```
            case 2:
                printf("nome: ");
                fgets(name, sizeof(name), stdin);
                name[strcspn(name, "\n")] = '\0';
                searchContact(name);
                break;
            case 3:
                printf("nome: ");
                fgets(name, sizeof(name), stdin);
                name[strcspn(name, "\n")] = '\0';
                removeContact(name);
                break;
            case 4:
                showContacts();
                break;
            case 0:
                printf("saindo...\n");
```

input

```
Escolha uma opção:
1 - adicionar contato
2 - buscar contato por nome
3 - remover contato
4 - exibir todos os contatos
0 - sair
Digite uma opção: 2
Nome: matheus
Telefone de matheus: 41999994142 (tempo de busca: 0.00 ms)
Escolha uma opção:
1 - adicionar contato
2 - buscar contato por nome
3 - remover contato
4 - exibir todos os contatos
0 - sair
Digite uma opção:
```

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>

#define TABLE_SIZE 100

typedef struct Contact {
    char name[50];
    char phone[20];
    struct Contact* next;
} Contact;

Contact* hashTable[TABLE_SIZE];

unsigned int hash(const char* str) {
    unsigned int hash = 0;
    while (*str) {
```

```
        hash = (hash << 5) + *str++;
    }
    return hash % TABLE_SIZE;
}

void initializeTable() {
    for (int i = 0; i < TABLE_SIZE; i++) {
        hashTable[i] = NULL;
    }
}

void addContact(const char* name, const char* phone) {
    unsigned int index = hash(name);
    Contact* newContact = (Contact*)malloc(sizeof(Contact));
    strcpy(newContact->name, name);
    strcpy(newContact->phone, phone);
    newContact->next = hashTable[index];
    hashTable[index] = newContact;
    printf("Contato adicionado com sucesso.\n");
}

void searchContact(const char* name) {
    unsigned int index = hash(name);
    Contact* contact = hashTable[index];
    clock_t start = clock();
    while (contact) {
        if (strcmp(contact->name, name) == 0) {
            clock_t end = clock();
            double time_spent = (double)(end - start) / CLOCKS_PER_SEC
* 1000;
            printf("Telefone de %s: %s (tempo de busca: %.2f ms)\n",
name, contact->phone, time_spent);
            return;
        }
        contact = contact->next;
    }
    printf("Contato não encontrado.\n");
}

void removeContact(const char* name) {
    unsigned int index = hash(name);
    Contact* contact = hashTable[index];
    Contact* prev = NULL;
```

```
    while (contact) {
        if (strcmp(contact->name, name) == 0) {
            if (prev) {
                prev->next = contact->next;
            } else {
                hashTable[index] = contact->next;
            }
            free(contact);
            printf("Contato removido com sucesso.\n");
            return;
        }
        prev = contact;
        contact = contact->next;
    }
    printf("Contato não encontrado.\n");
}

void showContacts() {
    for (int i = 0; i < TABLE_SIZE; i++) {
        Contact* contact = hashTable[i];
        while (contact) {
            printf("Nome: %s, Telefone: %s\n", contact->name,
contact->phone);
            contact = contact->next;
        }
    }
}

int main() {
    initializeTable();
    int option;
    char name[50];
    char phone[20];

    do {
        printf("Escolha uma opção:\n");
        printf("1 - adicionar contato\n");
        printf("2 - buscar contato por nome\n");
        printf("3 - remover contato\n");
        printf("4 - exibir todos os contatos\n");
        printf("0 - sair\n");
        printf("Digite uma opção: ");
```

```
        scanf("%d", &option);
        getchar();

        switch (option) {
            case 1:
                printf("nome: ");
                fgets(name, sizeof(name), stdin);
                name[strcspn(name, "\n")] = '\0';
                printf("telefone: ");
                fgets(phone, sizeof(phone), stdin);
                phone[strcspn(phone, "\n")] = '\0';
                addContact(name, phone);
                break;
            case 2:
                printf("nome: ");
                fgets(name, sizeof(name), stdin);
                name[strcspn(name, "\n")] = '\0';
                searchContact(name);
                break;
            case 3:
                printf("nome: ");
                fgets(name, sizeof(name), stdin);
                name[strcspn(name, "\n")] = '\0';
                removeContact(name);
                break;
            case 4:
                showContacts();
                break;
            case 0:
                printf("saindo...\n");
                break;
            default:
                printf("não existe essa opção\n");
                break;
        }
    } while (option != 0);

    return 0;
}
```