#include <stdio.h>

#include <conio.h>

#include <windows.h>

#include <string.h>

#include <stdlib.h>

typedef struct{

int code;

char kvartal[51];

char upravitel[51];

int etaji;

int apartamenti;

}FIGURE;

typedef struct list{

FIGURE figura;

struct list \*next;

}list;

list \* addelement(list \* prev)

{

list \*spisak;

int check;

int code;

int etaji;

int apartamenti;

char upravitel[51];

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

spisak->next=prev;

prev=spisak;

for(spisak=prev;;spisak=prev)

{

printf("Въведете уникален код по-голям от 0: ");

fflush(stdin);

scanf("%d",&code);

if(code<=0)continue;

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

{

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

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

}

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

{

printf("\n\nВече има въведен блок с такъв код!\n\n ");

continue;

}

else

{

break;

}

}

spisak=prev;

spisak->figura.code=code;

printf("Въведете име на квартал: ");

fflush(stdin);

scanf("%s",spisak->figura.kvartal);

printf("Въведете име на домоуправител: ");

fflush(stdin);

scanf("%s",upravitel);

strcpy(spisak->figura.upravitel,upravitel);

do{

check=0;

printf("\nВъведете брой на етажи: ");

fflush(stdin);

scanf("%d",&etaji);

if(etaji<=0)

{

printf("Въведете цяло положително число! ");

check=1;

}

}while(check);

spisak->figura.etaji=etaji;

do{

check=0;

printf("\nВъведете брой на апартаменти: ");

fflush(stdin);

scanf("%d",&apartamenti);

if(apartamenti<0)

{

printf("Въведете цяло положително число или 0! ");

check=1;

}

}while(check);

spisak->figura.apartamenti=apartamenti;

return prev;

}

list\* change(list\* prev)

{

list \*spisak;

int code,ch,check;

if(prev==NULL) return prev;

for(;;)

{

printf("Въведете код: ");

fflush(stdin);

scanf("%d",&code);

if(code<=0)

{

continue;

}

break;

}

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

{

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

{

printf("За промяна на квартал - 1\nдомоуправетул - 2\nетажи - 3\nапартаменти -4\n");

ch=getche();

switch(ch)

{

case '1':

printf("Въведете име на квартал: ");

fflush(stdin);

scanf("%s",spisak->figura.kvartal);

break;

case '2':

printf("Въведете име на домоуправител: ");

fflush(stdin);

scanf("%s",spisak->figura.upravitel);

break;

case '3':

do{

check=0;

printf("\nВъведете брой на етажи: ");

fflush(stdin);

scanf("%d",&spisak->figura.etaji);

if(spisak->figura.etaji<=0)

{

printf("Въведете цяло положително число! ");

check=1;

}

}while(check);

break;

case '4':

do{

check=0;

printf("\nВъведете брой на апартаменти: ");

fflush(stdin);

scanf("%d",&spisak->figura.apartamenti);

if(spisak->figura.apartamenti<0)

{

printf("Въведете цяло положително число или 0! ");

check=1;

}

}while(check);

}

break;

}

}

return prev;

}

list\* sortt(list\* prev)

{

list\* spisak;

int broi=0,i,check;

FIGURE \*sortiran, swap;

char kvartal[51];

if(prev==NULL)return prev;

printf("Въведете име на квартал: ");

fflush(stdin);

scanf("%s",kvartal);

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

{

if(strcmp(spisak->figura.kvartal,kvartal)==0)

{

broi++;

}

}

if(broi==0)

{

printf("\nНяма въведени апартаменти в този КВАРТАЛ.\n");

return prev;

}

sortiran=(FIGURE\*)malloc(broi\*sizeof(FIGURE));

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

{

if(strcmp(spisak->figura.kvartal,kvartal)==0)

{

sortiran[i].apartamenti=spisak->figura.apartamenti;

sortiran[i].code=spisak->figura.code;

sortiran[i].etaji=spisak->figura.etaji;

strcpy(sortiran[i].kvartal,spisak->figura.kvartal);

strcpy(sortiran[i].upravitel,spisak->figura.upravitel);

i++;

}

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

}

do{

check=1;

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

{

if(i<broi-1)

{

if(sortiran[i].code>sortiran[i+1].code)

{

swap.apartamenti=sortiran[i].apartamenti;

swap.code=sortiran[i].code;

swap.etaji=sortiran[i].etaji;

strcpy(swap.kvartal,sortiran[i].kvartal);

strcpy(swap.upravitel,sortiran[i].upravitel);

sortiran[i].apartamenti=sortiran[i+1].apartamenti;

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

sortiran[i].etaji=sortiran[i+1].etaji;

strcpy(sortiran[i].kvartal,sortiran[i+1].kvartal);

strcpy(sortiran[i].upravitel,sortiran[i+1].upravitel);

sortiran[i+1].apartamenti=swap.apartamenti;

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

sortiran[i+1].etaji=swap.etaji;

strcpy(sortiran[i+1].kvartal,swap.kvartal);

strcpy(sortiran[i+1].upravitel,swap.upravitel);

check=0;

}

}

}

}while(!check);

system("cls");

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

{

printf("Код %d\nЕтажи %d\nАпартаменти %d\nКвартал %s\nУправител %s\n\n\n",sortiran[i].code,sortiran[i].etaji,sortiran[i].apartamenti,sortiran[i].kvartal,sortiran[i].upravitel);

}

return prev;

}

list\* info(list\* prev)

{

list\* spisak;

int etaj,check;

if(prev==NULL)return prev;

do{

check=0;

printf("\nВъведете брой на етажи: ");

fflush(stdin);

scanf("%d",&etaj);

if(etaj<=0)

{

printf("Въведете цяло положително число! ");

check=1;

}

}while(check);

for(check=1,spisak=prev;spisak!=NULL;spisak=spisak->next)

{

if(spisak->figura.etaji>etaj)

{

printf("Код %d\nЕтажи %d\nАпартаменти %d\nКвартал %s\nУправител %s\n\n\n",spisak->figura.code,spisak->figura.etaji,spisak->figura.apartamenti,spisak->figura.kvartal,spisak->figura.upravitel);

check=0;

}

}

if(check)

{

system("cls");

printf("Не бяха открити блокове с следните изисквания!");

}

return prev;

}

int main(void)

{

list \*prev=NULL;

int ch;

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

for(;;)

{

printf ("1. Добавяне на нов блок.\n");

printf ("2. Промяна на данните за блок.\n");

printf ("3. Извеждане на апартаментите от даден квартал подредени по по номера.\n");

printf ("4. Извеждане на информация за апартаменти с етажи над избвания.\n");

printf ("5. ИЗХОД \n");

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

fflush(stdin);

ch=getche();

switch(ch)

{

case '1':

system("cls");

prev=addelement(prev);

system("Pause");

system("cls");

break;

case '2':

system("cls");

prev=change(prev);

system("Pause");

system("cls");

break;

case '3':

system("cls");

prev=sortt(prev);

system("Pause");

system("cls");

break;

case '4':

system("cls");

prev=info(prev);

system("Pause");

system("cls");

break;

case '5':

printf("\n\n");

system("Pause");

return 0;

default:

system("cls");

break;

}

}

}