1;  n++;  gotoxy(3,9); printf(">>>>press any key to start next record!");  getchar(); getchar();  }  return n;  }  void AddMember(Member\_Link l)  {  Member\_Node\* p, \* r, \* s;//实现添加操作的临时的结构体指针变量  char ch, flag = 0, num[10];  int temp;  r = l;  s = l->next;  system("cls");  while (r->next != NULL)  r = r->next;//将指针移至于链表最末尾，准备添加记录  while (1)//一次可输入多条记录，直至输入会员编号为0的记录结点添加操作  {  while (1)//输入会员编号，保证该编号没有被使用，若输入编号为0，则退出添加记录操作  {  system("cls");  mainmenu();  gotoxy(3, 4);  stringinput(num, 3, "input member number(press '0' return menu):",4);//格式化输入会员编号并检验  flag = 0;  if (strcmp(num, "0") == 0)//输入为0，则退出添加操作，返回主界面  {  return;  }  s = l->next;  while (s)//查询该会员编号是否已经存在，若存在则要求重新输入一个未被占用的编号  {  if (strcmp(s->data.num, num) == 0)  {  flag = 1;  break;  }  s = s->next;  }  if (flag == 1)//提示用户是否重新输入  {  gotoxy(3, 5);  getchar();  printf("====>The number %s is existing,please try again(y/n)?", num);  scanf("%c", &ch);  if (ch == 'y' || ch == 'Y')  continue;  else  return;  }  else  {  break;  }  }  p = (Member\_Node\*)malloc(sizeof(Member\_Node));//申请内存空间  if (!p)  {  printf("\n allocate memory failure ");//如没有申请到，打印提示信息  return;  }  //给图书记录赋值  strcpy(p->data.num, num);//将字符串num复制到p->data.num中  gotoxy(3, 5); stringinput(p->data.name, 15, "Member Name:",0);  gotoxy(3, 6); stringinput(p->data.sex, 15, "Member Sex:",1);  gotoxy(3, 7); stringinput(p->data.tele, 11, "Member Tele:",2);  gotoxy(3, 8); printf("Member Age:"); scanf("%d", &temp);  p->data.age = temp;  gotoxy(3, 9); stringinput(p->data.ic, 18, "Member ic:",3);  gotoxy(3, 10); printf(">>>>press any key to start next record!");  getchar(); getchar();  p->next = NULL;//表明这是链表的尾部结点  r->next = p;//将新建的结点加入链表尾部中  r = p;  saveflag = 1;  }  return;  }  //删除记录  int DelRoom(ROOM tp[], int n)  {  int i = 0, p = 0;  char findmess[20];  if (n < 0)  {  system("cls");  mainmenu();  gotoxy(3, 5);  printf("\n====>No room record!\n");  getchar();  return 0;  }  system("cls");  mainmenu();  gotoxy(3, 5);  stringinput(findmess, 10, "input the existing room number:",0);  p = LocateRoom(tp, n, findmess, "num");  if (p !=-1)  {  for (i = p + 1; i < n; i++) //删除此记录，后面记录向前移  {  strcpy(tp[i - 1].num, tp[i].num);  tp[i - 1].price = tp[i].price;  tp[i - 1].empty\_flag = tp[i].empty\_flag;  strcpy(tp[i - 1].client\_name, tp[i].client\_name);  strcpy(tp[i - 1].client\_ic, tp[i].client\_ic);  tp[i - 1].client\_date.year = tp[i].client\_date.year;  tp[i - 1].client\_date.mon = tp[i].client\_date.mon;  tp[i - 1].client\_date.day = tp[i].client\_date.day;  tp[i - 1].client\_totaldays = tp[i].client\_totaldays;  tp[i - 1].client\_ap = tp[i].client\_ap;  }  gotoxy(3, 9);  printf("=====>delete success!\n");  getchar(); getchar();  saveflag = 1;  n--;  }  else{  gotoxy(3, 9); printf("=====>Not find this book!\n");  getchar(); getchar();  return n;  }  }  void DelMember(Member\_Link l)  {  int sel;  Member\_Node\* p, \* r;  char findmess[20];  if (!l->next)  {  system("cls");  mainmenu();  gotoxy(3, 5);  printf("\n====>No Reader record!\n");  getchar();  return;  }  system("cls");  mainmenu();  gotoxy(3, 5);  printf("=====>l Delete by Member number =====>2 Delete by Member name");  gotoxy(3, 6);  printf("please choice[1,2]:");  scanf("%d", &sel);  if (sel == 1)  {  gotoxy(3, 7);  stringinput(findmess, 10, "input the existing member number:",0);  p = LocateMember(l, findmess, "num");  if (p)//p!=NULL  {  r = l;  while (r->next != p)  r = r->next;  r->next = p->next;//将p所指节点从链表中去除  free(p);//释放内存空间  gotoxy(3, 9);  printf("=====>delete success!");  getchar(); getchar();  saveflag = 1;  }  else  {  gotoxy(3, 9); printf("=====>Not find this Member!\n");  getchar(); getchar();  }  }  else if (sel == 2)//先按顾客名查询到该记录所在的节点  {  stringinput(findmess, 15, "input the existing member name:",0);  p = LocateMember(l, findmess, "name");  if (p)  {  r = l;  while (r->next != p)  r = r->next;  r->next = p->next;  free(p);  gotoxy(3, 9);  printf("=====>delete success!\n");  getchar(); getchar();  saveflag = 1;  }  else  {  gotoxy(3, 9); printf("=====>Not find this Member!\n");  getchar(); getchar();  }  }  else  {  gotoxy(3, 9); printf("\*\*\*\*\*Error:input has wrong! press any key to continue\*\*\*\*\*\*");  getchar(); getchar();  }  }  //保存记录  void SaveRoom(ROOM tp[],int n )  {  FILE\* fp;  int i=0;  int count = 0;  mainmenu();  fp = fopen("E:\\room", "wt");//以只写方式打开二进制文件  if (fp == NULL)//打开文件失败  {  system("cls");  mainmenu();  gotoxy(3, 5);  printf("=====>open file error!\n");  getchar();  return;  }  i = 0;  while (i<n)  {  if (fwrite(&tp[i], sizeof(ROOM), 1, fp) == 1)//每次写一条记录或一个节点信息至文件  {  count++;  i++;  continue;  }  else  {  break;  }  }  if (count > 0)  {  gotoxy(3, 8);  printf("=====>save book,total saved record number is:%d\n", count);  getchar();  saveflag = 0;  }  else  {  system("cls");  mainmenu();  gotoxy(3, 10);  printf("the current room link is empty,no record is saved!\n");  getchar();  }  fclose(fp);//关闭book文件  }  void SaveMember(Member\_Link l)  {  FILE\* fp;  Member\_Node\* p;  int count = 0;  mainmenu();  fp = fopen("E:\\member", "wb");//以只写方式打开二进制文件  if (fp == NULL)//打开文件失败  {  system("cls");  mainmenu();  gotoxy(3, 5);  printf("=====>open file error!\n");  getchar();  return;  }  p = l->next;  while (p)  {  if (fwrite(p, sizeof(Member\_Node), 1, fp) == 1)//每次写一条记录或一个节点信息至文件  {  p = p->next;  count++;  }  else  {  break;  }  }  if (count > 0)  {  gotoxy(3, 9);  printf("=====>save member,total saved record number is:%d\n", count);  getchar();  saveflag = 0;  }  else  {  system("cls");  mainmenu();  gotoxy(3, 10);  printf("the current member link is empty,no record is saved!\n");  getchar();  }  fclose(fp);//关闭reader文件  }  //查询记录  void QueryRoom(ROOM tp[],int n)//按图书编号或图书书名查询  {  int select;//1:按客房编号查 2：按顾客名查，其他：返回主界面  char searchinput[20];//保存用户输入的查询内容  int p=0;  if (n<0)//若链表为空  {  system("cls");  mainmenu();  gotoxy(3, 5);  printf("\n====>No Room Record!\n");  getchar(); getchar();  return;  }  system("cls");  mainmenu();  gotoxy(3, 5);  printf("=====>l Search by Room number =====>2 Search by client name");  gotoxy(3, 6);  printf("please choice[1,2]:");  scanf("%d", &select);  if (select == 1)//按客房编号查询  {  gotoxy(3, 7);  stringinput(searchinput, 15, "input the existing room number:",0);  p = LocateRoom(tp,n,searchinput,"num");  //在l中查找客房编号为searchinput值的节点，并返回节点的指针  if (p!=-1)//若p!=NULL  {  gotoxy(3, 8); printf("------------------------------------------------------");  gotoxy(3, 9); printf("Room Number:%s", tp[p].num);  gotoxy(3, 10); printf("Room Price:%d", tp[p].price);  gotoxy(3, 11); printf("Room Empty\_flag(1:Empty):%d", tp[p].empty\_flag);  gotoxy(3, 12); printf("Client\_name:%s", tp[p].client\_name);  gotoxy(3, 13); printf("Client\_ic:%s", tp[p].client\_ic);  gotoxy(3, 14); printf("Client\_date,%d-%d-%d", tp[p].client\_date.year,tp[p].client\_date.mon,tp[p].client\_date.day);  gotoxy(3, 15); printf("Client\_totaldays:%d",tp[p].client\_totaldays);  gotoxy(3, 16); printf("Client\_ap:%d",tp[p].client\_ap);  gotoxy(3, 17); printf("------------------------------------------------------");  gotoxy(3, 18); printf("press any key to return");  getchar(); getchar();  }  else  {  gotoxy(3, 8); printf("====>Not find this room!\n");  getchar(); getchar();  }  }  else if (select == 2)//按图书名查询  {  gotoxy(3, 7);  stringinput(searchinput, 15, "input the existing book name:",0);  p = LocateRoom(tp, n, searchinput,"name"); /\*用户选择2，通过名字查找，即在name字段中查找\*/  if (p!=-1)  {  gotoxy(3, 8); printf("------------------------------------------------------");  gotoxy(3, 9); printf("Room Number:%s", tp[p].num);  gotoxy(3, 10); printf("Room Price:%d", tp[p].price);  gotoxy(3, 11); printf("Room Empty\_flag(1:Empty):%d", tp[p].empty\_flag);  gotoxy(3, 12); printf("Client\_name:%s", tp[p].client\_name);  gotoxy(3, 13); printf("Client\_ic:%s", tp[p].client\_ic);  gotoxy(3, 14); printf("Client\_date,%d-%d-%d", tp[p].client\_date.year, tp[p].client\_date.mon, tp[p].client\_date.day);  gotoxy(3, 15); printf("Client\_totaldays:%d", tp[p].client\_totaldays);  gotoxy(3, 16); printf("Client\_ap:%d", tp[p].client\_ap);  gotoxy(3, 17); printf("------------------------------------------------------");  gotoxy(3, 18); printf("press any key to return");  getchar(); getchar();  }  else  {  gotoxy(3, 8); printf("====>Not find this room!\n");  getchar(); getchar();  }  }  else  {  gotoxy(3, 8); printf("\*\*\*\*\*Error:input has wrong! press any key to continue\*\*\*\*\*\*");  getchar(); getchar();  }  }  void QueryMember(Member\_Link l)  {  int select;//1:按图书编号查 2：按图书名查，其他：返回主界面  char searchinput[20];//保存用户输入的查询内容  Member\_Node\* p;  if (!l->next)//若链表为空  {  system("cls");  mainmenu();  gotoxy(3, 5);  printf("\n====>No Member Record!\n");  getchar(); getchar();  return;  }  system("cls");  mainmenu();  gotoxy(3, 5);  printf("=====>l Query by Member number =====>2 Query by Member name");  gotoxy(3, 6);  printf("please choice[1,2]:");  scanf("%d", &select);  if (select == 1)//按读者编号查询  {  gotoxy(3, 7);  stringinput(searchinput, 15, "input the existing reader number:",0);  p = LocateMember(l, searchinput, "num");  //在l中查找图书编号为searchinput值的节点，并返回节点的指针  if (p)//若p!=NULL  {  gotoxy(3, 8); printf("------------------------------------------------------");  gotoxy(3, 9); printf("Member Number:%s", p->data.num);  gotoxy(3, 10); printf("Member Name:%s", p->data.name);  gotoxy(3, 11); printf("Member IC", p->data.ic);  gotoxy(3, 12); printf("Member Sex:%s", p->data.sex);  gotoxy(3, 13); printf("Member Age:%d", p->data.age);  gotoxy(3, 14); printf("Member Tele:%s", p->data.tele);  gotoxy(3, 15); printf("------------------------------------------------------");  gotoxy(3, 18); printf("press any key to return");  getchar(); getchar();  }  else  {  gotoxy(3, 8); printf("====>Not find this Member!\n");  getchar(); getchar();  }  }  else if (select == 2)//按读者名查询  {  gotoxy(3, 7);  stringinput(searchinput, 15, "input the existing reader name:",0);  p = LocateMember(l, searchinput, "name");  if (p)  {  gotoxy(3, 8); printf("------------------------------------------------------");  gotoxy(3, 9); printf("Member Number:%s", p->data.num);  gotoxy(3, 10); printf("Member Name:%s", p->data.name);  gotoxy(3, 11); printf("Member IC", p->data.ic);  gotoxy(3, 12); printf("Member Sex:%s", p->data.sex);    //修改记录  void ModifyRoom(ROOM tp[],int n)  {  int p = 0;  char findmess[20];  if (n<0)  {  system("cls");  mainmenu();  gotoxy(3, 5);  printf("\n=====>No room record!\n");  getchar();  return;  }  system("cls");  mainmenu();  gotoxy(3, 5);  stringinput(findmess, 10, "input the existing room number；",0);//输入并检验该图书编号  p = LocateRoom(tp, n,findmess, "num");//查询该节点  if (p)//若p!=NULL,表明已经找到该节点  {  gotoxy(3, 6); printf("------------------------------------------------------");  gotoxy(3, 7); printf("Room Number:%s", tp[p].num);  gotoxy(3, 8); printf("Room Price:%d", tp[p].price);  gotoxy(3, 9); printf("Room Empty\_flag(1:Empty):%d", tp[p].empty\_flag);  gotoxy(3, 10); printf("Client\_name:%s", tp[p].client\_name);  gotoxy(3, 11); printf("Client\_ic:%s", tp[p].client\_ic);  gotoxy(3, 12); printf("Client\_date:%d-%d-%d",tp[p].client\_date.year, tp[p].client\_date.mon, tp[p].client\_date.day);  gotoxy(3, 13); printf("Client\_totaldays:%d", tp[p].client\_totaldays);  gotoxy(3, 14); printf("Client\_ap:%d", tp[p].client\_ap);  gotoxy(3, 15); printf("------------------------------------------------------");  getchar(); getchar();  gotoxy(3, 16); printf("please modify room recorder:");  gotoxy(3, 17); printf("Room Number:%s", tp[p].num);  gotoxy(3, 18); printf("Room Price:"); scanf("%d", &tp[p].price);  if (tp[p].empty\_flag == 0)  {  gotoxy(3,19); stringinput(tp[p].client\_name, 15, "Client\_name:",0);  gotoxy(3, 20); stringinput(tp[p].client\_ic, 20, "Client\_ic",3);  gotoxy(3, 21); printf("Client\_Register\_Date:%d-%d-%d", tp[p].client\_date.year, tp[p].client\_date.mon, tp[p].client\_date.day);  gotoxy(3, 22); printf("Client\_totaldays:"); scanf("%d", &tp[p].client\_totaldays);  gotoxy(3, 23); printf("Client\_ap(>=%d):", tp[p].price \* tp[p].client\_totaldays + 100); scanf("%d", &tp[p].client\_ap);  gotoxy(3, 24); printf("=====> Modify success!"); getchar(); getchar();  saveflag = 1;  }  else {  gotoxy(3, 19); printf("------------------------------------------------------");  gotoxy(3, 20); printf("=====> Modify success!"); getchar(); getchar();  }    }  else  {  gotoxy(3, 6); printf("=====>Not find this room!\n"); getchar(); getchar();  }  }  void ModifyMember(Member\_Link l)  {  Member\_Node\* p;  char findmess[20];  int temp;  if (!l->next)  {  system("cls");  mainmenu();  gotoxy(3, 5);  printf("\n=====>No Reader record!\n");  getchar();  return;  }  system("cls");  mainmenu();  gotoxy(3, 5);  stringinput(findmess, 10, "input the existing Reader number；",0);//输入并检验该读者编号  p = LocateMember(l, findmess, "num");//查询该节点  if (p)//若p!=NULL,表明已经找到该节点  {  gotoxy(3, 6); printf("------------------------------------------------------");  gotoxy(3, 9); printf("Member Number:%s", p->data.num);  gotoxy(3, 10); printf("Member Name:%s", p->data.name);  gotoxy(3, 11); printf("Member Sex:%s", p->data.sex);  gotoxy(3, 12); printf("Member Age:%d", p->data.age);  gotoxy(3, 13); printf("Member Tele:%s", p->data.tele);  gotoxy(3, 14); printf("------------------------------------------------------");  gotoxy(3, 15); printf("please modify Member recorder:");  gotoxy(3, 16); stringinput(p->data.name, 15, "Member Name:",0);  gotoxy(3, 17); stringinput(p->data.sex, 15, "Member Sex:",1);  gotoxy(3, 18); printf("Member Age:"); scanf("%d", &temp); p->data.age = temp;  gotoxy(3, 19); stringinput(p->data.tele, 15, "Member Tele:",2);  gotoxy(3, 20); printf("------------------------------------------------------");  gotoxy(3, 21); printf("=====>modify success!"); getchar(); getchar();  saveflag = 1;  }  else  {  gotoxy(3, 6); printf("=====>Not find this Member!\n"); getchar(); getchar();  }  }  //排序图书记录  void SortRoom(ROOM tp[],int n) //利用直接选择排序法实现按客房价格字段的升序排序，从低到高  {  int i = 0, j = 0, k = 0, flag = 0;  int x, y;//保存当前光标所在位置的坐标值  ROOM newinfo;  ROOM\* lll;  system("cls");  if (n<=0)  {  system("cls");  mainmenu();  gotoxy(3, 5);  printf("=====>Not roomrecord!");  getchar();  return;  }  //显示排序前的所有记录  system("cls");  mainmenu();  gotoxy(3, 4);  printf(HEADER1);  gotoxy(3, 4);  x = wherex(); y = wherey(); k = 0; i = 0;  while (i<n)  {  k++;  gotoxy(x,k+y);//换行  lll = &tp[i];  printf(FORMAT1, DATA1);//见头部宏定义  i++;  }  getchar(); getchar();  gotoxy(3, y + i+2);  y=y + i+2;  printf("=====>sort by price..............");  //排序  for (i = 0; i < n; i++)  {  flag = 0;  for (j = 0; j < n - 1; j++)  if ((tp[j].price > tp[j + 1].price))  {  flag = 1;  strcpy(newinfo.num, tp[j].num);  newinfo.price = tp[j].price;  newinfo.empty\_flag = tp[j].empty\_flag;  strcpy(newinfo.client\_name, tp[j].client\_name);  strcpy(newinfo.client\_ic, tp[j].client\_ic);  newinfo.client\_date.year = tp[j].client\_date.year;  newinfo.client\_date.mon = tp[j].client\_date.mon;  newinfo.client\_date.day = tp[j].client\_date.day;  newinfo.client\_totaldays = tp[j].client\_totaldays;  newinfo.client\_ap = tp[j].client\_ap;  strcpy(tp[j].num, tp[j + 1].num);  tp[j].price=tp[j + 1].price;  tp[j].empty\_flag=tp[j + 1].empty\_flag;  strcpy(tp[j].client\_name, tp[j + 1].client\_name);  strcpy(tp[j].client\_ic, tp[j + 1].client\_ic);;  tp[j].client\_date.year = tp[j + 1].client\_date.year;  tp[j].client\_date.mon = tp[j + 1].client\_date.mon;  tp[j].client\_date.mon = tp[j + 1].client\_date.mon;  tp[j].client\_totaldays = tp[j + 1].client\_totaldays;  tp[j].client\_ap = tp[j + 1].client\_ap;  strcpy(tp[j + 1].num, newinfo.num);  tp[j + 1].price = newinfo.price;  tp[j + 1].empty\_flag = newinfo.empty\_flag;  strcpy(tp[j + 1].client\_name, newinfo.client\_name);  strcpy(tp[j + 1].client\_ic, newinfo.client\_ic);  tp[j + 1].client\_date.year = newinfo.client\_date.year;  tp[j + 1].client\_date.mon = newinfo.client\_date.mon;  tp[j + 1].client\_date.mon = newinfo.client\_date.day;  tp[j + 1].client\_totaldays = newinfo.client\_totaldays;  tp[j + 1].client\_ap = newinfo.client\_ap;  }  if (flag == 0) break;  }  k=0;i=0;  while (i<n)  {  k++;  gotoxy(3,y+k+1);  lll = &tp[i];  printf(FORMAT1, DATA1);  i++;  };  saveflag = 1;  gotoxy(3,y+k+3);  printf("=====>sort complete!");  getchar(); getchar();  return;  }  void SortMember(Member\_Link l)  {  Member\_Link lll;//临时指针  Member\_Node\* p, \* q, \* r, \* s, \* h1;//临时指针  int x, y;//保存当前光标所在位置的坐标值  int i = 0;  if (l->next == NULL)  {  system("cls");  mainmenu();  gotoxy(3, 5);  printf("\n=====>Not Member record!");  getchar();  return;  }  h1 = p = (Member\_Node\*)malloc(sizeof(Member\_Node));//用于创建新的头节点  if (!p)  {  gotoxy(3, 5);  printf("allocate memory failure");//如果没有申请到打印信息  return;//返回主界面  }  //显示排序前的所有记录  system("cls");  mainmenu();  gotoxy(3, 4);  printf(HEADER2);  lll = l->next;  x = wherex(); y = wherey(); i = 0;  while (lll != NULL)//当p不为空时，进行下列操作  {  gotoxy(3, i + y);//换行  i++;  printf(FORMAT2, DATA2);//见头部宏定义  lll = lll->next;//指针后移  }  getchar(); getchar();  gotoxy(3, y + i+1);y=y+i+1;  printf("=====>sort by number..............\n");  //排序  p->next = l->next;//l所指节点为不存有任何记录的节点，下一个节点才有图书记录  while (p->next != NULL)//外层循环决定待排序位置  {  q = p->next;  r = p;  while (q->next != NULL)//内部循环找到当前关键字最小节点  {  if (strlen(q->next->data.num) < strlen(r->next->data.num))  {  r = q; q = q->next;  }//移至下一个节点  else if (strlen(q->next->data.num) == strlen(r->next->data.num) && strcmp(q->next->data.num, r->next->data.num) < 0)  {  r = q; q = q->next;  }//移至下一个节点  else q = q->next;  }  if (r != p)//表示原来的第一个节点不是关键字最小的节点，改变指针关系，将关键字最小的节点与本轮循环的首节点进行位置互换  {  s = r->next;//s指向最小节点  r->next = s->next;//r的指针域指向最小节点的下一个节点  s->next = p->next;//s的指针域指向当前p指针所指的下一个节点  p->next = s;//p的指针域指向本次循环结束后关键字最小的节点  }  p = p->next;//移至下一个节点  }  l->next = h1->next;//将排序好的链表首节点地址赋给原来链表的指针域  lll = l->next;  i=0;  while (lll != NULL)//当p不为空时，进行下列操作  {  i++;  gotoxy(3, y+i+1);  printf(FORMAT2, DATA2);  lll = lll->next;//指针后移  }  free(h1);  saveflag = 1;  gotoxy(3, y+i+3);  printf("=====>sort complete!");  getchar(); getchar();  return;  }  //统计记录  int isLeapyear(int year) /\*判断是否是闰年\*/  {  if (year % 4 == 0 && year % 100 || year % 400 == 0)  return 1;  else  return 0;  }  void CountRoom(ROOM tp[], int n)  {  int i = 0, j = 0, k = 0, year = 0, month = 0, leapFlag = 0;  int totalday = 0;  int count500 = 0, count200 = 0, count0 = 0;  int count500e = 0, count200e = 0, count0e = 0;  int a[13] = { 0,31,28,31,30,31,30,31,31,30,31,30,31 };  int b[13] = { 0,31,29,31,30,31,30,31,31,30,31,30,31 };  char roomnum[300];  char dot[4];  time\_t timep;  struct tm\* p;  time(&timep);  p=gmtime(&timep);  system("cls");  if (n <= 0)  {  system("cls");  mainmenu();  gotoxy(3, 5);  printf("=====>Not room record!");  getchar();  return;  }  /\*以当前系统时间重新赋值\*/  currentYear = 1900 + p->tm\_year;  currentMonth = 1 + p->tm\_mon;  currentDay = p->tm\_mday;  int current\_sec = p->tm\_sec; /\*获取当前秒\*/  int current\_min = p->tm\_min;/\*获取当前分\*/  int current\_hour = 8 + p->tm\_hour;/\*获取当前时\*/  i = 0;  while (i < n) /\*统计各档次客房的总数\*/  {  if (tp[i].price >= 500) { count500++; i++; continue; }  if (tp[i].price >= 200) { count200++; i++; continue; }  if (tp[i].price < 200) { count0++; i++; continue; }  }  i = 0;  while (i < n)/\*统计各档次客房的总数空闲值\*/  {  if (tp[i].price >= 500 && tp[i].empty\_flag == 1) { count500e++; i++; continue; }  if (tp[i].price >= 200 && tp[i].empty\_flag == 1) { count200e++; i++; continue; }  if (tp[i].price < 200 && tp[i].empty\_flag == 1) { count0e++; i++; continue; }  i++;  }  i = 0; strcpy(dot, ";"); strcpy(roomnum, " ");  while (i < n)/\*统计入住天数超期的客房号\*/  {  if (tp[i].empty\_flag == 0)  {  year = currentYear - tp[i].client\_date.year;  if (year != 0)  {  for (j = 2; j <= year; j++)  {  leapFlag = isLeapyear(tp[i].client\_date.year + j - 1);/\*判断是否是闰年\*/  if (leapFlag == 1)  for (k = 1; k <= 12; k++) { totalday = totalday + b[k]; }  else  for (k = 1; k <= 12; k++) { totalday = totalday + a[k]; }  }  leapFlag = isLeapyear(tp[i].client\_date.year);  if (leapFlag == 1)  for (k = 12; k <= tp[i].client\_date.mon; k--) { totalday = totalday + b[k]; }  else  for (k = 12; k <= tp[i].client\_date.mon; k--) { totalday = totalday + a[k]; }  totalday = totalday - tp[i].client\_date.day;  leapFlag = isLeapyear(currentYear);  if (leapFlag == 1)  for (k = 1; k <= currentMonth; k++) { totalday = totalday + b[k]; }  else  for (k = 1; k <= currentMonth; k++) { totalday = totalday + a[k]; }  totalday = totalday + tp[i].client\_date.day;  }  else  {  month = currentMonth - tp[i].client\_date.mon;  if (month != 0)  {  for (j = 2; j <= month; j++)  {  leapFlag = isLeapyear(tp[i].client\_date.year + j - 1);  if (leapFlag == 1)  {  totalday = totalday + b[tp[i].client\_date.mon + j - 1];  }  else  {  totalday = totalday + a[tp[i].client\_date.mon + j - 1];  }  }  leapFlag = isLeapyear(tp[i].client\_date.year);  if (leapFlag == 1)  {  totalday = totalday + b[tp[i].client\_date.mon] - tp[i].client\_date.day;  }  else  {  totalday = totalday + a[tp[i].client\_date.mon] - tp[i].client\_date.day;  }  totalday = totalday + currentDay;  }  else { totalday = currentDay - tp[i].client\_date.day; }  }  if (totalday > tp[i].client\_totaldays)  {  strcpy(roomnum, strcat(roomnum, tp[i].num)); strcat(roomnum, dot);  }  }  i++;  }  system("cls");  mainmenu();  gotoxy(3, 5); printf("-----the statistics result of Room-----");  gotoxy(3, 6); printf("Total number of rooms:%d", n);  gotoxy(3, 7); printf("Total number of price>=500:%d empty:%d", count500, count500e);  gotoxy(3, 8); printf("Total number of 200<=price<500:%d empty:%d", count200, count200e);  gotoxy(3, 9); printf("Total number of price<200:%d empty:%d", count0, count0e);  gotoxy(3,10);printf("List of room number exceeding the prospective day:%s",roomnum);  gotoxy(3, 11); printf("---------------------------------------");  getchar(); getchar();  }  void CountMember(Member\_Link l)//总数量、男性顾客数量、女性顾客数量  {  Member\_Node\* r = l->next;  int countc = 0, countm = 0, countf = 0, counte = 0;  char Readername[15];  if (!r)  {  system("cls");  mainmenu();  gotoxy(3, 5);  printf("=====>Not Member record!");  getchar();  return;  }  strcpy(Readername, r->data.name);  while (r)  {  countc++;//统计住客数量  if (strcmp(r->data.sex, "男") == 0) countm++;//统计男性  else countf++;//统计女性数量  r = r->next;  }  system("cls");  mainmenu();  gotoxy(3, 5);  printf("-----------------------the statistics result--------------------------");  gotoxy(3, 6); printf("Total number of Members:%d", countc);  gotoxy(3, 7); printf("Total number of male Members:%d", countm);  gotoxy(3, 8); printf("Total number of female Members:%d", countf);  gotoxy(3, 10); printf("-------------------------------------------------------------------");  getchar(); getchar();  }  //登记入住  void RegisterRoom(ROOM tp[], int n, Member\_Link ll)  {  Member\_Node\* p2; /\*定义会员记录指针变量\*/  ROOM\* lll;  char membernum[15], roomnum[15], ch;  int flag = 0, i, x, y, k, client\_price, staydays, ad\_pay, suffix = 0, ans = 0;  p2 = ll->next;  time\_t timep;  struct tm\* p;  time(&timep);  p = gmtime(&timep);  system("cls");  mainmenu();  gotoxy(3, 5);  printf("please input your upper limit of room price:");/\*输入想要房间的价位\*/  scanf("%d", &client\_price);  gotoxy(3, 5);  printf(HEADER1);  gotoxy(3, 6);  x = wherex(); y = wherey();  k = 0; i = 0;  while (i < n)  {  if (tp[i].empty\_flag == 1 && tp[i].price <= client\_price)\  {  k++;  gotoxy(3, k + y);  lll = &tp[i];  printf(FORMAT1, DATA1);  flag = 1;  suffix = i;/\*保存最后一条符合记录的客房记录的下标值，准备分配给客户\*/  }  i++;  }  if (flag == 0)  {  system("cls");  mainmenu();  gotoxy(3, 5);  cprintf("=====>Sorry ,there is no empty room your wanted!");  getchar(); getchar();  return;  }  getchar();  gotoxy(3, y + k + 2); x = wherex(); y = wherey();  flag = 0;  while (1)  {  if (!ans)  {  cprintf("=====>Are you the member of hotel?(y/n):");  }  else  {  gotoxy(3, y - 1);insline1();  gotoxy(3, y - 1);cprintf("=====>Are you the member of hotel?(Please input y/n):");  ans--;  }  scanf("%c", &ch);  if (ch == 'y' || ch == 'Y')  {  gotoxy(3, y + 1);  stringinput(membernum, 3, "Member Number:", 4);  while (p2)/\*查询该会员编号是否已经存在，若不存在则不允许执行会员登记住宿操作\*/  {  if (strcmp(p2->data.num, membernum) == 0)  {  flag = 1;  break;  }  p2 = p2->next;  }  if (flag == 0)  {  gotoxy(3, y + 2);  printf("The Member Num %s is not existing!", membernum);  getchar(); getchar();  return;  }  gotoxy(3, y + 2);  printf("The days of your stay:");  scanf("%d", &staydays);  gotoxy(3, y + 3);  printf("Please input advance payment[>=%d]:", tp[suffix].price \* staydays + 100);  scanf("%d", &ad\_pay);  getchar(); getchar();  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*将会员资料填入客房记录中\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  tp[suffix].empty\_flag = 0;  strcpy(tp[suffix].client\_name, p2->data.name);  strcpy(tp[suffix].client\_ic, p2->data.ic);  tp[suffix].client\_totaldays = staydays;  tp[suffix].client\_ap = ad\_pay;  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*处理日期\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  currentYear = p->tm\_year+1900;  currentMonth = p->tm\_mon+1;  currentDay = p->tm\_mday;  tp[suffix].client\_date.year = currentYear;  tp[suffix].client\_date.mon = currentMonth;  tp[suffix].client\_date.day = currentDay;  saveflag = 1;  gotoxy(3, y + 4);  printf("Register OK! Room %s is distributed to %s", tp[suffix].num, tp[suffix].client\_name);  getchar(); getchar();  break;  }  else if (ch == 'n' || ch == 'N')/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*非会员\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  {  gotoxy(3, y + 1); stringinput(tp[suffix].client\_name, 15, "Client\_name:", 0);  gotoxy(3, y + 2); stringinput(tp[suffix].client\_ic, 18, "Client\_ic:", 3);  gotoxy(3, y + 3); printf("The days of your stay:"); scanf("%d", &staydays);  gotoxy(3, y + 4); printf("Please input advance payment[>=%d]:", tp[suffix].price \* staydays + 100); scanf("%d", &ad\_pay);  tp[suffix].empty\_flag = 0;  tp[suffix].client\_totaldays = staydays;  tp[suffix].client\_ap = ad\_pay;  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*处理日期\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  currentYear = p->tm\_year+1900;  currentMonth = p->tm\_mon+1;  currentDay = p->tm\_mday;  tp[suffix].client\_date.year = currentYear;  tp[suffix].client\_date.mon = currentMonth;  tp[suffix].client\_date.day = currentDay;  saveflag = 1;  gotoxy(3, y + 6);  printf("Register OK! Room %s is distributed to %s", tp[suffix].num, tp[suffix].client\_name);  getchar(); getchar();  break;  }  else  {  ans++;  }  }  return;  }  //退房  void ReturnRoom(ROOM tp[], int n, Member\_Link ll)  {  Member\_Node\* p2;  ROOM\* lll;  char membernum[15], roomnum[15], clientname[15], ch;  int flag = 0, i, x, y, dtaydays, suffix = -1,ans=0;  float return\_pay;  p2 = ll->next;  system("cls");  mainmenu();  gotoxy(3, 5); printf("Please input your number of returned room:");  scanf("%s", &roomnum);  i = 0;  while (i < n)  {  if (strcmp(tp[i].num, roomnum) == 0)  {  suffix = i; break;  }  i++;  }  if (suffix == -1)  {  gotoxy(3, 6); printf("Room %s is not existing!", roomnum); getchar(); getchar();  return;  }  if (tp[suffix].empty\_flag == 1)  {  gotoxy(3, 6); printf("Room %s is empty!"); getchar(); getchar();  return;  }  getchar(); gotoxy(3, 6);  y = wherey();  flag = 0;  while (1)  {  if (!ans)  {  cprintf("=====>Are you the member of hotel?(y/n):");  }  else  {  gotoxy(3, y - 1); insline1();  gotoxy(3, y - 1); cprintf("=====>Are you the member of hotel?(Please input y/n):");  ans--;  }  scanf("%c", &ch);  if (ch == 'y' || ch == 'Y')  {  gotoxy(3, y + 1);  stringinput(membernum, 3, "Member Number:", 4);  while (p2)/\*查询该会员是否已经存在，若不存在则不允许执行会员退房操作\*/  {  if (strcmp(p2->data.num, membernum) == 0)  {  flag = 1;  break;  }  p2 = p2->next;  }  if (flag == 0)  {  gotoxy(3, y + 2); printf("The Member %s is not existing!", membernum);  getchar(); getchar(); return;  }  /\*结账退房\*/  gotoxy(3, y + 2);  printf("The days of your stay:%d,price:%d", tp[suffix].client\_totaldays, tp[suffix].price);  gotoxy(3, y + 3);  return\_pay = tp[suffix].client\_ap - tp[suffix].client\_totaldays \* tp[suffix].price \* 0.9;/\*会员九折优惠\*/  printf("Advance payment:%d,Return payment:%.2f", tp[suffix].client\_ap, return\_pay);  getchar(); getchar();  strcpy(clientname, tp[suffix].client\_name);  break;  }  else if (ch == 'n' || ch == 'N')/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*非会员\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  {  /\*结账退房\*/  gotoxy(3, y + 2);  printf("The days of your stay:%d,price:%d", tp[suffix].client\_totaldays, tp[suffix].price);  gotoxy(3, y + 3);  return\_pay = tp[suffix].client\_ap - tp[suffix].client\_totaldays \* tp[suffix].price;  printf("Advance payment:%d,Return payment:%.2f", tp[suffix].client\_ap, return\_pay);  getchar(); getchar();  strcpy(clientname, tp[suffix].client\_name);  getchar(); getchar();  break;  }  else  {  ans++;  }  }  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*将客房记录置空\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  tp[suffix].empty\_flag = 1;  strcpy(tp[suffix].client\_name, " ");  strcpy(tp[suffix].client\_ic, " ");  tp[suffix].client\_totaldays = 0;  tp[suffix].client\_ap = 0;  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*处理日期\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  tp[suffix].client\_date.year = 0;  tp[suffix].client\_date.mon = 0;  tp[suffix].client\_date.day = 0;  saveflag = 1;  gotoxy(3, y + 4); printf("Room %s is returned bu %s successfully!", tp[suffix].num, clientname);  getchar(); getchar();  return;  }  gotoxy(3, 13); printf("Member Age:%d", p->data.age);  gotoxy(3, 14); printf("Member Tele:%s", p->data.tele);  gotoxy(3, 15); printf("------------------------------------------------------");  gotoxy(3, 18); printf("press any key to return");  getchar(); getchar();  }  else  {  gotoxy(3, 8); printf("====>Not find this Member!\n");  getchar(); getchar();  }  }  else  {  gotoxy(3, 8); printf("\*\*\*\*\*Error:input has wrong! press any key to continue\*\*\*\*\*\*");  getchar(); getchar();  }  } |