#include <stdio.h>

#include <conio.h>

#include <windows.h>

#include <string.h>

#include <stdlib.h>

typedef struct{

char EGN[11];

char name[51];

int vuzrast;

char lekar[51];

char diagnoza[51];

}MAN;

typedef struct list{

MAN chovek;

struct list \*next;

}list;

list\* addel(list\* prev)

{

list\*spisak;

int check,i,vuzrast;

char lekar[51];

char diagnoza[51];

char EGN[11];

char name[51];

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

spisak->next=prev;

prev=spisak;

for(spisak=prev;;spisak=prev)

{

printf("Въведете ЕГН: ");

fflush(stdin);

fgets(EGN,11,stdin);

if(strlen(EGN)!=10)

{

printf("ЕГН-то е от 10 цифри!\n\n");

continue;

}

if( EGN[0]>'9' || EGN[0]<'0' || EGN[1]>'9' || EGN[1]<'0'

|| EGN[2]>'9' || EGN[2]<'0' ||EGN[3]>'9' || EGN[3]<'0'

|| EGN[4]>'9' || EGN[4]<'0' ||EGN[5]>'9' || EGN[5]<'0'

||EGN[6]>'9' || EGN[6]<'0'||EGN[7]>'9' || EGN[7]<'0'

||EGN[8]>'9' || EGN[8]<'0'||EGN[9]>'9' || EGN[9]<'0')

{

continue;

}

for(;;spisak=spisak->next)

{

if(strcmp(spisak->chovek.EGN,EGN)==0)

{

break;

}

if(spisak->next==NULL)

{

break;

}

}

if(strcmp(spisak->chovek.EGN,EGN)==0)

{

printf("\n\nВече има пациент с това егн!\n\n ");

continue;

}

else

{

break;

}

}

spisak=prev;

strcpy(spisak->chovek.EGN,EGN);

do{

check=0;

printf("Въведете име на пациент: ");

fflush(stdin);

fgets(name,51,stdin);

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

{

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

{

check=1;

break;

}

}

if(check)

{

check=0;

continue;

}

else

{

break;

}

}while(1);

strcpy(spisak->chovek.name,name);

do{

printf("Въведете възраст на пациента: ");

fflush(stdin);

scanf("%d",&vuzrast);

if(vuzrast>100 || vuzrast<0)

{

printf("Въведете година между 0 и 100.\n");

continue;

}

else

{

break;

}

}while(1);

spisak->chovek.vuzrast=vuzrast;

do{

check=0;

printf("Въведете име на лекар: ");

fflush(stdin);

fgets(lekar,51,stdin);

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

{

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

{

check=1;

break;

}

}

if(check)

{

check=0;

continue;

}

else

{

break;

}

}while(1);

strcpy(spisak->chovek.lekar,lekar);

do{

check=0;

printf("Въведете диагноза: ");

fflush(stdin);

fgets(diagnoza,51,stdin);

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

{

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

{

check=1;

break;

}

}

if(check)

{

check=0;

continue;

}

else

{

break;

}

}while(1);

strcpy(spisak->chovek.diagnoza,diagnoza);

return prev;

}

list\* editel(list\* prev)

{

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

int i,ch,broi=0,check;

char EGN[11];

char name[51];

int vuzrast;

char lekar[51];

char diagnoza[51];

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");

do{

check=0;

printf("Въведете ЕГН: ");

fflush(stdin);

fgets(EGN,11,stdin);

if(strlen(EGN)!=10)

{

printf("ЕГН-то е от 10 цифри!\n\n");

continue;

}

if( EGN[0]>'9' || EGN[0]<'0' || EGN[1]>'9' || EGN[1]<'0'

|| EGN[2]>'9' || EGN[2]<'0' ||EGN[3]>'9' || EGN[3]<'0'

|| EGN[4]>'9' || EGN[4]<'0' ||EGN[5]>'9' || EGN[5]<'0'

||EGN[6]>'9' || EGN[6]<'0'||EGN[7]>'9' || EGN[7]<'0'

||EGN[8]>'9' || EGN[8]<'0'||EGN[9]>'9' || EGN[9]<'0')

{

printf("Въвели сте невалиден номер!\n\n");

continue;

}

}while(check);

system("cls");

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

{

if(strcmp(spisak->chovek.EGN,EGN)==0)

{

switch(ch)

{

case '1':

if(head2!=NULL)

{

head2->next=spisak->next;

}

else

{

prev=spisak->next;

}

broi--;

check=1;

printf("\nУспешно изтрит!\n\n");

free(spisak);

break;

case '2':

printf("\nЗа промяна на: \nИме - 2\nГодини - 3\nЛекар - 4\nДиагноза - 5\n");

do{

fflush(stdin);

ch=getche();

if(ch=='2'||ch=='3'||ch=='4'||ch=='5')break;

}while(1);

system("cls");

switch(ch)

{

case '2':

do{

check=0;

printf("Въведете име на пациент: ");

fflush(stdin);

fgets(name,51,stdin);

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

{

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

{

check=1;

break;

}

}

if(check)

{

check=0;

continue;

}

else

{

break;

}

}while(1);

strcpy(spisak->chovek.name,name);

check=1;

break;

case '3':

do{

printf("Въведете възраст на пациента: ");

fflush(stdin);

scanf("%d",&vuzrast);

if(vuzrast>100 || vuzrast<0)

{

printf("Въведете година между 0 и 100.\n");

continue;

}

else

{

break;

}

}while(1);

spisak->chovek.vuzrast=vuzrast;

check=1;

break;

case '4':

do{

check=0;

printf("Въведете име на лекар: ");

fflush(stdin);

fgets(lekar,51,stdin);

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

{

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

{

check=1;

break;

}

}

if(check)

{

check=0;

continue;

}

else

{

break;

}

}while(1);

strcpy(spisak->chovek.lekar,lekar);

check=1;

break;

case '5':

do{

check=0;

printf("Въведете диагноза: ");

fflush(stdin);

fgets(diagnoza,51,stdin);

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

{

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

{

check=1;

break;

}

}

if(check)

{

check=0;

continue;

}

else

{

break;

}

}while(1);

strcpy(spisak->chovek.diagnoza,diagnoza);

check=1;

break;

}

break;

}

break;

}

}

if(!check)

{

printf("\nНе беше намерен пациент с такова ЕГН.\n\n");

}

break;

}

return prev;

}

list\* lookfor(list\* prev)

{

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

int i,check;

char lekar[51];

do{

check=0;

printf("Въведете име на лекар: ");

fflush(stdin);

fgets(lekar,51,stdin);

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

{

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

{

check=1;

break;

}

}

if(check)

{

check=0;

continue;

}

else

{

break;

}

}while(1);

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

{

if(strcmp(spisak->chovek.lekar,lekar)==0)

{

printf("ЕГН: %s",spisak->chovek.EGN);

printf("\nИме: %s",spisak->chovek.name);

printf("Години: %d",spisak->chovek.vuzrast);

printf("\nДиагноза: %s",spisak->chovek.diagnoza);

printf("Лекар: %s\n\n\n",spisak->chovek.lekar);

check=1;

}

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

}

if(!check) printf("\n\nНе беще открит пациент със зададения лекар!");

return prev;

}

list\* minmax(list\* prev)

{

int min,max,check;

list\* spisak;

printf("Въведете долна граница ");

do{

scanf("%d",&min);

if(min>100 || min<0)

{

printf("Въведете година между 0 и 100.\n");

continue;

}

else

{

break;

}

}while(1);

printf("Въведете горна граница ");

do{

scanf("%d",&max);

if(min>100 || min<0)

{

printf("Въведете година между 0 и 100.\n");

continue;

}

else

{

break;

}

}while(1);

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

{

if(min<=spisak->chovek.vuzrast&&max>=spisak->chovek.vuzrast)

{

printf("ЕГН: %s",spisak->chovek.EGN);

printf("\nИме: %s",spisak->chovek.name);

printf("Години: %d",spisak->chovek.vuzrast);

printf("\nДиагноза: %s",spisak->chovek.diagnoza);

printf("Лекар: %s\n\n\n",spisak->chovek.lekar);

check=1;

}

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

}

if(!check) printf("\n\nНе беще открит пациент в зададения интервал");

return prev;

}

int main(void)

{

list \*prev=NULL;

int ch,check=0;

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

for(;;)

{

printf ("1. Въвеждане на нов пациент.\n");

printf ("2. Изтриване на пациент.\n");

printf ("3. Изписване на информацията за пациенти с избран лекар на екрана.\n");

printf ("4. Изписване на информацията за пациенти с възраст в избран интервал.\n");

printf ("5. 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':

system("cls");

if(check) prev=minmax(prev);

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

system("Pause");

system("cls");

break;

case '5':

printf("\n\n");

system("Pause");

return 0;

default:

system("cls");

break;

}

}

}