#include<stdio.h>

#include <stdlib.h>

#include <string.h>

#include <time.h>

#include <unistd.h>

typedef struct contactStruct

{

int contactId;

char contactName[80];

char phoneNumber[30];

char faxNumber[30];

char homeNumber[30];

char email[80];

char address[100];

struct contactStruct \*nextContact;

struct contactStruct \*prevContact;

}strContact;

typedef struct userStruct

{

int userId;

char userName[30];

char userPassword[30];

int userType;

struct userStruct \*nextUser;

struct userStruct \*prevUser;

struct contactStruct \*firstContact;

struct contactStruct \*lastContact;

}strUser;

strUser \*firstUser=NULL;

strUser \*lastUser=NULL;

strUser \*currentUser=NULL;

void loadUsers();

void caesarCipher(char strNonCiph[], char strCiph[]);

void addUser(strUser \*tempUser);

void adminMenu();

void contactsMenu();

void menu3();

void menu4();

int main(int argc, char \*argv[])

{

loadUsers();

int opt,opt2;

char userName[30];

char userPassword[30];

char ciphUserPassword[30];

int foudUser=0;

while(1)

{

printf("Contact Management System\n");

printf("Log in:\n");

printf("Enter user's name: ");

scanf("%s",userName);

printf("Enter user's password: ");

//todo securepassword

scanf("%s",userPassword);

caesarCipher(userPassword,ciphUserPassword);

strUser \*iterU=firstUser;

while(iterU!=NULL)

{

// printf("%s|%s\n",iterU->userPassword,ciphUserPassword);

// if(strcmp(iterU->userName,userName)==0)

// printf("Usermatch\n");

// if(strcmp(iterU->userName,userName)==0)

// printf("Usermatch\n");

if(strcmp(iterU->userName,userName)==0 && strcmp(iterU->userPassword,ciphUserPassword)==0)

{

currentUser=iterU;

if(iterU->userType==0)

{

do

{

adminMenu();

scanf("%d",&opt);

switch(opt)

{

case 1:

registerUser();

break;

case 2:

break;

case 3:

break;

case 4:

listUsers();

break;

case 5:

loadContacts();

do

{

contactsMenu();

scanf("%d",&opt2);

}while(opt2!=5);

break;

}

}while(opt!=7);

}

}

iterU=iterU->nextUser;

}

}

// strUser \*iterU=firstUser;

// printf("\n");

// while(iterU!=NULL)

// {

// printf("uid: %d\n",iterU->userId);

// printf("name: %s\n",iterU->userName);

// printf("pass: %s\n",iterU->userPassword);

// printf("type: %d\n",iterU->userType);

// printf("\n");

// iterU=iterU->nextUser;

// }

}

void adminMenu()

{

printf("Menu: \n");

printf("1. Register user\n");

printf("2. Delete user\n");

printf("3. Modify user\n");

printf("4. List users\n");

printf("5. Contacts\n");

printf("6. Change password\n");

printf("7. Exit\n");

}

void loadContacts()

{

if(currentUser!=NULL)

{

FILE \*contactsFile;

int readItems;

strContact tempContact;

char fileName[50];

sprintf(fileName,"%d%s.txt",currentUser->userId,currentUser->userName);

contactsFile=fopen(fileName,"r");

char readString[200];

// int contactId;

// char contactName[80];

// char phoneNumber[15];

// char faxNumber[15];

// char homeNumber[15];

// char email[80];

// char address[100];

// struct contactStruct \*nextContact;

// struct contactStruct \*prevContact;

if(contactsFile!=NULL)

{

printf("read\n");

do

{

readItems=fscanf(contactsFile,"%d,%[^,],%[^,],%[^,],%[^,],%[^,],%s\n",&tempContact.contactId,tempContact.contactName,tempContact.phoneNumber,tempContact.faxNumber,tempContact.homeNumber,tempContact.email,tempContact.address);

//printf("%d,%s,%s,%s,%s,%s,%s--\n",tempContact.contactId,tempContact.contactName,tempContact.phoneNumber,tempContact.faxNumber,tempContact.homeNumber,tempContact.email,tempContact.address);printf("%d,%s,%s,%s,%s,%s,%s\n",tempContact.contactId,tempContact.contactName,tempContact.phoneNumber,tempContact.faxNumber,tempContact.homeNumber,tempContact.email,tempContact.address);

//getch();

//printf("%s\n",readString);

//sscanf(readString,

if(readItems!=EOF)

{

addContact(&tempContact);

printf("%d | %s | %s | %s | %s | %s | %s\n",tempContact.contactId,tempContact.contactName,tempContact.phoneNumber,tempContact.faxNumber,tempContact.homeNumber,tempContact.email,tempContact.address);

//printf("%d | %s | %s | %d\n",tempUser.userId,tempUser.userName,tempUser.userPassword, tempUser.userType);

}

}while(readItems!=EOF);

fclose(contactsFile);

}

else

{

printf("No contacts file\n");

}

}

}

void addContact(strContact \*tempContact)

{

if(currentUser->firstContact==NULL && currentUser->lastContact==NULL)

{

if(tempContact!=NULL)

{

strUser \*newUser = (strUser \*)malloc(sizeof(strUser));

strContact \*newContact =(strContact \*)malloc(sizeof(strContact));

newContact->contactId=tempContact->contactId;

sprintf(newContact->contactName,tempContact->contactName);

sprintf(newContact->phoneNumber,tempContact->phoneNumber);

sprintf(newContact->faxNumber,tempContact->faxNumber);

sprintf(newContact->homeNumber,tempContact->homeNumber);

sprintf(newContact->email,tempContact->email);

sprintf(newContact->address,tempContact->address);

newContact->nextContact=NULL;

newContact->prevContact=NULL;

currentUser->firstContact=newContact;

currentUser->lastContact=newContact;

}

}

else

{

if(tempContact!=NULL)

{

strUser \*newUser = (strUser \*)malloc(sizeof(strUser));

strContact \*newContact =(strContact \*)malloc(sizeof(strContact));

newContact->contactId=tempContact->contactId;

sprintf(newContact->contactName,tempContact->contactName);

sprintf(newContact->phoneNumber,tempContact->phoneNumber);

sprintf(newContact->faxNumber,tempContact->faxNumber);

sprintf(newContact->homeNumber,tempContact->homeNumber);

sprintf(newContact->email,tempContact->email);

sprintf(newContact->address,tempContact->address);

currentUser->lastContact->nextContact=newContact;

newContact->nextContact=NULL;

newContact->prevContact=currentUser->lastContact;

currentUser->lastContact=newContact;

}

}

}

void registerUser()

{

strUser tempUser;

fflush(stdin);

printf("Enter User's Name: ");

scanf("%s",tempUser.userName);

printf("Enter User's Password: ");

fflush(stdin);

scanf("%s",tempUser.userPassword);

caesarCipher(tempUser.userPassword,tempUser.userPassword);

printf("Enter User's role: \n");

printf("1. Administrator: \n");

printf("2. User: \n");

scanf("%d",&tempUser.userType);

if(tempUser.userType==1)

{

tempUser.userType=0;

}

else

{

tempUser.userType=1;

}

tempUser.userId=lastUser->userId+1;

addUser(&tempUser);

saveUsersFile();

}

void saveUsersFile()

{

strUser \*iterU=firstUser;

FILE \*usersFile;

usersFile=fopen("users.txt","w");

if(usersFile!=NULL)

{

while(iterU!=NULL)

{

//printf("%d,%s,%s,%d\n",iterU->userId,iterU->userName,iterU->userPassword,iterU->userType);

fprintf(usersFile,"%d,%s,%s,%d\n",iterU->userId,iterU->userName,iterU->userPassword,iterU->userType);

iterU=iterU->nextUser;

}

fclose(usersFile);

}

}

void listUsers()

{

strUser \*iterU=firstUser;

printf(" UserId || UserName || User's Role\n",iterU->userId,iterU->userName);

printf("----------------------------------------------------------------\n",iterU->userId,iterU->userName);

while(iterU!=NULL)

{

if(iterU->userType==0)

{

printf(" %03d || %30s || Administrator\n",iterU->userId,iterU->userName);

}

else

printf(" %03d || %30s || User\n",iterU->userId,iterU->userName);

//fprintf(usersFile,"%d,%s,%s,%d\n",iterU->userId,iterU->userName,iterU->userPassword,iterU->userType);

iterU=iterU->nextUser;

}

printf("\n");

}

void loadUsers()

{

FILE \*usersFile;

int readItems;

strUser tempUser;

usersFile=fopen("users.txt","r");

char readString[200];

if(usersFile!=NULL)

{

//printf("read\n");

do

{

readItems=fscanf(usersFile,"%d,%[^,],%[^,],%d\n",&tempUser.userId,tempUser.userName,tempUser.userPassword, &tempUser.userType);

//printf("%s\n",readString);

//sscanf(readString,

if(readItems!=EOF)

{

addUser(&tempUser);

//printf("%d | %s | %s | %d\n",tempUser.userId,tempUser.userName,tempUser.userPassword, tempUser.userType);

}

}while(readItems!=EOF);

fclose(usersFile);

}

else

{

printf("No read");

addUser(NULL);

}

}

void addUser(strUser \*tempUser)

{

if(firstUser==NULL && lastUser==NULL)

{

if(tempUser==NULL)

{

strUser \*newUser = (strUser \*)malloc(sizeof(strUser));

newUser->userId=1;

sprintf(newUser->userName,"Admin");

caesarCipher("Admin",newUser->userPassword);

newUser->userType=0;

newUser->nextUser=NULL;

newUser->prevUser=NULL;

newUser->firstContact=NULL;

newUser->lastContact=NULL;

firstUser=newUser;

lastUser=newUser;

}

else

{

strUser \*newUser = (strUser \*)malloc(sizeof(strUser));

newUser->userId=tempUser->userId;

sprintf(newUser->userName,tempUser->userName);

sprintf(newUser->userPassword,tempUser->userPassword);

newUser->userType=tempUser->userType;

newUser->nextUser=NULL;

newUser->prevUser=NULL;

newUser->firstContact=NULL;

newUser->lastContact=NULL;

firstUser=newUser;

lastUser=newUser;

}

}

else

{

if(tempUser!=NULL)

{

strUser \*newUser = (strUser \*)malloc(sizeof(strUser));

newUser->userId=tempUser->userId;

sprintf(newUser->userName,tempUser->userName);

sprintf(newUser->userPassword,tempUser->userPassword);

newUser->userType=tempUser->userType;

lastUser->nextUser=newUser;

newUser->nextUser=NULL;

newUser->prevUser=lastUser;

lastUser=newUser;

newUser->firstContact=NULL;

newUser->lastContact=NULL;

}

}

}

void caesarCipher(char strNonCiph[], char strCiph[])

{

int i=0;

while(strNonCiph[i]!='\0')

{

strCiph[i]=strNonCiph[i]+3;

i++;

}

strCiph[i]=strNonCiph[i];

}

void contactsMenu()

{

printf("Menu: \n");

printf("1. Register contact\n");

printf("2. List contacts\n");

printf("3. Modify contact\n");

printf("4. Delete contact\n");

printf("5. Exit\n");

}