```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
struct Contact {
    char name[50];
    char email[50];
    char phoneNumber[15];
};
void addContact(struct Contact **addressBook, int *size) {
    (*size)++;
if (*size == 1) {
    *addressBook = malloc(sizeof(struct Contact));
} else {
    *addressBook = realloc(*addressBook, (*size) * sizeof(struct Contact));
}

    if (*addressBook == NULL) {
        printf("Memory allocation error.\n");
        exit(EXIT_FAILURE);
    }
    printf("\nEnter details for the new contact:\n");
    printf("Name: ");
    scanf("%s", (*addressBook)[*size - 1].name);
    printf("Email: ");
    scanf("%s", (*addressBook)[*size - 1].email);
    printf("Phone Number: ");
    scanf("%s", (*addressBook)[*size - 1].phoneNumber);

    printf("Contact added successfully!\n");
}
void deleteContact(struct Contact **addressBook, int *size, int index) {
    if (index >= 0 && index < *size) {
        for (int i = index; i < *size - 1; i++) {
            (*addressBook)[i] = (*addressBook)[i + 1];
        }
        (*size)--;
        *addressBook = realloc(*addressBook, (*size) * sizeof(struct Contact));
        if (*size > 0 && *addressBook == NULL) {
            printf("Memory allocation error during deletion.\n");
            exit(EXIT_FAILURE);
        }
        printf("Contact deleted successfully!\n");
    } else {
        printf("Invalid index. No contact deleted.\n");
    }
}
void displayContacts(const struct Contact *addressBook, int size) {
    printf("\nAddress Book:\n");
    for (int i = 0; i < size; i++) {
        printf("Contact %d:\n", i + 1);
        printf("Name: %s\n", addressBook[i].name);
        printf("Email: %s\n", addressBook[i].email);
        printf("Phone Number: %s\n", addressBook[i].phoneNumber);
        printf("\n");
    }
}
void freeAddressBook(struct Contact *addressBook) {
    free(addressBook);
}
int main() {
    struct Contact *addressBook = NULL;
    int size = 0;
    int choice, index;

    do {
        printf("\nAddress Book Menu:\n");
```

```c
        printf("1. Add Contact\n");
        printf("2. Delete Contact\n");
        printf("3. Display Contacts\n");
        printf("4. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                addContact(&addressBook, &size);
                break;
            case 2:
                printf("Enter the index of the contact to delete: ");
                scanf("%d", &index);
                deleteContact(&addressBook, &size, index - 1);
                break;
            case 3:
                displayContacts(addressBook, size);
                break;
            case 4:
                printf("Exiting program. Freeing memory...\n");
                break;
            default:
                printf("Invalid choice. Please enter a valid option.\n");
        }
    } while (choice != 4);
    freeAddressBook(addressBook);

    return 0;
}
```