#include <stdio.h>

#include <stdlib.h>

#include <stdint.h>

#include <string.h>

typedef struct Transaction {

uint32\_t transaction\_id;

char type[10];

double amount;

struct Transaction \*next;

} Transaction;

typedef struct Account {

uint32\_t account\_number;

char account\_name[50];

double balance;

char contact\_number[15];

int transaction\_count;

Transaction \*transactions;

struct Account \*next;

} Account;

Account \*head = NULL;

void create\_account();

void deposit();

void withdraw();

void transfer();

void display\_all\_accounts();

void find\_account();

void save\_data();

void load\_data();

void show\_transaction\_history();

void balance\_enquiry();

void menu();

int main() {

load\_data();

char choice;

while (1) {

menu();

printf("\nEnter your choice: ");

scanf(" %c", &choice);

switch (choice) {

case 'C': case 'c': create\_account(); break;

case 'H': case 'h': show\_transaction\_history(); break;

case 'W': case 'w': withdraw(); break;

case 'D': case 'd': deposit(); break;

case 'B': case 'b': balance\_enquiry(); break;

case 'T': case 't': transfer(); break;

case 'E': case 'e': display\_all\_accounts(); break;

case 'S': case 's': save\_data(); break;

case 'F': case 'f': find\_account(); break;

case 'Q': case 'q': save\_data(); exit(0);

default: printf("Invalid choice! Try again.\n");

}

}

return 0;

}

void menu() {

printf("\n------------------MENU---------------------------\n");

printf("C/c: Create account\nH/h: Transaction history\nW/w: Withdraw\n");

printf("D/d: Deposit\nB/b: Balance enquiry\nT/t: Transfer money\n");

printf("E/e: Display all accounts\nS/s: Save accounts info\n");

printf("F/f: Find account\nQ/q: Quit\n");

}

void create\_account() {

Account \*new\_acc = (Account \*)malloc(sizeof(Account));

if (!new\_acc) return;

printf("Enter Account Number: ");

scanf("%u", &new\_acc->account\_number);

printf("Enter Account Holder Name: ");

getchar(); fgets(new\_acc->account\_name, sizeof(new\_acc->account\_name), stdin); new\_acc->account\_name[strcspn(new\_acc->account\_name, " ")] ='NULL';

printf("Enter Contact Number: ");

scanf("%s", new\_acc->contact\_number);

new\_acc->balance = 0.0;

new\_acc->transaction\_count = 0;

new\_acc->transactions = NULL;

new\_acc->next = head;

head = new\_acc;

printf("Account created successfully!\n");

}

void save\_data() {

FILE \*fp = fopen("accounts.dat", "wb");

if (!fp) return;

Account \*acc = head;

while (acc) {

fwrite(acc, sizeof(Account), 1, fp);

acc = acc->next;

}

fclose(fp);

}

void load\_data() {

FILE \*fp = fopen("accounts.dat", "rb");

if (!fp) return;

while (1) {

Account \*acc = (Account \*)malloc(sizeof(Account));

if (fread(acc, sizeof(Account), 1, fp) == 1) {

acc->next = head;

head = acc;

} else { free(acc); break; }

}

fclose(fp);

}

void find\_account() {

uint32\_t acc\_no;

printf("Enter Account Number: ");

scanf("%u", &acc\_no);

Account \*acc = head;

while (acc && acc->account\_number != acc\_no) acc = acc->next;

if (!acc) {

printf("Account not found!\n");

return;

}

printf("Account No: %u | Name: %s | Balance: %.2f\n", acc->account\_number, acc->account\_name, acc->balance);

}