#include <stdio.h>

#include <conio.h>

#include <windows.h>

#include <string.h>

#include <stdlib.h>

typedef struct{

char number[7];

char name[31];

int year;

int mount;

int day;

char avtorime[21];

char avtorprezime[21];

char avtorfamiliq[21];

}DOC;

typedef struct list{

DOC kniga;

struct list \*next;

}list;

list\* addel(list\* prev)

{

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

int i,ch,cn,broi=0,check;

char number[7];

char name[31];

int day, mount, year;

char avtorime[21];

char avtorprezime[21];

char avtorfamiliq[21];

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

spisak->next=prev;

prev=spisak;

for(spisak=prev;;spisak=prev)

{

printf("Въведете номер: ");

fflush(stdin);

fgets(number,7,stdin);

if( number[0]>'Я' || number[0]<'А' || number[1]>'Я' || number[1]<'А'

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

|| number[4]>'9' || number[4]<'0' ||number[5]>'Я' || number[5]<'А')

{

continue;

}

for(;;spisak=spisak->next)

{

if(strcmp(spisak->kniga.number,number)==0)

{

break;

}

if(spisak->next==NULL)

{

break;

}

}

if(strcmp(spisak->kniga.number,number)==0)

{

printf("\n\nВече има документ с такъв номер!\n\n ");

continue;

}

else

{

break;

}

}

spisak=prev;

strcpy(spisak->kniga.number,number);

do{

check=0;

printf("Въведете име на документ: ");

fflush(stdin);

fgets(name,31,stdin);

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

{

if( (!(name[i]>='a'&&name[i]<='z')) && (!(name[i]>='A'&&name[i]<='Z')) &&

(!(name[i]>='0'&&name[i]<='9')) && (!(name[i]>='а'&&name[i]<='я')) &&

(!(name[i]>='А'&&name[i]<='Я')) )

{

check=1;

break;

}

}

if(check)

{

check=0;

continue;

}

if( (name[0]>='A' && name[0]<='Z') || (name[0]>='А' && name[0]<='Я') )

{

break;

}

}while(1);

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

do{

printf("Въведете година на издаване: ");

fflush(stdin);

scanf("%d",&year);

if(year>2013 || year<1)

{

printf("Въведете година между 1 и 2013.\n");

continue;

}

else

{

break;

}

}while(1);

do{

printf("Въведете месец на издаване: ");

fflush(stdin);

scanf("%d",&mount);

if(mount<1 || mount>12)

{

printf("Невалидна дата\n");

continue;

}

else

{

break;

}

}while(1);

do{

printf("Въведете ден на издаване: ");

fflush(stdin);

scanf("%d",&day);

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

{

if(day>31||day<1)

{

printf("Невалидна дата\n");

continue;

}

else

{

break;

}

}

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

{

if(day>30||day<1)

{

printf("Невалидна дата\n");

continue;

}

else

{

break;

}

}

if(mount==2)

{

if(year%4==0)

{

if(day>29||day<1)

{

printf("Невалидна дата\n");

continue;

}

else

{

break;

}

}

else

{

if(day>28||day<1)

{

printf("Невалидна дата\n");

continue;

}

else

{

break;

}

}

}

}while(1);

spisak->kniga.year=year;

spisak->kniga.day=day;

spisak->kniga.mount=mount;

printf("Въведете име на автора: ");

fflush(stdin);

fgets(avtorime,21,stdin);

strcpy(spisak->kniga.avtorime,avtorime);

printf("Въведете презиме на автора: ");

fflush(stdin);

fgets(avtorprezime,21,stdin);

strcpy(spisak->kniga.avtorprezime,avtorprezime);

printf("Въведете фамилия на автора: ");

fflush(stdin);

fgets(avtorfamiliq,21,stdin);

strcpy(spisak->kniga.avtorfamiliq,avtorfamiliq);

return prev;

}

list\* delel(list\* prev)

{

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

int i,ch,cn,broi=0,check;

char number[7];

for(check=0;;)

{

printf("Въведете номер: ");

fflush(stdin);

fgets(number,7,stdin);

if( number[0]>'Я' || number[0]<'А' || number[1]>'Я' || number[1]<'А'

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

|| number[4]>'9' || number[4]<'0' ||number[5]>'Я' || number[5]<'А' )

{

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

continue;

}

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

{

if(strcmp(spisak->kniga.number,number)==0)

{

if(head2!=NULL)

{

head2->next=spisak->next;

}

else

{

prev=spisak->next;

}

check=1;

broi--;

printf("\nУспешно изтрит!\n\n");

free(spisak);

break;

}

if(spisak->next==NULL)

{

break;

}

}

if(!check)

{

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

}

break;

}

return prev;

}

list\* printdoc(list\* prev)

{

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

int i,ch,cn,broi=0,check;

char number[7];

for(check=0;;spisak=prev)

{

printf("Въведете номер: ");

fflush(stdin);

fgets(number,7,stdin);

if ( number[0]>'Я' || number[0]<'А' || number[1]>'Я' || number[1]<'А'

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

|| number[4]>'9' || number[4]<'0' ||number[5]>'Я' || number[5]<'А' )

{

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

continue;

}

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

{

if(strcmp(spisak->kniga.number,number)==0)

{

printf("Номер: %s",spisak->kniga.number);

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

printf("\nДата на издаване: %d.%d.%d",spisak->kniga.day,spisak->kniga.mount,spisak->kniga.year);

printf("\nАвтор: %s %s %s",spisak->kniga.avtorime,spisak->kniga.avtorprezime,spisak->kniga.avtorfamiliq);

check=1;

break;

}

if(spisak->next==NULL)

{

break;

}

}

if(!check)

{

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

}

break;

}

return prev;

}

list\* filee(list\* prev)

{

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

int i,ch,cn,broi=0,check;

char foutname[256];

char finname[256];

FILE \*fin, \*fout;

for(check=0;;)

{

printf("Въведете файл за запис: ");

fflush(stdin);

scanf("%s",foutname);

i=strlen(foutname);

if(foutname[i-1]=='t'&&foutname[i-2]=='x'&&foutname[i-3]=='t')

{

fout=fopen(foutname,"w");

if(fout!=NULL)

{

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

{

if(i!=0)fprintf(fout,"\n");

fprintf(fout,"%s",spisak->kniga.number);

fprintf(fout,"\n%s",spisak->kniga.name);

fprintf(fout,"%d.%d.%d\n",spisak->kniga.day,spisak->kniga.mount,spisak->kniga.year);

fprintf(fout,"%s%s%s",spisak->kniga.avtorime,spisak->kniga.avtorprezime,spisak->kniga.avtorfamiliq);

check=1;

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

}

}

else

{

printf("\nФайла не беше отворен.");

}

break;

}

if(foutname[i-1]=='t'&&foutname[i-2]=='a'&&foutname[i-3]=='d')

{

fout=fopen(foutname,"a+b");

if(fout!=NULL)

{

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

{

fwrite(&(spisak->kniga),sizeof(DOC),1,fout);

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

}

check=1;

}

else

{

printf("\nФайла не беше отворен.");

}

break;

}

if(!check)

{

printf("\nГрешка при отваряне на файла.Трябва да е от тип .txt или .dat\n");

}

}

fclose(fout);

return prev;

}

list\* filereed(list\* prev)

{

DOC knijka, \*sort;

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

int i,ch,cn,broi=0,check;

char foutname[256];

char finname[256];

FILE \*fin, \*fout;

for(broi=0,check=0;;)

{

printf("Въведете файл за четене: ");

fflush(stdin);

scanf("%s",finname);

i=strlen(finname);

if(finname[i-1]=='t'&&finname[i-2]=='x'&&finname[i-3]=='t')

{

fin=fopen(finname,"r");

if(fin!=NULL)

{

check=1;

free(spisak);

prev=NULL;

do{

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

spisak->next=prev;

prev=spisak;

for(i=0;;i++)

{

ch=fgetc(fin);

if(ch!='\n')

{

spisak->kniga.number[i]=ch;

}

else

{

spisak->kniga.number[i]='\0';

break;

}

}

for(i=0;;i++)

{

ch=fgetc(fin);

if(ch!='\n')

{

spisak->kniga.name[i]=ch;

}

else

{

spisak->kniga.name[i]='\0';

break;

}

}

//////////////////////////

ch=fgetc(fin);

cn=fgetc(fin);

if(cn=='.') spisak->kniga.day=(int)(ch-48);

else

{

spisak->kniga.day=(int)(ch-48)\*10+(cn-48);

ch=fgetc(fin);

}

ch=fgetc(fin);

cn=fgetc(fin);

if(cn=='.') spisak->kniga.mount=(int)(ch-48);

else

{

spisak->kniga.mount=(int)(ch-48)\*10+(cn-48);

ch=fgetc(fin);

}

for(spisak->kniga.year=0,i=1000;;i=i/10)

{

ch=fgetc(fin);

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

spisak->kniga.year=spisak->kniga.year+(ch-48)\*i;

}

if(i!=0) spisak->kniga.year=spisak->kniga.year/(i\*10);

for(i=0;;i++)

{

ch=fgetc(fin);

if(ch!='\n')

{

spisak->kniga.avtorime[i]=ch;

}

else

{

spisak->kniga.avtorime[i]='\0';

break;

}

}

for(i=0;;i++)

{

ch=fgetc(fin);

if(ch!='\n')

{

spisak->kniga.avtorprezime[i]=ch;

}

else

{

spisak->kniga.avtorprezime[i]='\0';

break;

}

}

for(i=0;;i++)

{

ch=fgetc(fin);

if(ch!=EOF&&ch!='\n')

{

spisak->kniga.avtorfamiliq[i]=ch;

}

else

{

spisak->kniga.avtorfamiliq[i]='\0';

broi++;

break;

}

}

}while(ch=fgetc(fin)!=EOF);

}

else

{

printf("\nФайла не беше отворен.");

}

break;

}

if(finname[i-1]=='t'&&finname[i-2]=='a'&&finname[i-3]=='d')

{

fin=fopen(finname,"rb");

if(fin!=NULL)

{

free(spisak);

spisak=NULL;

prev=NULL;

do{

if(!fread(&knijka,sizeof(DOC),1,fin))

{

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

spisak->next=prev;

prev=spisak;

spisak->kniga.year=knijka.year;

spisak->kniga.day=knijka.day;

spisak->kniga.mount=knijka.mount;

strcpy(spisak->kniga.name,knijka.name);

strcpy(spisak->kniga.avtorime,knijka.avtorime);

strcpy(spisak->kniga.avtorprezime,knijka.avtorprezime);

strcpy(spisak->kniga.avtorfamiliq,knijka.avtorfamiliq);

broi++;

}

else

{

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

break;

}

}while(1);

check=1;

}

else

{

printf("\nФайла не беше отворен.");

}

break;

}

if(!check)

{

printf("\nГрешка при отваряне на файла.Трябва да е от тип .txt или .dat\n");

}

}

if(fin!=NULL) fclose(fin);

return prev;

}

list\* printall(list\* prev)

{

list\* spisak;

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

{

printf("номер: %s",spisak->kniga.number);

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

printf("\nДата на издаване: %d.%d.%d",spisak->kniga.day,spisak->kniga.mount,spisak->kniga.year);

printf("\nАвтор: %s %s %s\n\n\n\n",spisak->kniga.avtorime,spisak->kniga.avtorprezime,spisak->kniga.avtorfamiliq);

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

}

return prev;

}

list\* podata(list\* prev)

{

DOC knijka, \*sort;

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

int i,ch,cn,broi=0,check;

char number[7];

char name[31];

int day, mount, year;

printf("Въведете днешна дата!\n\n");

do{

printf("Въведете година: ");

fflush(stdin);

scanf("%d",&year);

if(year>2013 || year<1)

{

printf("Въведете година между 1 и 2013.\n");

continue;

}

else

{

break;

}

}while(1);

do{

printf("Въведете месец: ");

fflush(stdin);

scanf("%d",&mount);

if(mount<1 || mount>12)

{

printf("Невалидна дата\n");

continue;

}

else

{

break;

}

}while(1);

do{

printf("Въведете ден: ");

fflush(stdin);

scanf("%d",&day);

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

{

if(day>31||day<1)

{

printf("Невалидна дата\n");

continue;

}

else

{

break;

}

}

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

{

if(day>30||day<1)

{

printf("Невалидна дата\n");

continue;

}

else

{

break;

}

}

if(mount==2)

{

if(year%4==0)

{

if(day>29||day<1)

{

printf("Невалидна дата\n");

continue;

}

else

{

break;

}

}

else

{

if(day>28||day<1)

{

printf("Невалидна дата\n");

continue;

}

else

{

break;

}

}

}

}while(1);

do{

printf("Въведете последните N дни: ");

scanf("%d",&cn);

}while(cn<=0);

for(;;)

{

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

{

if(day<cn)

{

cn=cn-day;

mount=mount-1;

switch(mount)

{

case 0:

day=31;

mount=12;

year=year-1;

break;

case 2:

if(year%4==0) day=29;

else day=28;

break;

case 4:

day=30;

break;

case 6:

day=30;

break;

case 7:

day=31;

break;

case 9:

day=30;

break;

case 11:

day=30;

break;

}

continue;

}

else

{

day=day-cn;

break;

}

}

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

{

if(day<cn)

{

cn=cn-day;

mount=mount-1;

day=31;

continue;

}

else

{

day=day-cn;

break;

}

}

if(mount==2)

{

if(day<cn)

{

cn=cn-day;

mount=mount-1;

day=31;

continue;

}

else

{

day=day-cn;

break;

}

}

}

printf("\n\nВсички документи след %d.%d.%d:\n\n\n",day,mount,year);

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

{

if(year<spisak->kniga.year)

{

printf("номер: %s",spisak->kniga.number);

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

printf("\nДата на издаване: %d.%d.%d",spisak->kniga.day,spisak->kniga.mount,spisak->kniga.year);

printf("\nАвтор: %s %s %s\n\n\n\n",spisak->kniga.avtorime,spisak->kniga.avtorprezime,spisak->kniga.avtorfamiliq);

}

else if(mount<spisak->kniga.mount&&year==spisak->kniga.year)

{

printf("номер: %s",spisak->kniga.number);

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

printf("\nДата на издаване: %d.%d.%d",spisak->kniga.day,spisak->kniga.mount,spisak->kniga.year);

printf("\nАвтор: %s %s %s\n\n\n\n",spisak->kniga.avtorime,spisak->kniga.avtorprezime,spisak->kniga.avtorfamiliq);

}

else if(spisak->kniga.day>day&&year==spisak->kniga.year&&mount==spisak->kniga.mount)

{

printf("номер: %s",spisak->kniga.number);

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

printf("\nДата на издаване: %d.%d.%d",spisak->kniga.day,spisak->kniga.mount,spisak->kniga.year);

printf("\nАвтор: %s %s %s\n\n\n\n",spisak->kniga.avtorime,spisak->kniga.avtorprezime,spisak->kniga.avtorfamiliq);

}

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

}

return prev;

}

list\* sortt(list\* prev)

{

DOC knijka, \*sort;

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

int i,ch,cn,broi=0,check;

char number[7];

char name[31];

int day, mount, year;

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

{

if(spisak->next==NULL)

{

if(spisak!=NULL) broi++;

break;

}

else

{

broi++;

}

}

spisak=prev;

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

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

{

sort[i].year=spisak->kniga.year;

sort[i].day=spisak->kniga.day;

sort[i].mount=spisak->kniga.mount;

strcpy(sort[i].number,spisak->kniga.number);

strcpy(sort[i].name,spisak->kniga.name);

strcpy(sort[i].avtorime,spisak->kniga.avtorime);

strcpy(sort[i].avtorprezime,spisak->kniga.avtorprezime);

strcpy(sort[i].avtorfamiliq,spisak->kniga.avtorfamiliq);

}

do{

check=1;

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

{

if(i<broi-1)

{

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

{

knijka.year=sort[i].year;

knijka.day=sort[i].day;

knijka.mount=sort[i].mount;

strcpy(knijka.number,sort[i].number);

strcpy(knijka.name,sort[i].name);

strcpy(knijka.avtorime,sort[i].avtorime);

strcpy(knijka.avtorprezime,sort[i].avtorprezime);

strcpy(knijka.avtorfamiliq,sort[i].avtorfamiliq);

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

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

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

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

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

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

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

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

sort[i+1].year=knijka.year;

sort[i+1].day=knijka.day;

sort[i+1].mount=knijka.mount;

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

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

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

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

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

check=0;

}

}

}

}while(!check);

do{

check=1;

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

{

if(i<broi-1)

{

if(sort[i].year==sort[i+1].year&&sort[i].mount>sort[i+1].mount)

{

knijka.year=sort[i].year;

knijka.day=sort[i].day;

knijka.mount=sort[i].mount;

strcpy(knijka.number,sort[i].number);

strcpy(knijka.name,sort[i].name);

strcpy(knijka.avtorime,sort[i].avtorime);

strcpy(knijka.avtorprezime,sort[i].avtorprezime);

strcpy(knijka.avtorfamiliq,sort[i].avtorfamiliq);

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

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

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

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

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

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

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

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

sort[i+1].year=knijka.year;

sort[i+1].day=knijka.day;

sort[i+1].mount=knijka.mount;

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

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

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

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

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

check=0;

}

}

}

}while(!check);

do{

check=1;

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

{

if(i<broi-1)

{

if(sort[i].year==sort[i+1].year&&sort[i].mount==sort[i+1].mount&&sort[i].day>sort[i+1].day)

{

knijka.year=sort[i].year;

knijka.day=sort[i].day;

knijka.mount=sort[i].mount;

strcpy(knijka.number,sort[i].number);

strcpy(knijka.name,sort[i].name);

strcpy(knijka.avtorime,sort[i].avtorime);

strcpy(knijka.avtorprezime,sort[i].avtorprezime);

strcpy(knijka.avtorfamiliq,sort[i].avtorfamiliq);

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

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

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

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

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

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

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

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

sort[i+1].year=knijka.year;

sort[i+1].day=knijka.day;

sort[i+1].mount=knijka.mount;

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

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

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

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

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

check=0;

}

}

}

}while(!check);

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

{

printf("номер: %s",sort[i].number);

printf("\nИме: %s",sort[i].name);

printf("\nДата на издаване: %d.%d.%d",sort[i].day,sort[i].mount,sort[i].year);

printf("\nАвтор: %s %s %s\n",sort[i].avtorime,sort[i].avtorprezime,sort[i].avtorfamiliq);

}

return prev;

}

int main(void)

{

list \*prev=NULL;

int ch;

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

for(;;)

{

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

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

printf ("3. Изписване на информацията за документ на екрана.\n");

printf ("4. Запис на данните за документите в файл.\n");

printf ("5. Прочитане на данните за документите от файл.\n");

printf ("6. Изписване на данните за всички документи на екрана.\n");

printf ("7. Извеждане на всички документи в последните N дни.\n");

printf ("8. Сортиране по дата.\n");

printf ("9. ИЗХОД \n");

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

fflush(stdin);

ch=getche();

switch(ch)

{

case '1':

system("cls");

prev=addel(prev);

system("Pause");

system("cls");

break;

case '2':

system("cls");

prev=delel(prev);

system("Pause");

system("cls");

break;

case '3':

system("cls");

prev=printdoc(prev);

system("Pause");

system("cls");

break;

case '4':

system("cls");

prev=filee(prev);

system("Pause");

system("cls");

break;

case '5':

system("cls");

prev=filereed(prev);

system("Pause");

system("cls");

break;

case '6':

system("cls");

prev=printall(prev);

system("Pause");

system("cls");

break;

case '7':

system("cls");

prev=podata(prev);

system("Pause");

system("cls");

break;

case '8':

system("cls");

prev=sortt(prev);

system("Pause");

system("cls");

break;

case '9':

printf("\n\n");

system("Pause");

return 0;

default:

system("cls");

break;

}

}

}