#include <stdio.h>

#include <conio.h>

#include <windows.h>

#include <string.h>

#include <stdlib.h>

typedef struct{

unsigned long long EGN;

char adres[70];

int zaplata;

int year;

int mount;

int day;

char ime[21];

char prezime[21];

char familiq[21];

}RABOTNIK;

typedef struct list{

RABOTNIK chovek;

struct list \*next;

}list;

list\* addel(list\* prev)

{

list \*spisak=NULL;

int i,check,greshka;

unsigned long long EGN;

unsigned long long EGN2,MESEC,DEN;

char adres[70];

int zaplata;

int year;

int mount;

int day;

char ime[21];

char prezime[21];

char familiq[21];

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

spisak->next=prev;

prev=spisak;

for(spisak=prev;;spisak=prev)

{

printf("Въведете ЕГН: ");// na rodeni predi 2000 i sled 1910

fflush(stdin);

scanf("%llu",&EGN);

if(EGN<1000000000||EGN>9999999999)

{

continue;

}

for(greshka=1;;)

{

EGN2=EGN;

EGN2=EGN2/100000000;

EGN2=EGN2+1900;

MESEC=EGN/1000000;

MESEC=MESEC%100;

DEN=EGN/10000;

DEN=DEN%100;

if(EGN2<1913)

{

greshka=0;

break;

}

else if(MESEC>12||MESEC<1)

{

greshka=0;

break;

}

else if(EGN2)

{

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

{

if(DEN>31||DEN<1)

{

greshka=0;

break;

}

else

{

break;

}

}

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

{

if(DEN>30||DEN<1)

{

greshka=0;

break;

}

else

{

break;

}

}

if(MESEC==2)

{

if(EGN2%4==0)

{

if(DEN>29||DEN<1)

{

greshka=0;

break;

}

else

{

break;

}

}

else

{

if(DEN>28||DEN<1)

{

greshka=0;

break;

}

else

{

break;

}

}

break;

}

}

}

if(!greshka)

{

printf("Въведете коректно ЕГН!!");

continue;

}

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

{

if(EGN==spisak->chovek.EGN)break;

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

}

if(EGN==spisak->chovek.EGN)

{

printf("\n\nВече има работник с такъв ЕГН!\n\n ");

continue;

}

else

{

break;

}

}

spisak=prev;

spisak->chovek.EGN=EGN;

do{

check=0;

printf("Въведете адрес на работника: ");

fflush(stdin);

fgets(adres,71,stdin);

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

{

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

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

(!(adres[i]>='А'&&adres[i]<='Я')) )

{

check=1;

break;

}

}

if(check)

{

check=0;

continue;

}

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

{

break;

}

}while(1);

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

do{

printf("Въведете заплата на работника: ");

fflush(stdin);

scanf("%d",&zaplata);

if(zaplata<=0)

{

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

continue;

}

else

{

break;

}

}while(1);

do{

printf("Въведете година на пастъпване на работа: ");

fflush(stdin);

scanf("%d",&year);

if(year<1900)

{

printf("Въведете число по голямо от 1900.\n");

continue;

}

else

{

break;

}

}while(1);

do{

printf("Въведете месец на пастъпване на работа: ");

fflush(stdin);

scanf("%d",&mount);

if(mount<1 || mount>12)

{

printf("Невалидна дата\n");

continue;

}

else

{

break;

}

}while(1);

do{

printf("Въведете ден на пастъпване на работа: ");

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->chovek.zaplata=zaplata;

spisak->chovek.year=year;

spisak->chovek.day=day;

spisak->chovek.mount=mount;

printf("Въведете име на работника: ");

fflush(stdin);

scanf("%s",ime);

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

printf("Въведете презиме на работника: ");

fflush(stdin);

scanf("%s",prezime);

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

printf("Въведете фамилия на работника: ");

fflush(stdin);

scanf("%s",familiq);

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

return prev;

}

list\* delel(list\* prev)

{

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

int check,greshka;

unsigned long long EGN,EGN2,MESEC,DEN;

for(check=0;;)

{

for(spisak=prev;;spisak=prev)

{

printf("Въведете ЕГН: ");// na rodeni predi 2000 i sled 1910

fflush(stdin);

scanf("%llu",&EGN);

if(EGN<1000000000||EGN>9999999999)

{

continue;

}

for(greshka=1;;)

{

EGN2=EGN;

EGN2=EGN2/100000000;

EGN2=EGN2+1900;

MESEC=EGN/1000000;

MESEC=MESEC%100;

DEN=EGN/10000;

DEN=DEN%100;

if(EGN2<1913)

{

greshka=0;

break;

}

else if(MESEC>12||MESEC<1)

{

greshka=0;

break;

}

else if(EGN2)

{

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

{

if(DEN>31||DEN<1)

{

greshka=0;

break;

}

else

{

break;

}

}

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

{

if(DEN>30||DEN<1)

{

greshka=0;

break;

}

else

{

break;

}

}

if(MESEC==2)

{

if(EGN2%4==0)

{

if(DEN>29||DEN<1)

{

greshka=0;

break;

}

else

{

break;

}

}

else

{

if(DEN>28||DEN<1)

{

greshka=0;

break;

}

else

{

break;

}

}

break;

}

}

}

if(!greshka)

{

printf("Въведете коректно ЕГН!!");

continue;

}

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

{

if(spisak->chovek.EGN==EGN)

{

if(head2!=NULL)

{

head2->next=spisak->next;

}

else

{

prev=spisak->next;

}

check=1;

printf("\nУспешно изтрит!\n\n");

free(spisak);

break;

}

if(spisak->next==NULL)

{

break;

}

}

if(!check)

{

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

}

break;

}

return prev;

}

}

list\* printdoc(list\* prev)

{

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

int i,check;

char ime[21];

char prezime[21];

char familiq[21];

unsigned long long EGN;

do{

printf("За извеждане по име натиснете 2, а за извеждане по ЕГН натиснете 1 ");

scanf("%d",&i);

if(i==1||i==2)

{

break;

}

else

{

printf("\nВъведете 1 или 2\n");

}

}while(1);

switch(i)

{

case 1:

for(check=0;;spisak=prev)

{

printf("Въведете ЕГН: ");

fflush(stdin);

scanf("%llu",&EGN);

if(EGN<1000000000||EGN>999999999999)continue;

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

{

if(spisak->chovek.EGN==EGN)

{

printf("\n\n\n\nЕГН: %llu\nАдрес: %s",spisak->chovek.EGN,spisak->chovek.adres);

printf("\nЗаплата: %d",spisak->chovek.zaplata);

printf("\nДата на постъпване на работа: %d.%d.%d",spisak->chovek.day,spisak->chovek.mount,spisak->chovek.year);

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

check=1;

break;

}

if(spisak->next==NULL)

{

break;

}

}

if(!check)

{

printf("\nНе беше намерен работник с това ЕГН.\n\n");

}

break;

}

break;

case 2:

printf("Въведете име на работника: ");

fflush(stdin);

scanf("%s",ime);

printf("Въведете презиме на работника: ");

fflush(stdin);

scanf("%s",prezime);

printf("Въведете фамилия на работника: ");

fflush(stdin);

scanf("%s",familiq);

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

{

if( (strcmp(spisak->chovek.ime,ime)==0) && (strcmp(spisak->chovek.prezime,prezime)==0) && (strcmp(spisak->chovek.familiq,familiq)==0) )

{

printf("\n\n\n\nЕГН: %llu\nАдрес: %s",spisak->chovek.EGN,spisak->chovek.adres);

printf("\nЗаплата: %d",spisak->chovek.zaplata);

printf("\nДата на постъпване на работа: %d.%d.%d",spisak->chovek.day,spisak->chovek.mount,spisak->chovek.year);

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

check=1;

}

if(spisak->next==NULL)

{

break;

}

}

if(!check)

{

printf("\nНе беше намерен работник с това име.\n\n");

}

break;

}

return prev;

}

list\* filee(list\* prev)

{

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

int i,broi=0,check;

char foutname[256];

FILE \*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,"Адрес: %s\n",spisak->chovek.adres);

fprintf(fout,"ЕГН: %llu",spisak->chovek.EGN);

fprintf(fout,"\nЗаплата: %d",spisak->chovek.zaplata);

fprintf(fout,"\nДата на постъпване на работа: %d.%d.%d",spisak->chovek.day,spisak->chovek.mount,spisak->chovek.year);

fprintf(fout,"\nИме: %s %s %s\n",spisak->chovek.ime,spisak->chovek.prezime,spisak->chovek.familiq);

check=1;

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

}

}

else

{

printf("\nФайла не беше отворен.");

}

break;

}

if(!check)

{

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

}

}

fclose(fout);

return prev;

}

list\* filereed(list\* prev)

{

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

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

char foutname[256];

char finname[256];

FILE \*fin;

for(spisak=prev,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;

if(spisak!=NULL)

{

for(spisak=prev;;spisak=head2)

{

head2=spisak->next;

free(spisak);

if(head2==NULL)break;

}

}

prev=NULL;

do{

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

spisak->next=prev;

prev=spisak;

for(a=0;a<7;a++)

{

fgetc(fin);

}

for(i=0;;i++)

{

ch=fgetc(fin);

if(ch!='\n')

{

spisak->chovek.adres[i]=ch;

}

else

{

spisak->chovek.adres[i]='\0';

break;

}

}

for(a=0;a<5;a++)

{

fgetc(fin);

}

fscanf(fin,"%llu",&spisak->chovek.EGN);

for(a=0;a<9;a++)

{

fgetc(fin);

}

fscanf(fin,"%llu",&spisak->chovek.zaplata);

for(a=0;a<31;a++)

{

fgetc(fin);

}

ch=fgetc(fin);

cn=fgetc(fin);

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

else

{

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

ch=fgetc(fin);

}

ch=fgetc(fin);

cn=fgetc(fin);

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

else

{

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

ch=fgetc(fin);

}

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

{

ch=fgetc(fin);

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

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

}

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

for(a=0;a<5;a++)

{

fgetc(fin);

}

for(i=0;;i++)

{

ch=fgetc(fin);

if(ch!=' ')

{

spisak->chovek.ime[i]=ch;

}

else

{

spisak->chovek.ime[i]='\0';

break;

}

}

for(i=0;;i++)

{

ch=fgetc(fin);

if(ch!=' ')

{

spisak->chovek.prezime[i]=ch;

}

else

{

spisak->chovek.prezime[i]='\0';

break;

}

}

for(i=0;;i++)

{

ch=fgetc(fin);

if(ch!='\n')

{

spisak->chovek.familiq[i]=ch;

}

else

{

spisak->chovek.familiq[i]='\0';

break;

}

}

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

}

}

if(!check)

{

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

}

if(ch==EOF)break;

}

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

return prev;

}

list\* printall(list\* prev)

{

list\* spisak;

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

{

printf("\n\n\n\nЕГН: %llu\nАдрес: %s",spisak->chovek.EGN,spisak->chovek.adres);

printf("\nЗаплата: %d",spisak->chovek.zaplata);

printf("\nДата на постъпване на работа: %d.%d.%d",spisak->chovek.day,spisak->chovek.mount,spisak->chovek.year);

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

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

}

return prev;

}

list\* pensiq(list\* prev)

{

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

int greshka;

int broi=0, day, mount, year;

int dolo,gore;

unsigned long long EGN2,MESEC,DEN;

printf("Въведете днешна дата!\n\n");

do{

printf("Въведете година: ");

fflush(stdin);

scanf("%d",&year);

if(year<1990)

{

printf("Въведете година след 1990.\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);

for(greshka=1,spisak=prev;;spisak=spisak->next)

{

EGN2=spisak->chovek.EGN;

EGN2=EGN2/100000000;

EGN2=EGN2+1900;

MESEC=(spisak->chovek.EGN)/1000000;

MESEC=MESEC%100;

DEN=(spisak->chovek.EGN)/10000;

DEN=DEN%100;

if(EGN2>year)

{

greshka=0;

break;

}

else if(MESEC>mount&&year==EGN2)

{

greshka=0;

break;

}

else if(DEN>day&&year==EGN2&&MESEC==mount)

{

greshka=0;

break;

}

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

}

if(!greshka)

{

printf("Въведете коректна днешна дата! Не може да има човек роден след днешната дата!!");

return prev;

}

printf("Хората родени между %d.%d.%d и %d.%d.%d: ",day,mount,year-65,day,mount,year-62);

system("Pause");

for(dolo=year-65,gore=year-62,broi=0,spisak=prev;spisak!=NULL;spisak=spisak->next)

{

EGN2=spisak->chovek.EGN;

EGN2=EGN2/100000000;

EGN2=EGN2+1900;

MESEC=(spisak->chovek.EGN)/1000000;

MESEC=MESEC%100;

DEN=(spisak->chovek.EGN)/10000;

DEN=DEN%100; //DEN.MESEC.EGN2 rojdenna data

if(EGN2>=dolo&&EGN2<=gore)

{

if(EGN2>dolo&&EGN2<gore)

{

broi++;

continue;

}

if(EGN2==dolo)

{

if(MESEC>mount)

{

broi++;

continue;

}

if(MESEC==mount)

{

if(DEN>day)

{

broi++;

continue;

}

else

{

continue;

}

}

else

{

continue;

}

}

if(EGN2==gore)

{

if(MESEC<mount)

{

broi++;

continue;

}

if(MESEC==mount)

{

if(DEN<day)

{

broi++;

continue;

}

else

{

continue;

}

}

else

{

continue;

}

}

}

}

printf("Броят е %d",broi);

return prev;

}

list\* staj(list\* prev)

{

RABOTNIK knijka, \*sort;

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

int i,broi,check;

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

{

if(spisak->next==NULL)

{

if(spisak!=NULL) broi++;

break;

}

else

{

broi++;

}

}

spisak=prev;

if(broi==0) return prev;

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

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

{

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

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

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

strcpy(sort[i].adres,spisak->chovek.adres);

sort[i].zaplata=spisak->chovek.zaplata;

sort[i].EGN=spisak->chovek.EGN;

strcpy(sort[i].ime,spisak->chovek.ime);

strcpy(sort[i].prezime,spisak->chovek.prezime);

strcpy(sort[i].familiq,spisak->chovek.familiq);

}

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;

knijka.zaplata=sort[i].zaplata;

knijka.EGN=sort[i].EGN;

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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;

knijka.zaplata=sort[i].zaplata;

knijka.EGN=sort[i].EGN;

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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;

knijka.zaplata=sort[i].zaplata;

knijka.EGN=sort[i].EGN;

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

check=0;

}

}

}

}while(!check);

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

{

printf("ЕГН: %llu",sort[i].EGN);

printf("\nАдрес: %s",sort[i].adres);

printf("\nЗаплата: %d",sort[i].zaplata);

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

printf("\nИме: %s %s %s\n\n\n\n\n\n",sort[i].ime,sort[i].prezime,sort[i].familiq);

}

return prev;

}

int main(void)

{

list \*prev=NULL;

int ch;

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

for(;;)

{

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

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

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

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

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

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

printf ("7. Извеждане на броя работници на 3 години от пенсиониране.\n");

printf ("8. Сортиране по стаж на работниците.\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=pensiq(prev);

system("Pause");

system("cls");

break;

case '8':

system("cls");

prev=staj(prev);

system("Pause");

system("cls");

break;

case '9':

printf("\n\n");

system("Pause");

return 0;

default:

system("cls");

break;

}

}

}