#include<stdio.h>

#include<stdlib.h>

//Definizione della struttura DatiClienti

struct DatiClienti

{

unsigned int NumeroDiConto;//Numero di conto del cliente

char Cognome[25];//Nome del cliente

char Nome[25];//Cognome del cliente

int SaldoDelConto;// Saldo del conto

};

unsigned int ImmettereScelta(void);

void CreazioneFile(FILE \*LeggereFile);

void AggiornareRecord(FILE \*PuntatoreAFile);

void NuovoRecord(FILE \*PuntatoreAFile);

void CancellareRecord(FILE \*PuntatoreAFile);

int main()

{

FILE \*PuntatoreAFileCrediti;//puntatore al file Crediti

unsigned int Scelta;//Variabile che memorizza la scelta dell'utente

PuntatoreAFileCrediti=fopen("Crediti.doc","rb+");//aperura del file in mdalità lettura scrittura

if(PuntatoreAFileCrediti==NULL)

printf("\n Il file non è stato aperto correttamente\n");

else

{

//Si concede all'utente di specificare l'azione che si vuole effettuare

Scelta=ImmettereScelta();

while(Scelta!=5)

{

switch(Scelta)

{

//Crea un file di testo dal file di record

case 1:

CreazioneFile(PuntatoreAFileCrediti);

break;

//Aggiorna un record

case 2:

AggiornareRecord(PuntatoreAFileCrediti);

break;

//Crea un nuovo record

case 3:

NuovoRecord(PuntatoreAFileCrediti);

break;

//Cancella un record

case 4:

CancellareRecord(PuntatoreAFileCrediti);

break;

//Messaggio di errore

default:

printf("\n Scelta non corretta\n");

break;

}//Fine dello switch

}

fclose(PuntatoreAFileCrediti);//Si chiude il file crediti

}

}

//funzione per immettere la scelta

unsigned int ImmettereScelta(void)

{

unsigned int MenuScelta;//Scelta da parte dell'utente

//Si stamapano le opzioni disponibili

printf("%s", "\nImmetere la sua scelta\n""1 - Memorizzare un file di testo formattato di account per la stampa\n""2 - Aggiornare un account di un cliente\n""3 - Aggiungere un nuovo account\n""4 - Cancellare un account\n""5 - Fine programma\n");

scanf("%u",&MenuScelta);//Ricevi la risposta dall'utente

return (MenuScelta);

}

//Funzione per la creazione del file

void CreazioneFile(FILE \*LeggereFile)

{

FILE \*ScrivereFile;

int Risultato;//variabile per verificare se fread ha letto byte dal file

//crea un oggetto DcatiCliente con informazioni predefinite

struct DatiClienti Cliente=(0,"","",0);

ScrivereFile=fopen("Profili.doc","w");

if(ScrivereFile==NULL)

printf("\nImpossibile aprire file\n");

else

{

rewind(LeggereFile);//sposta il puntatore all'inizio del file

fprintf(ScrivereFile,"%-6s%-16s%-11s%10s\n","Numero di conto","Nome","Cognome","Saldo");

//Copia tutti i record su file di testo

while(!feof(LeggereFile))

{

Risultato=fread(&Cliente,sizeof(struct DatiClienti),1,LeggereFile);

//Scrive un singolo record sul file di testo

if((Risultato!=0)&&(Cliente.NumeroDiConto!=0))

{

fprintf(ScrivereFile,"\n%-6d%-16s%-11s%10.2f\n",Cliente.NumeroDiConto,Cliente.Cognome,Cliente.Nome,Cliente.SaldoDelConto);

}

}

fclose(ScrivereFile);

}

}

//Funzione per aggionare il record

void AggiornareRecord(FILE \*PuntatoreAFile)

{

unsigned int Conto;//numero del conto

double Transazione;//ammontare della transazione

struct DatiClienti Cliente={0,"","",0.0};//Crea un oggetto del DatiClienti senza informazioni

//Si immette il numero di conto da aggiornare

printf("%s","\nImmettere numero conto da aggiornare: \n");

scanf("%d",&Conto);

//Sposta il puntatore del file al record corretto

fseek(PuntatoreAFile,(Conto-1)\*sizeof(struct DatiClienti),SEEK\_SET);

//Leggi il record dal file

fread(&Cliente,sizeof(struct DatiClienti),1,PuntatoreAFile);

//Stampa un messaggio di errore se il numero di conto non esiste

if(Cliente.NumeroDiConto==0)

printf("\nIl conto #%d non ha nessuna informazione\n",Conto);

else

{

//aggiorna il record

printf("%-6d%-16s%-11s%10.2f\n\n",Cliente.NumeroDiConto,Cliente.Cognome,Cliente.Nome,Cliente.SaldoDelConto);

//Riciedi l'ammontare della transazione all'utente

printf("%s","\nImmettere accredito(+) o pagamento(-): \n");

scanf("%lf",&Transazione);

Cliente.SaldoDelConto=Cliente.SaldoDelConto+Transazione;//Aggiorna il saldo del record

printf("\n%-6d%-16s%-11s%-10.2f\n", Cliente.NumeroDiConto,Cliente.Cognome,Cliente.Nome,Cliente.SaldoDelConto);

//sposta i puntatore del file al record corretto nel file

fseek(PuntatoreAFile,(Conto-1)\*sizeof(struct DatiClienti),SEEK\_SET);

//Scrive il record aggiornato al posto del vecchio record nel file

fwrite(&Cliente,sizeof(struct DatiClienti),1,PuntatoreAFile);

}

}

void NuovoRecord(FILE \*PuntatoreAFile)

{

struct DatiClienti Cliente={0,"","",0.0};

unsigned int NumeroConto;

//Ottieni numero conto da creare

printf("%s","\n Immettere il nuovo numero di conto: \n");

scanf("%d",&NumeroConto);

//sposta il puntatore del file al record corretto

fseek(PuntatoreAFile,(NumeroConto-1)\*sizeof(struct DatiClienti),SEEK\_SET);

//leggi il record dal file

fread(&Cliente,sizeof(struct DatiClienti),1,PuntatoreAFile);

//Stampa un messaggio di errore se il conto esiste già

if(Cliente.NumeroDiConto!=0)

printf("\n Il conto #%d esiste già\n",Cliente.NumeroDiConto);

else

{

//Crea il record

//L'utente inserisce il cognome,nome e il saldo

printf("%s","\nInserire cognome,nome,saldo del conto\n");

scanf("%14s%9s%lf",Cliente.Cognome,Cliente.Nome,Cliente.SaldoDelConto);

Cliente.NumeroDiConto=NumeroConto;

//sposta il puntatore del file al record corretto

fseek(PuntatoreAFile,(Cliente.NumeroDiConto-1)\*sizeof(struct DatiClienti),SEEK\_SET);

//inserisci il record nel file

fwrite(&Cliente, sizeof(struct DatiClienti),1,PuntatoreAFile);

}

}

void CancellareRecord(FILE \*PuntatoreAFile)

{

struct DatiClienti Cliente;

struct DatiClienti ClienteVuoto={0,"","",0.0};

unsigned int NumeroConto;

//ottieni i numero del conto da cancellare

printf("%s","\n Immettere il numero del conto da cancellare:\n");

scanf("%d",&NumeroConto);

//sposta il puntatore del file nella posizione corretta

fseek(PuntatoreAFile,(NumeroConto-1)\*sizeof(struct DatiClienti),SEEK\_SET);

//Leggi il record dal file

fread(&Cliente,sizeof(struct DatiClienti),1,PuntatoreAFile);

//Stampa un messaggio di errore se il record del file non esiste

if(Cliente.NumeroDiConto==0)

printf("\n Il conto #%d non esiste\n",NumeroConto);

else

{

//Cancellazione del record

//Sposta il puntatore del file nella posizione corretta

fseek(PuntatoreAFile,(NumeroConto-1)\*sizeof(struct DatiClienti),SEEK\_SET);

//Sostituisci il record esistente con il record vuoto

fwrite(&ClienteVuoto,sizeof(struct DatiClienti),1,PuntatoreAFile);

}

}