#include <stdio.h>

#include <conio.h>

#include <windows.h>

#include <string.h>

#include <stdlib.h>

typedef struct{

unsigned long long code;

char name[40];

float cena;

int broi;

char data[11];

int srok;

int nadvi6avane;

}FIGURE;

typedef struct list{

FIGURE figura;

struct list \*next;

}list;

list \* addelement(list \* prev)

{

list \*spisak;

int check,i,a;

unsigned long long code;

float cena;

int broi;

int day2,mount2,year2;

char data[11];

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

spisak->next=prev;

prev=spisak;

for(;;)

{

do{

check=0;

printf("Въведете дата ДД.ПП.ГГГГ: ");

fflush(stdin);

scanf("%s",data);

if(strlen(data)!=10)

{

check=1;

continue;

}

for(i=0;i<10;i++)

{

if(i!=2&&i!=5)

{

if(data[i]<'0'&&data[i]>'9')

{

check=1;

break;

}

}

else

{

if(data[i]!='.')

{

check=1;

break;

}

}

}

}while(check);

day2=(data[0]-48)\*10+(data[1]-48);

mount2=(data[3]-48)\*10+(data[4]-48);

year2=(data[6]-48)\*1000+(data[7]-48)\*100+(data[8]-48)\*10+(data[9]-48);

check=1;

do{

if(year2<1)

{

printf("Въведете година от новата ера.\n");

check=0;

}

break;

}while(1);

if(check==0)continue;

do{

if(mount2<1 || mount2>12)

{

printf("Невалидна дата\n");

check=0;

}

break;

}while(1);

do{

if(mount2==1||mount2==3||mount2==5||mount2==7||mount2==8||mount2==10||mount2==12)

{

if(day2>31||day2<1)

{

printf("Невалидна дата\n");

check=0;

}

break;

}

if(mount2==4||mount2==6||mount2==9||mount2==11)

{

if(day2>30||day2<1)

{

printf("Невалидна дата\n");

check=0;

}

break;

}

if(mount2==2)

{

if(year2%4==0)

{

if(day2>29||day2<1)

{

printf("Невалидна дата\n");

check=0;

}

break;

}

else

{

if(day2>28||day2<1)

{

printf("Невалидна дата\n");

check=0;

}

break;

}

}

break;

}while(1);

if(check==0)continue;

else

{

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

}

for(spisak=prev;;spisak=prev)

{

printf("Въведете уникален код(12 цифрен): ");

fflush(stdin);

scanf("%llu",&code);

if(code<100000000000||code>999999999999)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 телефон с такъв код!\n\n ");

continue;

}

else

{

break;

}

}

spisak=prev;

spisak->figura.code=code;

printf("Въведете модел на телефона: ");

fflush(stdin);

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

do{

check=0;

printf("\nВъведете цена на телефона ");

fflush(stdin);

scanf("%f",&cena);

if(cena<=0)

{

printf("Въведете цяло положително число! ");

check=1;

}

}while(check);

spisak->figura.cena=cena;

do{

check=0;

printf("Въведете бройки на телефона ");

scanf("%d",&broi);

if(broi<=0)

{

printf("Въведете цяло положително число! ");

check=1;

}

}while(check);

spisak->figura.broi=broi;

do{

printf("Vuvedete broi dni godnost: ");

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

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

{

continue;

}

else

{

break;

}

}while(1);

break;

}

return prev;

}

list\* del(list \*prev)

{

int check=0,a;

list \*head2, \*spisak;

unsigned long long code;

if(prev==NULL) return prev;

do{

printf("Въведете уникален код(12 цифрен): ");

fflush(stdin);

scanf("%llu",&code);

if(code<100000000000||code>999999999999)continue;

}while(0);

for(check=0,head2=NULL,spisak=prev;spisak!=NULL;head2=spisak,spisak=spisak->next)

{

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

{

check=1;

printf("Коригирайте бройката с положително или отрицателно число: ");

scanf("%d",&a);

if((spisak->figura.broi+a)<0) break;

if(spisak->figura.broi+a==0)

{

if(head2!=NULL)

{

head2->next=spisak->next;

}

else

{

prev=spisak->next;

}

free(spisak);

break;

}

else

{

spisak->figura.broi=spisak->figura.broi+a;

break;

}

}

}

if(!check)

{

printf("\nНе беше намерен телефон с такъв номер.\n\n");

}

else

{

printf("успешно изпълнена операция!");

}

return prev;

}

list\* podata(list\* prev)

{

list \*spisak=NULL;

int cn,br=0,flag=0;

int m,n;

int day2, mount2, year2,yearoriginal;

int day, mount, year;

if(prev==NULL)

{

printf("Празен списък!!");

return prev;

}

printf("Въведете днешна дата!\n\n");

do{

printf("Въведете година: ");

fflush(stdin);

scanf("%d",&year2);

if(year2<1)

{

printf("Въведете година от новата ера.\n");

continue;

}

else

{

yearoriginal=year2;

break;

}

}while(1);

do{

printf("Въведете месец: ");

fflush(stdin);

scanf("%d",&mount2);

if(mount2<1 || mount2>12)

{

printf("Невалидна дата\n");

continue;

}

else

{

break;

}

}while(1);

do{

printf("Въведете ден: ");

fflush(stdin);

scanf("%d",&day2);

if(mount2==1||mount2==3||mount2==5||mount2==7||mount2==8||mount2==10||mount2==12)

{

if(day2>31||day2<1)

{

printf("Невалидна дата\n");

continue;

}

else

{

break;

}

}

if(mount2==4||mount2==6||mount2==9||mount2==11)

{

if(day2>30||day2<1)

{

printf("Невалидна дата\n");

continue;

}

else

{

break;

}

}

if(mount2==2)

{

if(year2%4==0)

{

if(day2>29||day2<1)

{

printf("Невалидна дата\n");

continue;

}

else

{

break;

}

}

else

{

if(day2>28||day2<1)

{

printf("Невалидна дата\n");

continue;

}

else

{

break;

}

}

}

}while(1);

system("Pause");

for(m=0,spisak=prev;spisak!=NULL;spisak=spisak->next)

{

day=(spisak->figura.data[0]-48)\*10+(spisak->figura.data[1]-48);

mount=(spisak->figura.data[3]-48)\*10+(spisak->figura.data[4]-48);

year=(spisak->figura.data[6]-48)\*1000+(spisak->figura.data[7]-48)\*100+(spisak->figura.data[8]-48)\*10+(spisak->figura.data[9]-48);

yearoriginal=year2;

cn=spisak->figura.srok;

if(year2<year)

{

printf("stoka s data za dostavka sled dne6nata !!Vuvedete validna data");

return prev;

}

flag++;

if(year2==year)

{

if(mount2>=mount)

{

if(mount2==mount)

{

if(day2==day)

{

m=0;

}

if(day2>day)

{

m=day2-day;

}

if(day2<day)

{

printf("Ima stoka s data za dostavka sled dne6nata !!Vuvedete validna data");

return prev;

}

}

else

{

n=mount2;

m=0;

m=day2;

n--;

for(;;n--)

{

if(n==1||n==3||n==5||n==7||n==8||n==10||n==12)

{

m=m+31;

}

if(n==4||n==6||n==9||n==11)

{

m=m+30;

}

if(n==2)

{

if(year2%4==0)

{

m=m+29;

}

else

{

m=m+28;

}

}

if(n==mount)break;

}

m=m-day;

}

}

else

{

printf("Ima stoka s data za dostavka sled dne6nata !!Vuvedete validna data");

return prev;

}

}

else

{

m=0;

if(mount2==mount)

{

do{

if(year2%4==0)

{

m=m+366;

}

else

{

m=m+365;

}

year2--;

}while(year2!=year);

if(day2==day)

{

m=m;

}

if(day2>day)

{

m=m+day2-day;

}

if(day2<day)

{

m=m-(day-day2);

}

}

if(mount2>mount)

{

n=mount2;

m=0;

do{

if(year2%4==0)

{

m=m+366;

}

else

{

m=m+365;

}

year2--;

}while(year2!=year);

for(;;n--)

{

if(n==mount)

{

if(n==1||n==3||n==5||n==7||n==8||n==10||n==12)

{

m=m+31-day;

}

if(n==4||n==6||n==9||n==11)

{

m=m+30-day;

}

if(n==2)

{

if(year2%4==0)

{

m=m+29-day;

}

else

{

m=m+28-day;

}

}

}

if(n>mount)

{

m=m+day2;

do{

n--;

if(n==1||n==3||n==5||n==7||n==8||n==10||n==12)

{

m=m+31;

}

if(n==4||n==6||n==9||n==11)

{

m=m+30;

}

if(n==2)

{

if(year2%4==0)

{

m=m+29;

}

else

{

m=m+28;

}

}

if(n==mount)

{

m=m-day;

break;

}

}while(1);

}

break;

}

}

if(mount2<mount)

{

n=mount2;

m=0;

do{

if(year2%4==0)

{

m=m+366;

}

else

{

m=m+365;

}

year2--;

}while(year2!=year);

if(n==1||n==3||n==5||n==7||n==8||n==10||n==12)

{

m=m-(31-day2);

}

if(n==4||n==6||n==9||n==11)

{

m=m-(30-day2);

}

if(n==2)

{

if(year2%4==0)

{

m=m-(29+day2);

}

else

{

m=m-(28+day2);

}

}

n++;

while(1)

{

if(n==mount)

{

m=m-day;

break;

}

if(n==1||n==3||n==5||n==7||n==8||n==10||n==12)

{

m=m-(31);

}

if(n==4||n==6||n==9||n==11)

{

m=m-(30);

}

if(n==2)

{

if(year2%4==0)

{

m=m-(29);

}

else

{

m=m-(28);

}

}

n++;

}

}

}

spisak->figura.nadvi6avane=m-cn;

year2=yearoriginal;

}

system("cls");

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

{

if(spisak->figura.nadvi6avane>0)

{

printf("Код: %llu\n Наименование: %s\nЦена: %f\nБрой: %d\nСрок на годност %d\nПроизведено на: %s\n\n",spisak->figura.code,spisak->figura.name,spisak->figura.cena,spisak->figura.broi,spisak->figura.srok,spisak->figura.data);

}

}

return prev;

}

list\* ponomer(list\* prev)

{

list\*spisak;

FIGURE \*sort,swap;

int check,broi=0,i;

if(prev==NULL)return prev;

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

{

broi++;

}

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

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

{

sort[i]=spisak->figura;

}

do{

check=1;

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

{

if(i<broi-1)

{

if(sort[i].code<sort[i+1].code)

{

swap=sort[i];

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

sort[i+1]=swap;

check=0;

}

}

}

}while(!check);

system("cls");

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

printf("Код: %llu\n Наименование: %s\nЦена: %f\nБрой: %d\nСрок на годност %d\nПроизведено на: %s\n\n",sort[i].code,sort[i].name,sort[i].cena,sort[i].broi,sort[i].srok,sort[i].data);

return prev;

}

list\* pocena(list\* prev)

{

list\*spisak;

FIGURE \*sort,swap;

int check,broi=0,i;

if(prev==NULL)return prev;

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

{

broi++;

}

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

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

{

sort[i]=spisak->figura;

}

do{

check=1;

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

{

if(i<broi-1)

{

if(sort[i].cena<sort[i+1].cena)

{

swap=sort[i];

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

sort[i+1]=swap;

check=0;

}

}

}

}while(!check);

system("cls");

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

printf("Код: %llu\n Наименование: %s\nЦена: %f\nБрой: %d\nСрок на годност %d\nПроизведено на: %s\n\n",sort[i].code,sort[i].name,sort[i].cena,sort[i].broi,sort[i].srok,sort[i].data);

return prev;

}

list\* pobroi(list\* prev)

{

list\*spisak;

FIGURE \*sort,swap;

int check,broi=0,i;

if(prev==NULL)return prev;

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

{

broi++;

}

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

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

{

sort[i]=spisak->figura;

}

do{

check=1;

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

{

if(i<broi-1)

{

if(sort[i].broi<sort[i+1].broi)

{

swap=sort[i];

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

sort[i+1]=swap;

check=0;

}

}

}

}while(!check);

system("cls");

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

printf("Код: %llu\n Наименование: %s\nЦена: %f\nБрой: %d\nСрок на годност %d\nПроизведено на: %s\n\n",sort[i].code,sort[i].name,sort[i].cena,sort[i].broi,sort[i].srok,sort[i].data);

return prev;

}

list\* poime(list\* prev)

{

list\*spisak;

FIGURE \*sort,swap;

int check,broi=0,i;

if(prev==NULL)return prev;

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

{

broi++;

}

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

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

{

sort[i]=spisak->figura;

}

do{

check=1;

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

{

if(i<broi-1)

{

if(strcmp(sort[i].name,sort[i+1].name)>0)

{

swap=sort[i];

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

sort[i+1]=swap;

check=0;

}

}

}

}while(!check);

system("cls");

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

printf("Код: %llu\n Наименование: %s\nЦена: %f\nБрой: %d\nСрок на годност %d\nПроизведено на: %s\n\n",sort[i].code,sort[i].name,sort[i].cena,sort[i].broi,sort[i].srok,sort[i].data);

return prev;

}

int main(void)

{

list \*prev=NULL;

int ch;

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

for(;;)

{

printf ("1. Добавяне на нова стока.\n");

printf ("2. Променяне на наличностите от зададена стока .\n");

printf ("3. Сортировка.\n");

printf ("4. Извеждане информация за всички стока с изтекъл срок.\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=del(prev);

system("Pause");

system("cls");

break;

case '3':

system("cls");

if(prev==NULL)

{

system("cls");

continue;

}

do{

printf("По номер - 1,по наименование - 2,по цена -3, по количество -4 ");

fflush(stdin);

ch=getche();

}while(ch!='1'&&ch!='3'&&ch!='2'&&ch!='4');

switch(ch)

{

case '1':

prev=ponomer(prev);

break;

case'2':

prev=poime(prev);

break;

case'3':

prev=pocena(prev);

break;

case'4':

prev=pobroi(prev);

break;

}

system("Pause");

system("cls");

break;

case '4':

system("cls");

prev=podata(prev);

system("Pause");

system("cls");

break;

case '5':

printf("\n\n");

system("Pause");

return 0;

default:

system("cls");

break;

}

}

}