中国地质大学(武汉)

C语言课程设计报告书

姓 名：万宇扬

专 业：数学类

学 号：20201001509

# 房地产登记系统

## 一、背景

房地产开发商使用房产销售管理系统对住宅房产的销售进行管理。住宅开发项目的每套房产都有唯一的编号。 销售部对所销售的房源存有如下信息:编号、朝向楼层(总层数\_层位)、面积、价格。销售部每天接待客户的电话咨询和来访咨询，向客户介绍房产的信息，记录客户的电话号码、姓名和有购买意向的住房编号;销售部还对咨询过的客户进行电话回访.并记录回访时间与次数:销售部将客户咨询与回访情况，作为预测项目销售前景与制订推销对策的依据:销售部还负责与客户签订购房合同。

## 二、需求分析

一、编写住宅房产销售管理系统，提供系统菜单，实现房产数据的查询、统计和销售管理。销售管理系统使用的数据文件有:

(1)文本文件source. txt存储房源信息，每套房产的信息有五项内容:编号、朝向、楼层

(2)文本文件guest.txt 记录客户的信息，每位客户的信息包含以下内容:电话、姓名、有购买意向的住房编号(最多保存3个)、通话次数、最近电话回访日期。

(3) 文本文件sold. txt记录已销售 房产的信息.每套已销售房产的信息有以下内容:编号.面积、价格、售出时间、合同号、房主姓名、房主电话。

二、销售管理系统将实现的主要功能有:

(1)新客户登记:登记首次前来咨询客户的电话姓名和有购买意向的住房编号(最多可以记录3个不同房号)。新客户信息添加到guest.txt文件。

(2) 老客户信息修改:增加、修改或删除有购买意向的房号:修改最近电话回访日期:修改通话次数;修改客户的电话，修改后的客户信息存储到guest. txt文件。

(3) 提供房源信息查询:输人朝向楼层.面积范围、价格范围，系统显示出符合条件的

(4) 统计:根据客户文件中“有购买意向的住房编号”，对房源进行排序.找出最受欢迎房源前10名，输出其各项指标。

(5)登记已经签订了销售合同的房源.将其信息添加到已销售房源文件sold. xt.

定义房产结构体.其成员除了房产信息文件中的五项内容之外，可以再用来统计有购买意向的客户数。

## 三、总体设计

根据需求分析分别设计了本程序使用的数据类型、数据存储方法与程序功能模块。

1. 数据类型

利用链表数据类型灵活占用空间的特性，在程序内部采用单链表的储存方法；定义房源、客户、合同三种结构如下：

1.房源

typedef struct estate\_temp{

int id;

int sold;

int direction;

int f\_level;

int size;

int price;

struct estate\_temp \*next;

char infor[ISIZE];

}estate;

2.客户

typedef struct guest\_temp{

char p\_number[PSIZE];

char name[NSIZE];

int call\_number;

int last\_call[3];

struct guest\_temp \*next;

int order[3];//order最大储存三个数据，0表示无数据存于此处

}guest;

3.合同

typedef struct sold\_temp{

int size;

int price;

int time[3];

int id;

int name[NSIZE];

char p\_number[PSIZE];

struct sold\_temp \*next;

}sold\_estate;

1. 存储方式

采用临时储存方法，将输入与输出功能所操作的对象分离，实现文本读入程序形成临时数据库（链表），再由程序操作临时数据库，防止因程序的错误运行而修改本地数据。本地数据以文本文档（后缀.txt）方式储存，按照其结构的对应顺序储存数据。程序所在的文件夹应有以下文档：

source.txt

guest.txt

sold.txt

为了方便读取数据并及时捕获到错位的错误信息，每一条单独的存储首添加标识符，例如：

estate 1 1 1 4 300 40000 小别墅

estate 2 0 2 4 300 40000 小别墅

estate 3 0 3 4 300 5000 经济适用房

estate 4 0 4 4 300 5000 经济适用房

又如：

guest 100011 钟离 1 2020 12 31 1 10 13

guest 100012 胡桃 1 2021 1 1 5 12 0

程序将在读入对应的标识符后将后续数据依次读入临时链表，若数据有误，程序会检查到错误并返回，阻止程序使用错误的操作改动原始数据。





1. 程序功能模块化

由本程序的需求设计，本程序将有一下几大功能，每个功能将封装成一个函数：

1. 新客户添加void add(void)
2. 客户信息修改void change(void)
3. 房源信息查询void source(void)
4. 数据统计void statistics(void)
5. 预定登记void book(void)
6. 保存数据void read()
7. 读取数据void save()
8. 自动保存功能void Auto\_save\_change()
9. 退出程序

# 功能实现

## 一、预处理

将源代码分为struct，basic，function，main四个源文件进行四个层面的编写，其中struct.h用以定义结构体以及全局变量与基本宏，basic.c用以存放底层函数供function.c调用；function.c用以存放功能函数，main函数调用function.c中的函数以实现程序的运行。

三个链表起始点的声明已经在struct.h中完成：

estate \*pestate=NULL;

guest \*pguest=NULL;

sold\_estate \*psold=NULL;

## 二、编写数据处理函数

首先编写save与read函数分别用以读取文本文档数据。Read函数的功能是将文本文档的内容读入程序。打开一个文本文档：

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

fprintf(stderr,"Error:打开文件source.txt失败\n");

exit(-1);

}

读入对应的数据头标识符：

while(fscanf(fp,"%s",buffer)==1){

statement（读取数据）

}

为了将新建链表结点操作封装，在basic.c中定义新建特定节点的函数：

estate\* newnode\_estate(estate \*now){

estate \*newnode;

newnode=(estate \*)malloc(sizeof(estate));

if(newnode==NULL){

puts("newnode fail");

exit(-1);

}

if(now==NULL){

now=newnode;

pestate=newnode;

}

else {

now->next=newnode;

now=now->next;

}

now->sold=0;

now->next=NULL;

return now;

}

当数据读入出现异常，如数据形式不与设定的数据接口匹配时，系统将发出警报并退出到主调函数，该功能用宏实现：

#define WARN {fputs("Error:数据格式错误\n",stderr);return;}

同理进行客户、合同的读入，在程序内临时建立三列临时储存链表。

Save函数的功能是将程序内的临时数据按预设的接口写入文本文档。

//储存source

if((fp=fopen("source.txt","w"))==NULL){

fprintf(stderr,"Error:打开文件失败\n");

exit(-1);

}

estate \*now=pestate;

if(now!=NULL){

fprintf(fp,"estate %d %d %d %d ",now->id,now->sold,now->direction,now->f\_level);

fprintf(fp,"%d %d %s\n",now->size,now->price,now->infor);

while(now->next!=NULL){

now=now->next;

fprintf(fp,"estate %d %d %d %d ",now->id,now->sold,now->direction,now->f\_level);

fprintf(fp,"%d %d %s\n",now->size,now->price,now->infor);

}

}

fclose(fp);

同理将客户与合同链表中的数据写入接口文档。

为了实现自动储存功能的打开和关闭，定义全局变量：

bool Auto\_save\_mode=false;

同时在function.c中定义改变变量Auto\_save\_mode的函数：

void Auto\_save\_change(){

if(Auto\_save\_mode==true){

Auto\_save\_mode=false;

}

else Auto\_save\_mode=true;

}

## 三、实现菜单显示与主程序

编写显示菜单的函数，显示出预期的功能：

char show\_menu\_main(void){

if(Auto\_save\_mode==1)save();

puts("\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n\n");

puts("\*\*\*\*\*欢迎使用房地产管理程序\*\*\*\*\*\n");

puts("1)新客户登记 2)修改客户数据 3)房源数据 \n"); //add/change/source

printf("4)统计 5)预订登记 6)%s自动保存模式\n\n",Auto\_save\_mode?"关闭":"开启"); //save/read/Autp\_save\_change

puts("7)读取数据 8)保存数据 q)退出\n");

puts("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n\n");

char ch='0',c='0';

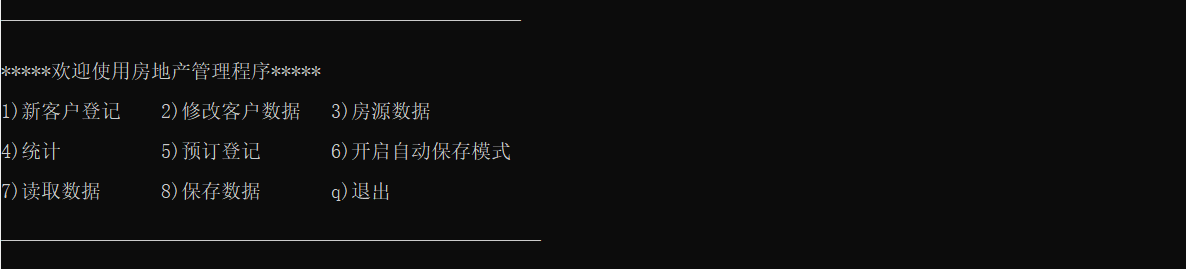
ch=tolower(getchar());

while ((c = getchar()) != EOF && c != '\n');//清空缓存区

//printf("%c %c",ch,c);

return ch;

}



程序的main函数将用while语句循环调用show\_menu\_main函数以读取用户的输入，该输入将会返回给主调函数main，main函数将读取菜单的返回值并传递给switch，将程序引入各功能模块。如下：

while(ch!='q'&&ch!=EOF){

switch (ch=show\_menu\_main()){

case '1':add();

break;

case '2':change();

break;

case '3':source();

break;

case '4':statistics();

break;

case '5':book();

break;

case '6':Auto\_save\_change();

break;

case '7':read();

break;

case '8':save();

break;

default:break;

}

}

考虑到用户体验，在程序退出时增加保存数据的提示：

puts("退出前是否保存？（保存请按一,不保存按任意键）\n");

if(getchar()=='1')save();



## 四、主要功能函数编写

编写add函数，读取用户的输入，调用newnode函数生成新的链表结点，将数据填入临时存储链表。输入后自动显示本次输入的客户信息方便用户检查输入是否有误。

编写change函数，用户将输入想要改变信息的客户名称，程序将顺序检查临时链表中的所有数据直到查找到指定客户。将临时链表指针指向该客户数据，此时调出子菜单，该部分封装成一个函数写入basic.c：

char show\_menu\_change(void){

char ch='0',c='0';

puts("\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n\n");

puts("\n您要修改哪一项数据：\n");

puts("1)电话 2)通话次数 3)最后一次通话时间 \n"); //add/change/source

puts("4)意向房源 5)删除该项 q)返回\n");

puts("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n\n");

ch=tolower(getchar());

while ((c = getchar()) != EOF && c != '\n');//清空缓存区

//printf("%c %c",ch,c);

return ch;

}

其中用户选择删除该项后，程序转入“删除”模块，该模块由basic.c中的del函数实现：

void del(guest \*now){

int i=0;

guest \*finder;

estate \*e\_finder;

for(i=0;i<3;i++){

// printf("success in for i=%d\n",i);

for(e\_finder=pestate;e\_finder!=NULL;e\_finder=e\_finder->next){

// printf("success in for e\_finder->id=%d\n",e\_finder->id);

if(e\_finder->id==now->order[i]){

// printf("id%d sold--\n",e\_finder->id);

e\_finder->sold--;

break;

}

}

}

puts("yes");

if(pguest->next==NULL)pguest=NULL;

else{

for(finder=pguest;finder->next!=now;finder=finder->next);

puts("yes");

finder->next=now->next;

}

free(now);

}

该函数将修改临时链表，释放原来被删除数据对应空间。

编写source函数以显示房源数据。该函数读取用户输入，按用户输入的范围内遍历链表，找到对应房源项，再输出到屏幕上。

编写statistic函数遍历临时链表以统计房源被预订的数量并顺序排列后输出到屏幕上、

编写book函数创建新合同，实现方法与add函数相同。

输出某一指定房源结点的所有数据被封装为一个函数定义在basic.c中：

void print\_estate(estate \*finder){

estate \*now=finder;

printf("代号%3d ",now->id);

printf("卖出%2d套 ",now->sold);

printf("朝%s ",directions[now->direction-1]);

printf("%3d层 ",now->f\_level);

printf("面积%6d ",now->size);

printf("价格%6d ",now->price);

printf("简介：%s\n\n",now->infor);

return;

}

其中，读取用户输入的客户期望房源操作，考虑到程序需要：

一：检索房源是否存在；

二：在房源所在结点处改变sold数据，即房源被预定数要改变；

三：用户至多可输入三个房源。

多个函数都有类似操作，因此将其封装为一个函数定义于basic.c：

void get\_guest\_order(guest \* now)

当读到用户输入非法数据（房源不存在）时，程序将自动忽略该次读入，并用零代替数据位做占位符，下次读取时读到某处guest.order数据为零则认为此处没有数据，程序跳过此处。当读取到用户输入的end字符串时，程序判定输入结束。

puts("请输入客户期望的房源(如有多套，最多输入三套，中间用空格隔开，最后输入\"end\"如：1 10 19 end)");

if(scanf("%s",buffer)!=1){

WARN;

}

if(!strstr(buffer,"end")){

now->order[0]=string\_to\_int(buffer);

statement；

}

# 课设成果总结

## 一、程序流程图

## 二、程序编写心得

鄙人自以为c语言以烂熟于胸，但编写过程中仍然遇到了许多磕绊，第一处是在指针解引用时忽略了指针为NULL的情况，第二处是没有区分开定义的数十个各种类型的指针与变量导致使用混淆，造成程序对错误对象进行操作而崩溃。前后编写了十余小时，七百多行代码，比预期花得时间长的多。我以为终归是不能轻敌的，同时也不能高估了自己的实力，还是虚心学习为上乘。

话说回来，也只有程序设计课上会有很多女生来问我问题，这也是鄙人少有能与异性接触的机会，实属荣幸。

## 三、未来展望

转到软件工程专业以后鄙人便可天天与程序打交道了。

暑假加油学c艹和python (\*^ω^\*)

# 附录：源代码

## struct.h

#include<stdbool.h>

#define NSIZE 10

#define PSIZE 15

#define ISIZE 40

#define SSIZE 20

char \*directions[4]={"南","东","西","北"};

typedef struct estate\_temp{

int id;

int sold;

int direction;

int f\_level;

int size;

int price;

struct estate\_temp \*next;

char infor[ISIZE];

}estate;

typedef struct guest\_temp{

char p\_number[PSIZE];

char name[NSIZE];

int call\_number;

int last\_call[3];

struct guest\_temp \*next;

int order[3];//order最大储存三个数据，0表示无数据存于此处

}guest;

//面积、价格、售出时间、合同号、房主姓名、房主电话

typedef struct sold\_temp{

int size;

int price;

int time[3];

int id;

int name[NSIZE];

char p\_number[PSIZE];

struct sold\_temp \*next;

}sold\_estate;

## basic.c

#include<stdio.h>

#include<ctype.h>

#include<stdlib.h>

#include<string.h>

#include"struct.h"

#define WARN {fputs("Error:数据格式错误\n",stderr);return;}

bool Auto\_save\_mode=false;

char buffer[50];

estate \*pestate=NULL;

guest \*pguest=NULL;

sold\_estate \*psold=NULL;

int string\_to\_int(char \*p){

//将字符串转化为int类型

char ch='0';int i = 0,ret = 0;

while(ch = \*(p+i)){

if(ch=='\0')break;

//printf("检索第%d个字符中\n",i+1);

if(!isdigit(ch)){

printf("error\n");

return 0;

}

ret = ret\*10 + (int)(ch-'0');

i++;

}

printf("ret=%d\n",ret);

return ret;

}

estate\* newnode\_estate(estate \*now){

estate \*newnode;

newnode=(estate \*)malloc(sizeof(estate));

if(newnode==NULL){

puts("newnode fail");

exit(-1);

}

if(now==NULL){

now=newnode;

pestate=newnode;

}

else {

now->next=newnode;

now=now->next;

}

now->sold=0;

now->next=NULL;

return now;

}

guest\* newnode\_guest(guest \*now){

guest \*newnode;

newnode=(guest \*)malloc(sizeof(guest));

if(newnode==NULL){

puts("newnode fail");

exit(-1);

}

if(now==NULL){

now=newnode;

pguest=newnode;

}

else {

now->next=newnode;

now=now->next;

}

now->next=NULL;

return now;

}

sold\_estate\* newnode\_sold(sold\_estate \*now){

sold\_estate \*newnode;

newnode=(sold\_estate \*)malloc(sizeof(sold\_estate));

if(newnode==NULL){

puts("newnode fail");

exit(-1);

}

if(now==NULL){

now=newnode;

psold=newnode;

}

else {

now->next=newnode;

now=now->next;

}

now->next=NULL;

return now;

}

char show\_menu\_change(void){

char ch='0',c='0';

puts("\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n\n");

puts("\n您要修改哪一项数据：\n");

puts("1)电话 2)通话次数 3)最后一次通话时间 \n"); //add/change/source

puts("4)意向房源 5)删除该项 q)返回\n");

puts("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n\n");

ch=tolower(getchar());

while ((c = getchar()) != EOF && c != '\n');//清空缓存区

//printf("%c %c",ch,c);

return ch;

}

void print\_estate(estate \*finder){

estate \*now=finder;

printf("代号%3d ",now->id);

printf("卖出%2d套 ",now->sold);

printf("朝%s ",directions[now->direction-1]);

printf("%3d层 ",now->f\_level);

printf("面积%6d ",now->size);

printf("价格%6d ",now->price);

printf("简介：%s\n\n",now->infor);

return;

}

void del(guest \*now){

int i=0;

guest \*finder;

estate \*e\_finder;

for(i=0;i<3;i++){

// printf("success in for i=%d\n",i);

for(e\_finder=pestate;e\_finder!=NULL;e\_finder=e\_finder->next){

// printf("success in for e\_finder->id=%d\n",e\_finder->id);

if(e\_finder->id==now->order[i]){

// printf("id%d sold--\n",e\_finder->id);

e\_finder->sold--;

break;

}

}

}

puts("yes");

if(pguest->next==NULL)pguest=NULL;

else{

for(finder=pguest;finder->next!=now;finder=finder->next);

puts("yes");

finder->next=now->next;

}

free(now);

}

void get\_guest\_order(guest \* now){

estate \*finder=pestate;

char c;

puts("请输入客户期望的房源(如有多套，最多输入三套，中间用空格隔开，最后输入\"end\"如：1 10 19 end)");

if(scanf("%s",buffer)!=1){

WARN;

}

if(!strstr(buffer,"end")){

now->order[0]=string\_to\_int(buffer);

if(pestate!=NULL){

for(finder=pestate;finder!=NULL;finder=finder->next){

if(finder->id==now->order[0]){

finder->sold++;

break;

}

}

}

if(finder==NULL){

puts("没有找到第一次输入的房源，系统将自动忽略第一次输入\n");

now->order[0]=0;

}

scanf("%s",buffer);

if(!strstr(buffer,"end")){

now->order[1]=string\_to\_int(buffer);

if(pestate!=NULL){

for(finder=pestate;finder!=NULL;finder=finder->next){

if(finder->id==now->order[1]){

finder->sold++;

break;

}

}

}

if(finder==NULL){

puts("没有找到第二次输入的房源，系统将自动忽略第二次输入\n");

now->order[1]=0;

}

scanf("%s",buffer);

if(!strstr(buffer,"end")){

now->order[2]=string\_to\_int(buffer);

if(pestate!=NULL){

for(finder=pestate;finder!=NULL;finder=finder->next){

if(finder->id==now->order[2]){

finder->sold++;

break;

}

}

}

if(finder==NULL){

puts("没有找到第三次输入的房源，系统将自动忽略第三次输入\n");

now->order[2]=0;

}

}

else {

now->order[2]=0;

}

}

else{

now->order[1]=0;

now->order[2]=0;

}

}

else{

now->order[0]=0;

now->order[1]=0;

now->order[2]=0;

}

while ((c = getchar()) != EOF && c != '\n');//清空缓存区

}

## function.c

#include"basic.h"

void add(void){

guest \*now=pguest;

estate \*finder=pestate;

char c;

if(now!=NULL)while(now->next!=NULL)now=now->next;

now=newnode\_guest(now);

puts("请输入客户的电话");

if(scanf("%s",now->p\_number)!=1){

WARN;

}

while ((c = getchar()) != EOF && c != '\n');//清空缓存区

puts("请输入客户的姓名");

if(scanf("%s",now->name)!=1){

WARN;

}

while ((c = getchar()) != EOF && c != '\n');//清空缓存区

puts("请输入与客户通话的次数");

if(scanf("%d",&now->call\_number)!=1){

WARN;

}

while ((c = getchar()) != EOF && c != '\n');//清空缓存区

puts("请输入最后一次与客户通话的时间(年月日用空格隔开，如：2020 12 31)");

if(scanf("%d %d %d",&now->last\_call[0],&now->last\_call[1],&now->last\_call[2])!=3){

WARN;

}

while ((c = getchar()) != EOF && c != '\n');//清空缓存区

get\_guest\_order(now);

printf("保存新客户%s成功",now->name);

while ((c = getchar()) != EOF && c != '\n');//清空缓存区

}

void change(void){

char ch='0',c;

int i=0;

guest \*finder=pguest,\*now;

puts("请输入要改变信息客户的姓名\n");

scanf("%s",buffer);

if(pestate!=NULL){

for(finder=pguest;(finder!=NULL);finder=finder->next)

if(strcmp(finder->name,buffer)==0)break;

}

if(finder==NULL){

puts("没有找到该客户名\n");

while ((c = getchar()) != EOF && c != '\n');//清空缓存区

return;

}

else {

now=finder;

printf("找到用户：%s",finder->name);

printf("电话%9s\n",finder->p\_number);

printf("电话次数%2d\n",finder->call\_number);

printf("上次通话时间%4d年%2d月%2d日\n",finder->last\_call[0],finder->last\_call[1],finder->last\_call[2]);

for(i=0;i<3;i++){

if(now->order[i]!=0)

printf("该买家意向房源%d:%d\n",i+1,now->order[i]);

}

while(ch!='q'&&ch!=EOF){

switch (ch=show\_menu\_change()){

case '1':puts("请输入客户的电话");

if(scanf("%s",now->p\_number)!=1){

WARN;

}

break;

case '2':puts("请输入与客户通话的次数");

if(scanf("%d",&now->call\_number)!=1){

WARN;

}

break;

case '3':puts("请输入最后一次与客户通话的时间(年月日用空格隔开，如：2020 12 31)");

if(scanf("%d %d %d",&now->last\_call[0],&now->last\_call[1],&now->last\_call[2])!=3){

WARN;

}

break;

case '4':get\_guest\_order(now);

break;

case '5':del(now);

default:break;

}

while ((c = getchar()) != EOF && c != '\n');//清空缓存区

if(ch=='5')break;

}

printf("修改客户%s成功",now->name);

}

return;

}

void source(void){

char c;

estate \*finder=pestate;

int token;

if(pestate==NULL){

puts("\nError:没有检测到任何房源\n");

}

struct find\_flag{

int direction;

int f\_level[2];

int size[2];

int price[2];

}flag;

puts("\n请输入查找房源范围：\n");

puts("\n朝向：输入0-全选；1-南；2-东；3-西；4-北\n");

scanf("%d",&flag.direction);

while ((c = getchar()) != EOF && c != '\n');//清空缓存区

puts("\n楼层数：输入最低与最高（例：输入5 10，输出所有五到十层的房源）\n");

scanf("%d %d",&flag.f\_level[0],&flag.f\_level[1]);

while ((c = getchar()) != EOF && c != '\n');//清空缓存区

puts("\n面积：输入最低与最高（例：输入50 200，输出所有五十到两百平方米的房源）\n");

scanf("%d %d",&flag.size[0],&flag.size[1]);

while ((c = getchar()) != EOF && c != '\n');//清空缓存区

puts("\n价格：输入最低与最高（例：输入500 20000，输出所有500到20000元每平方米的房源）\n");

scanf("%d %d",&flag.price[0],&flag.price[1]);

while ((c = getchar()) != EOF && c != '\n');//清空缓存区

puts("\n查找中\n") ;

for(finder=pestate;finder!=NULL;finder=finder->next){

token=(finder->f\_level>flag.f\_level[0]&&finder->f\_level<flag.f\_level[1]);

token=token&&(finder->size>flag.size[0]&&finder->size<flag.size[1]);

token=token&&(finder->price>flag.price[0]&&finder->price<flag.price[1]);

if(token==1){

if(flag.direction==0||flag.direction==finder->direction)

print\_estate(finder);

}

}

}

void statistics(void){

int i=0,max\_sold=0,cnt=0;

estate \*now=pestate;

while(now!=NULL){

if(now->sold>max\_sold){

max\_sold=now->sold;

cnt++;

}

now=now->next;

}

printf("客户意向房源共%d套\n",cnt);

for(i=max\_sold;i>0;i--){

for(now=pestate;now!=NULL;now=now->next){

if(now->sold==i)print\_estate(now);

}

}

}

void book(void){

sold\_estate \*now=psold;

char c;

if(now!=NULL)while(now->next!=NULL)now=now->next;

now=newnode\_sold(now);

puts("请输入面积");

if(scanf("%d",&now->size)!=1){

WARN;

}

while ((c = getchar()) != EOF && c != '\n');//清空缓存区

puts("请输入价格");

if(scanf("%d",&now->price)!=1){

WARN;

}

while ((c = getchar()) != EOF && c != '\n');//清空缓存区

puts("请输入成交时间(年月日用空格隔开，如：2020 12 31)");

if(scanf("%d %d %d",&now->time[0],&now->time[1],&now->time[2])!=3){

WARN;

}

while ((c = getchar()) != EOF && c != '\n');//清空缓存区

puts("请输入单号");

if(scanf("%d",&now->id)!=1){

WARN;

}

while ((c = getchar()) != EOF && c != '\n');//清空缓存区

puts("请输入客户的姓名");

if(scanf("%s",now->name)!=1){

WARN;

}

while ((c = getchar()) != EOF && c != '\n');//清空缓存区

puts("请输入客户的电话");

if(scanf("%s",now->p\_number)!=1){

WARN;

}

while ((c = getchar()) != EOF && c != '\n');//清空缓存区

printf("保存新合同%d成功",now->id);

while ((c = getchar()) != EOF && c != '\n');//清空缓存区

}

void read(){

int i=0,k=0;

FILE \*fp;

puts("\n正在读取数据\n");

//读取source

estate \*now=pestate;

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

fprintf(stderr,"Error:打开文件source.txt失败\n");

exit(-1);

}

if(now!=NULL)while(now->next!=NULL)now=now->next;

while(fscanf(fp,"%s",buffer)==1){

if(!strcmp(buffer,"estate")){

now=newnode\_estate(now);

}

else{

puts("fail on estate");

WARN;

}

puts("success in estate");

if(fscanf(fp,"%d",&now->id)!=1)

WARN;

printf("代号%3d ",now->id);

if(fscanf(fp,"%d",&now->sold)!=1)

WARN;

printf("卖出%2d套 ",now->sold);

if(fscanf(fp,"%d",&now->direction) !=1&&!(now->direction>0&&now->direction<5))

WARN;

printf("朝%s ",directions[now->direction-1]);

if(fscanf(fp,"%d",&now->f\_level) !=1)

WARN;

printf("%3d层 ",now->f\_level);

if(fscanf(fp,"%d",&now->size) !=1)

WARN;

printf("面积%6d ",now->size);

if(fscanf(fp,"%d",&now->price) !=1&&!(now->direction>0&&now->direction<5))

WARN;

printf("价格%6d ",now->price);

if(fscanf(fp,"%s",now->infor)!=1)

WARN;

printf("简介：%s\n\n",now->infor);

}

fclose(fp);

//读取guest

guest \*guest\_now=pguest;

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

fprintf(stderr,"Error:打开文件guests.txt失败\n");

exit(-1);

}

if(guest\_now!=NULL)while(guest\_now->next!=NULL)guest\_now=guest\_now->next;

puts("yes");

while(fscanf(fp,"%s",buffer)==1){

if(!strcmp(buffer,"guest")){

guest\_now=newnode\_guest(guest\_now);

}

else{

puts("fail on guest");

WARN;

}

if(fscanf(fp,"%s",guest\_now->p\_number)!=1)

WARN;

printf("电话%9s\n",guest\_now->p\_number);

if(fscanf(fp,"%s",guest\_now->name) !=1)

WARN;

printf("姓名%8s\n",guest\_now->name);

if(fscanf(fp,"%d",&guest\_now->call\_number)!=1)

WARN;

printf("电话次数%2d\n",guest\_now->call\_number);

if(fscanf(fp,"%d %d %d",&guest\_now->last\_call[0],&guest\_now->last\_call[1],&guest\_now->last\_call[2]) !=3)

WARN;

printf("上次通话时间%4d年%2d月%2d日\n",guest\_now->last\_call[0],guest\_now->last\_call[1],guest\_now->last\_call[2]);

if(fscanf(fp,"%d %d %d",&guest\_now->order[0],&guest\_now->order[1],&guest\_now->order[2])!=3)

WARN;

for(i=0;i<3;i++){

if(guest\_now->order[i]!=0)

printf("该买家意向房源%d:%d\n",i+1,guest\_now->order[i]);

}

putchar('\n');

}

//读取sold

sold\_estate \*sold\_now=psold;

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

fprintf(stderr,"Error:打开文件sold.txt失败\n");

exit(-1);

}

if(sold\_now!=NULL)while(sold\_now->next!=NULL)sold\_now=sold\_now->next;

while(fscanf(fp,"%s",buffer)==1){

if(!strcmp(buffer,"sold")){

sold\_now=newnode\_sold(sold\_now);

}

else{

puts("fail on sold");

WARN;

}

if(fscanf(fp,"%d",&sold\_now->size)!=1)

WARN;

printf("面积%2d\n",sold\_now->size);

if(fscanf(fp,"%d",&sold\_now->price)!=1)

WARN;

printf("价格%2d\n",sold\_now->price);

if(fscanf(fp,"%d %d %d",&sold\_now->time[0],&sold\_now->time[1],&sold\_now->time[2]) !=3)

WARN;

printf("成交时间%4d年%2d月%2d日\n",sold\_now->time[0],sold\_now->time[1],sold\_now->time[2]);

if(fscanf(fp,"%d",&sold\_now->id)!=1)

WARN;

printf("合同单号%2d\n",sold\_now->id);

if(fscanf(fp,"%s",sold\_now->name)!=1)

WARN;

printf("房主%s\n",sold\_now->name);

if(fscanf(fp,"%s",sold\_now->p\_number)!=1)

WARN;

printf("电话%9s\n",sold\_now->p\_number);

putchar('\n');

}

fclose(fp);

puts("\n读取成功\n") ;

}

void save(){

int i=0;

FILE \*fp;

//储存source

if((fp=fopen("source.txt","w"))==NULL){

fprintf(stderr,"Error:打开文件失败\n");

exit(-1);

}

estate \*now=pestate;

if(now!=NULL){

fprintf(fp,"estate %d %d %d %d ",now->id,now->sold,now->direction,now->f\_level);

fprintf(fp,"%d %d %s\n",now->size,now->price,now->infor);

while(now->next!=NULL){

now=now->next;

fprintf(fp,"estate %d %d %d %d ",now->id,now->sold,now->direction,now->f\_level);

fprintf(fp,"%d %d %s\n",now->size,now->price,now->infor);

}

}

fclose(fp);

//储存guest

if((fp=fopen("guests.txt","w"))==NULL){

fprintf(stderr,"Error:打开文件失败\n");

exit(-1);

}

guest \*guest\_now=pguest;

if(guest\_now!=NULL){

fprintf(fp,"guest %s %s %d ",guest\_now->p\_number,guest\_now->name,guest\_now->call\_number);

fprintf(fp,"%d %d %d ",guest\_now->last\_call[0],guest\_now->last\_call[1],guest\_now->last\_call[2]);

fprintf(fp,"%d %d %d\n",guest\_now->order[0],guest\_now->order[1],guest\_now->order[2]);

while(guest\_now->next!=NULL){

guest\_now=guest\_now->next;

fprintf(fp,"guest %s %s %d ",guest\_now->p\_number,guest\_now->name,guest\_now->call\_number);

fprintf(fp,"%d %d %d ",guest\_now->last\_call[0],guest\_now->last\_call[1],guest\_now->last\_call[2]);

fprintf(fp,"%d %d %d\n",guest\_now->order[0],guest\_now->order[1],guest\_now->order[2]);

}

}

fclose(fp);

//储存sold

if((fp=fopen("sold.txt","w"))==NULL){

fprintf(stderr,"Error:打开文件失败\n");

exit(-1);

}

sold\_estate \*sold\_now=psold;

if(sold\_now!=NULL){

fprintf(fp,"sold %d %d %d %d %d ",sold\_now->size,sold\_now->price,sold\_now->time[0],sold\_now->time[1],sold\_now->time[2]);

fprintf(fp,"%d %s %s\n",sold\_now->id,sold\_now->name,sold\_now->p\_number);

while(sold\_now->next!=NULL){

sold\_now=sold\_now->next;

fprintf(fp,"sold %d %d %d %d %d ",sold\_now->size,sold\_now->price,sold\_now->time[0],sold\_now->time[1],sold\_now->time[2]);

fprintf(fp,"%d %s %s\n",sold\_now->id,sold\_now->name,sold\_now->p\_number);

}

}

fclose(fp);

puts("\n保存成功\n");

}

void Auto\_save\_change(){

if(Auto\_save\_mode==true){

Auto\_save\_mode=false;

}

else Auto\_save\_mode=true;

}

## main.c

#include<stdio.h>

#include<stdbool.h>

#include"function.h"

extern bool Auto\_save\_mode;

char show\_menu\_main(void);

int main(){

puts("程序加载中");

int i=0;char ch;

read();

while(ch!='q'&&ch!=EOF){

switch (ch=show\_menu\_main()){

case '1':add();

break;

case '2':change();

break;

case '3':source();

break;

case '4':statistics();

break;

case '5':book();

break;

case '6':Auto\_save\_change();

break;

case '7':read();

break;

case '8':save();

break;

default:break;

}

}

puts("退出前是否保存？（保存请按一,不保存按任意键）\n");

if(getchar()=='1')save();

puts("程序结束，感谢您的使用\n");

return 0;

}

char show\_menu\_main(void){

if(Auto\_save\_mode==1)save();

puts("\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n\n");

puts("\*\*\*\*\*欢迎使用房地产管理程序\*\*\*\*\*\n");

puts("1)新客户登记 2)修改客户数据 3)房源数据 \n"); //add/change/source

printf("4)统计 5)预订登记 6)%s自动保存模式\n\n",Auto\_save\_mode?"关闭":"开启"); //save/read/Autp\_save\_change

puts("7)读取数据 8)保存数据 q)退出\n");

puts("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n\n");

char ch='0',c='0';

ch=tolower(getchar());

while ((c = getchar()) != EOF && c != '\n');//清空缓存区

//printf("%c %c",ch,c);

return ch;

}