#include <stdio.h>

#include <stdlib.h>

#include <string.h>

// Define a structure for a contact

struct Contact {

char name[50];

char phone[15];

struct Contact\* next;

};

// Function to create a new contact

struct Contact\* createContact(const char\* name, const char\* phone) {

struct Contact\* newContact = (struct Contact\*)malloc(sizeof(struct Contact));

if (newContact == NULL) {

printf("Memory allocation failed.\n");

exit(EXIT\_FAILURE);

}

strcpy(newContact->name, name);

strcpy(newContact->phone, phone);

newContact->next = NULL;

return newContact;

}

// Function to add a new contact to the phonebook

void addContact(struct Contact\*\* head, const char\* name, const char\* phone) {

struct Contact\* newContact = createContact(name, phone);

newContact->next = \*head;

\*head = newContact;

printf("Contact added successfully.\n");

}

// Function to search for a contact in the phonebook

void searchContact(const struct Contact\* head, const char\* name) {

const struct Contact\* current = head;

while (current != NULL) {

if (strcmp(current->name, name) == 0) {

printf("Contact found: %s, %s\n", current->name, current->phone);

return;

}

current = current->next;

}

printf("Contact not found.\n");

}

// Function to delete a contact from the phonebook

void deleteContact(struct Contact\*\* head, const char\* name) {

struct Contact\* current = \*head;

struct Contact\* prev = NULL;

while (current != NULL) {

if (strcmp(current->name, name) == 0) {

if (prev == NULL) {

\*head = current->next;

} else {

prev->next = current->next;

}

free(current);

printf("Contact deleted successfully.\n");

return;

}

prev = current;

current = current->next;

}

printf("Contact not found.\n");

}

// Function to display all contacts in the phonebook

void displayContacts(const struct Contact\* head) {

const struct Contact\* current = head;

while (current != NULL) {

printf("%s, %s\n", current->name, current->phone);

current = current->next;

}

}

// Function to free the memory allocated for the contacts

void freeContacts(struct Contact\* head) {

struct Contact\* current = head;

struct Contact\* next;

while (current != NULL) {

next = current->next;

free(current);

current = next;

}

}

int main() {

struct Contact\* phonebook = NULL;

int choice;

char name[50];

char phone[15];

do {

printf("\nPhonebook Menu:\n");

printf("1. Add Contact\n");

printf("2. Search Contact\n");

printf("3. Delete Contact\n");

printf("4. Display Contacts\n");

printf("5. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1:

printf("Enter contact name: ");

scanf("%s", name);

printf("Enter contact phone number: ");

scanf("%s", phone);

addContact(&phonebook, name, phone);

break;

case 2:

printf("Enter contact name to search: ");

scanf("%s", name);

searchContact(phonebook, name);

break;

case 3:

printf("Enter contact name to delete: ");

scanf("%s", name);

deleteContact(&phonebook, name);

break;

case 4:

printf("Contacts in the phonebook:\n");

displayContacts(phonebook);

break;

case 5:

printf("Exiting the phonebook application.\n");

break;

default:

printf("Invalid choice. Please enter a valid option.\n");

}

} while (choice != 5);

// Free the allocated memory for contacts before exiting

freeContacts(phonebook);

return 0;

}