

main.c



Share

Run

Output

Clear

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 #define MAX 100
6 #define FILENAME "contacts.txt"
7
8 typedef struct {
9     char name[50];
10    char phone[20];
11    char email[50];
12 } Contact;
13
14 Contact contacts[MAX];
15 int contactCount = 0;
16
17 void loadContacts() {
18     FILE *fp = fopen(FILENAME, "r");
19     if (!fp) {
20         printf("No saved contacts found. Starting fresh.\n");
```

```
* No saved contacts found. Starting fresh.
```

```
--- Contact Manager ---
```

1. Add Contact
2. View Contacts
3. Edit Contact
4. Delete Contact
5. Save & Exit

```
Enter choice: |
```

main.c



Share

Run

Output

Clear

```
20     printf("No saved contacts found. Starting fresh.\n");
21     return;
22 }
23 while (fscanf(fp, "%49[^\n],%19[^\n],%49[^\n]\n",
24             contacts[contactCount].name,
25             contacts[contactCount].phone,
26             contacts[contactCount].email) == 3) {
27     contactCount++;
28 }
29 fclose(fp);
30 }
31 void saveContacts() {
32     FILE *fp = fopen(FILENAME, "w");
33     if (!fp) {
34         printf("Error saving contacts.\n");
35         return;
36     }
37     for (int i = 0; i < contactCount; i++) {
38         fprintf(fp, "%s,%s,%s\n", contacts[i].name, contacts[i].
            phone, contacts[i].email);
```

^ No saved contacts found. Starting fresh.

--- Contact Manager ---

1. Add Contact
2. View Contacts
3. Edit Contact
4. Delete Contact
5. Save & Exit

Enter choice:

main.c



Share

Run

Output

Clear

```
38     fprintf(fp, "%s,%s,%s\n", contacts[i].name, contacts[i]
    .phone, contacts[i].email);
39 }
40 fclose(fp);
41 }
42 void addContact() {
43     if (contactCount >= MAX) {
44         printf("Contact list is full!\n");
45         return;
46     }
47     printf("Enter name: ");
48     getchar();
49     fgets(contacts[contactCount].name, sizeof
        (contacts[contactCount].name), stdin);
50     contacts[contactCount].name[strcspn(contacts[contactCount]
        .name, "\n")] = 0;
51
52     printf("Enter phone: ");
53     fgets(contacts[contactCount].phone, sizeof
        (contacts[contactCount].phone), stdin);
```

No saved contacts found. Starting fresh.

--- Contact Manager ---

1. Add Contact
2. View Contacts
3. Edit Contact
4. Delete Contact
5. Save & Exit

Enter choice:

main.c



Share

Run

Output

Clear

```
53     fgets(contacts[contactCount].phone, sizeof
54           (contacts[contactCount].phone), stdin);
55     contacts[contactCount].phone[strlen(contacts[contactCount]
56           .phone, "\n")] = 0;
57     printf("Enter email: ");
58     fgets(contacts[contactCount].email, sizeof
59           (contacts[contactCount].email), stdin);
60     contacts[contactCount].email[strlen(contacts[contactCount]
61           .email, "\n")] = 0;
62     contactCount++;
63     printf("Contact added!\n");
64 }
65 void viewContacts() {
66     if (contactCount == 0) {
67         printf("No contacts to display.\n");
68         return;
69     }
70     printf("\n--- Contact List ---\n");
71     for (int i = 0; i < contactCount; i++) {
72         printf("%d. %s\n", i+1, contacts[i].phone);
73     }
74 }
```

No saved contacts found. Starting fresh.

--- Contact Manager ---

1. Add Contact
2. View Contacts
3. Edit Contact
4. Delete Contact
5. Save & Exit

Enter choice:

main.c



Share

Run

Output

Clear

```
67 printf("\n--- Contact List ---\n");
68 for (int i = 0; i < contactCount; i++) {
69     printf("%d. Name: %s | Phone: %s | Email: %s\n",
70           i + 1, contacts[i].name, contacts[i].phone,
71           contacts[i].email);
72 }
73 void editContact() {
74     viewContacts();
75     if (contactCount == 0) return;
76     int index;
77     printf("Enter contact number to edit: ");
78     scanf("%d", &index);
79     if (index < 1 || index > contactCount) {
80         printf("Invalid contact number.\n");
81         return;
82     }
83     index--;
84     getchar();
85     printf("Enter new name (leave blank to keep %s): ",
```

^ No saved contacts found. Starting fresh.

--- Contact Manager ---

1. Add Contact
2. View Contacts
3. Edit Contact
4. Delete Contact
5. Save & Exit

Enter choice:

=== Session Ended. Please Run the code again ===

main.c



Share

Run

Output

Clear

```
85 printf("Enter new name (leave blank to keep \"%s\"): ",
    contacts[index].name);
86 char newName[50];
87 fgets(newName, sizeof(newName), stdin);
88 if (newName[0] != '\n') {
89     newName[strcspn(newName, "\n")] = 0;
90     strcpy(contacts[index].name, newName);
91 }
92 printf("Enter new phone (leave blank to keep \"%s\"): ",
    contacts[index].phone);
93 char newPhone[20];
94 fgets(newPhone, sizeof(newPhone), stdin);
95 if (newPhone[0] != '\n') {
96     newPhone[strcspn(newPhone, "\n")] = 0;
97     strcpy(contacts[index].phone, newPhone);
98 }
99 printf("Enter new email (leave blank to keep \"%s\"): ",
    contacts[index].email);
100 char newEmail[50];
101 fgets(newEmail, sizeof(newEmail), stdin);
```

\* No saved contacts found. Starting fresh.

--- Contact Manager ---

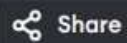
1. Add Contact
2. View Contacts
3. Edit Contact
4. Delete Contact
5. Save & Exit

Enter choice:

=== Session Ended. Please Run the code again ===



main.c



Share

Run

Output

Clear

```
101 fgets(newEmail, sizeof(newEmail), stdin);
102 if (newEmail[0] != '\n') {
103     newEmail[strcspn(newEmail, "\n")] = 0;
104     strcpy(contacts[index].email, newEmail);
105 }
106 printf("Contact updated!\n");
107 }
108 void deleteContact() {
109     viewContacts();
110     if (contactCount == 0) return;
111     int index;
112     printf("Enter contact number to delete: ");
113     scanf("%d", &index);
114     if (index < 1 || index > contactCount) {
115         printf("Invalid contact number.\n");
116         return;
117     }
118     index--;
119     for (int i = index; i < contactCount - 1; i++) {
120         contacts[i] = contacts[i + 1];
```

^ No saved contacts found. Starting fresh.

--- Contact Manager ---

1. Add Contact
2. View Contacts
3. Edit Contact
4. Delete Contact
5. Save & Exit

Enter choice:

=== Session Ended. Please Run the code again ===

main.c



Share

Run

Output

Clear

```
120     contacts[i] = contacts[i + 1];
121 }
122 contactCount--;
123 printf("Contact deleted!\n");
124 }
125 int main() {
126     loadContacts();
127     int choice;
128     do {
129         printf("\n--- Contact Manager ---\n");
130         printf("1. Add Contact\n");
131         printf("2. View Contacts\n");
132         printf("3. Edit Contact\n");
133         printf("4. Delete Contact\n");
134         printf("5. Save & Exit\n");
135         printf("Enter choice: ");
136         scanf("%d", &choice);
137         switch (choice) {
138             case 1: addContact(); break;
139             case 2: viewContacts(): break;
```

^ No saved contacts found. Starting fresh.

--- Contact Manager ---

1. Add Contact
2. View Contacts
3. Edit Contact
4. Delete Contact
5. Save & Exit

Enter choice:

=== Session Ended. Please Run the code again ===



main.c



Share

Run

Output

Clear

```
129 printf("\n--- Contact Manager ---\n");
130 printf("1. Add Contact\n");
131 printf("2. View Contacts\n");
132 printf("3. Edit Contact\n");
133 printf("4. Delete Contact\n");
134 printf("5. Save & Exit\n");
135 printf("Enter choice: ");
136 scanf("%d", &choice);
137 switch (choice) {
138     case 1: addContact(); break;
139     case 2: viewContacts(); break;
140     case 3: editContact(); break;
141     case 4: deleteContact(); break;
142     case 5: saveContacts(); printf("Contacts saved.\n");
        printf("Goodbye!\n"); break;
143     default: printf("Invalid choice.\n");
144 }
145 } while (choice != 5);
146 return 0;
147 }
```

▲ No saved contacts found. Starting fresh.

--- Contact Manager ---

1. Add Contact
2. View Contacts
3. Edit Contact
4. Delete Contact
5. Save & Exit

Enter choice:

=== Session Ended. Please Run the code again ===