#include <stdio.h>

#include <conio.h>

#include <windows.h>

#include <string.h>

#include <stdlib.h>

typedef struct{

int code;

char grad[51];

char ylica[51];

int nomer;

float plo6t;

int stai;

float naem;

}KVARTIRA;

typedef struct list{

KVARTIRA naem;

struct list \*next;

}list;

list\* addel(list\* prev)

{

list\*spisak;

int check,i;

int code;

char grad[51];

char ylica[51];

int nomer;

float plo6t;

int stai;

float naem;

spisak=(list \*)malloc(sizeof(list));

spisak->next=prev;

prev=spisak;

for(spisak=prev;;spisak=prev)

{

printf("Въведете уникален код: ");

fflush(stdin);

scanf("%d",&code);

if(code<=0)continue;

for(spisak=prev;;spisak=spisak->next)

{

if(code==spisak->naem.code)break;

if(spisak->next==NULL)break;

}

if(code==spisak->naem.code)

{

printf("\n\nВече има въведена квартира с такъв код!\n\n ");

continue;

}

else

{

break;

}

}

spisak=prev;

spisak->naem.code=code;

do{

check=0;

printf("Въведете град на квартирата: ");

fflush(stdin);

fgets(grad,51,stdin);

for(i=0;i<(strlen(grad)-1);i++)

{

if((!(grad[i]>='A'&&grad[i]<='Z')) && (!(grad[i]>='А'&&grad[i]<='Я')) )

{

check=1;

break;

}

}

if(check)

{

check=0;

continue;

}

else

{

break;

}

}while(1);

strcpy(spisak->naem.grad,grad);

do{

check=0;

printf("Въведете улица на квартирата: ");

fflush(stdin);

fgets(ylica,51,stdin);

for(i=0;i<(strlen(ylica)-1);i++)

{

if((!(ylica[i]>='A'&&ylica[i]<='Z')) && (!(ylica[i]>='А'&&ylica[i]<='Я')) )

{

check=1;

break;

}

}

if(check)

{

check=0;

continue;

}

else

{

break;

}

}while(1);

strcpy(spisak->naem.ylica,ylica);

do{

printf("Въведете номер на улицата: ");

fflush(stdin);

scanf("%d",&nomer);

if(nomer<0)

{

printf("Въведете положително число.\n");

continue;

}

else

{

break;

}

}while(1);

spisak->naem.nomer=nomer;

do{

printf("Въведете брой площ: ");

fflush(stdin);

scanf("%f",&plo6t);

if(plo6t<0)

{

printf("Въведете положително число.\n");

continue;

}

else

{

break;

}

}while(1);

spisak->naem.plo6t=plo6t;

do{

printf("Въведете брой стаи: ");

fflush(stdin);

scanf("%d",&stai);

if(stai<0)

{

printf("Въведете положително число.\n");

continue;

}

else

{

break;

}

}while(1);

spisak->naem.stai=stai;

do{

printf("Въведете месечен наем: ");

fflush(stdin);

scanf("%f",&naem);

if(naem<0)

{

printf("Въведете положително число.\n");

continue;

}

else

{

break;

}

}while(1);

spisak->naem.naem=naem;

return prev;

}

list\* editel(list\* prev)

{

list \*spisak=NULL, \*head2=NULL;

int i,ch,broi=0,check;

int code;

char grad[51];

char ylica[51];

int nomer;

float plo6t;

int stai;

float naem;

for(check=0;;)

{

do{

printf("За промяня нятиснете 2.\nЗа изтриване - 1 ");

fflush(stdin);

ch=getche();

if(ch=='1'|| ch=='2')break;

}while(1);

system("cls");

printf("Въведете уникален код: ");

fflush(stdin);

scanf("%d",&code);

if(code<=0)continue;

system("cls");

for(head2=NULL,spisak=prev;spisak!=NULL;head2=spisak,spisak=spisak->next)

{

if(spisak->naem.code==code)

{

if(ch=='2')

{

printf("\nЗа промяна на: \nНомер на улица - 3\nГрад - 4\nУлица - 5\nПлощ - 6\nСтаи - 7\nНаем - 8\n");

do{

fflush(stdin);

ch=getche();

if(ch=='3'||ch=='4'||ch=='5'||ch=='6'||ch=='7'||ch=='8')break;

}while(1);

system("cls");

switch(ch)

{

case '3':

do{

printf("Въведете номер: ");

fflush(stdin);

scanf("%d",&nomer);

if(nomer<0)

{

printf("Въведете положително число.\n");

continue;

}

else

{

break;

}

}while(1);

spisak->naem.nomer=nomer;

check=1;

break;

case '4':

do{

check=0;

printf("Въведете град: ");

fflush(stdin);

fgets(grad,51,stdin);

for(i=0;i<(strlen(grad)-1);i++)

{

if((!(grad[i]>='A'&&grad[i]<='Z')) && (!(grad[i]>='А'&&grad[i]<='Я')) )

{

check=1;

break;

}

}

if(check)

{

check=0;

continue;

}

else

{

break;

}

}while(1);

strcpy(spisak->naem.grad,grad);

check=1;

break;

case '5':

do{

check=0;

printf("Въведете улица: ");

fflush(stdin);

fgets(ylica,51,stdin);

for(i=0;i<(strlen(ylica)-1);i++)

{

if((!(ylica[i]>='A'&&ylica[i]<='Z')) && (!(ylica[i]>='А'&&ylica[i]<='Я')) )

{

check=1;

break;

}

}

if(check)

{

check=0;

continue;

}

else

{

break;

}

}while(1);

strcpy(spisak->naem.ylica,ylica);

check=1;

break;

case '6':

do{

printf("Въведете площ: ");

fflush(stdin);

scanf("%f",&plo6t);

if(plo6t<0)

{

printf("Въведете положително число.\n");

continue;

}

else

{

spisak->naem.plo6t=plo6t;

break;

}

}while(1);

check=1;

break;

case '7':

do{

printf("Въведете стаи: ");

fflush(stdin);

scanf("%d",&stai);

if(stai<0)

{

printf("Въведете положително число.\n");

continue;

}

else

{

break;

}

}while(1);

spisak->naem.stai=stai;

check=1;

break;

case '8':

do{

printf("Въведете наем: ");

fflush(stdin);

scanf("%f",&naem);

if(naem<0)

{

printf("Въведете положително число.\n");

continue;

}

else

{

break;

}

}while(1);

spisak->naem.naem=naem;

check=1;

break;

}

break;

}/////

else

{

if(head2!=NULL)

{

head2->next=spisak->next;

}

else

{

prev=spisak->next;

}

check=1;

printf("\nУспешно изтрита!\n\n");

free(spisak);

break;

}

}

}

if(!check)

{

printf("\nНе беше намерен КВАРТИРА с този номер.\n\n");

}

break;

}

return prev;

}

list\* lookfor(list\* prev)

{

KVARTIRA \*sort=NULL, swap;

list \*spisak=NULL, \*head2=NULL;

int i,check,br=0,b;

char grad[51];

for(spisak=prev;;spisak=spisak->next)

{

if(spisak!=NULL)

{

br++;

}

else

{

break;

}

}

if(sort!=NULL) free(sort);

sort=(KVARTIRA \*)malloc(br\*sizeof(KVARTIRA));

do{

check=0;

printf("Въведете име на град: ");

fflush(stdin);

fgets(grad,51,stdin);

b=strlen(grad)-1;

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

{

if((!(grad[i]>='A'&&grad[i]<='Z')) && (!(grad[i]>='А'&&grad[i]<='Я')) )

{

check=1;

break;

}

}

if(check)

{

check=0;

continue;

}

else

{

break;

}

}while(1);

for(i=0,check=0,spisak=prev;;spisak=spisak->next)

{

if(strcmp(spisak->naem.grad,grad)==0)

{

sort[i].code=spisak->naem.code;

strcpy(sort[i].grad, spisak->naem.grad);

sort[i].naem=spisak->naem.naem;

sort[i].nomer=spisak->naem.nomer;

sort[i].plo6t=spisak->naem.plo6t;

sort[i].stai=spisak->naem.stai;

strcpy(sort[i].ylica,spisak->naem.ylica);

i++;

check=1;

}

if(spisak->next==NULL)break;

}

if(!check)

{

printf("\n\nНе беще открита квартира в този град!");

return prev;

}

do{

check=1;

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

{

if(i<br-1)

{

if(sort[i].naem>sort[i+1].naem)

{

swap.code=sort[i].code;

swap.naem=sort[i].naem;

swap.nomer=sort[i].nomer;

swap.plo6t=sort[i].plo6t;

swap.stai=sort[i].stai;

strcpy(swap.grad,sort[i].grad);

strcpy(swap.ylica,sort[i].ylica);

sort[i].code=sort[i+1].code;

strcpy(sort[i].grad, sort[i+1].grad);

sort[i].naem=sort[i+1].naem;

sort[i].nomer=sort[i+1].nomer;

sort[i].plo6t=sort[i+1].plo6t;

sort[i].stai=sort[i+1].stai;

strcpy(sort[i].ylica,sort[i+1].ylica);

sort[i+1].code=swap.code;

strcpy(sort[i+1].grad, swap.grad);

sort[i+1].naem=swap.naem;

sort[i+1].nomer=swap.nomer;

sort[i+1].plo6t=swap.plo6t;

sort[i+1].stai=swap.stai;

strcpy(sort[i+1].ylica,swap.ylica);

check=0;

}

}

}

}while(!check);

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

{

printf("\nКод: %d",sort[i].code);

printf("\nград: %s улица %s номер %d",sort[i].grad,sort[i].ylica,sort[i].nomer);

printf("\nСтаи: %d \nПлощ %f",sort[i].stai,sort[i].plo6t);

printf("\nНаем: %f\n\n\n",sort[i].naem);

}

return prev;

}

int main(void)

{

list \*spisak,\*prev=NULL;

int ch,check=0;

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

for(;;)

{

printf ("1. Въвеждане на нова квартира.\n");

printf ("2. Променяне на данни за квартира.\n");

printf ("3. Изписване на квартира за даден град.\n");

printf ("4. ИЗХОД \n");

printf ("Изберете опция от следното меню: ");

fflush(stdin);

ch=getche();

switch(ch)

{

case '1':

system("cls");

prev=addel(prev);

check=1;

system("Pause");

system("cls");

break;

case '2':

system("cls");

if(check) prev=editel(prev);

else printf("Започнете от опция 1\n\n");

system("Pause");

system("cls");

break;

case '3':

system("cls");

if(check) prev=lookfor(prev);

else printf("Започнете от опция 1\n\n");

system("Pause");

system("cls");

break;

case '4':

printf("\n\n");

system("Pause");

return 0;

default:

system("cls");

break;

}

}

}