



Chapters

1	2	3	4
5	6	7	8
9	10	11	12
13			

Chapter 11

- [Theory \(4\)](#)
- [Programming exercises \(3/3\)](#)
- [Quizzes \(3/3\)](#)
- [Open exercises](#)

Course

- [Table of contents](#)
- [Extra materials](#)
- [Bulletin board](#)
- [My corner](#)

Communication

- [Forum](#)
- [Conference](#)

Tutoring

- [Ask a tutor](#)

Phone directory - removing a person's data

Feedback [Ask a tutor](#)

Please use this option only if you find any issues or mistakes in the content. If you have comments or questions regarding the subject matter, then please use [Ask a tutor](#).

 Page: **3/3**

- [\(1\) Phone directory - adding data done](#)
- [\(2\) Phone directory - printing data done](#)
- [\(3\) Phone directory - removing a person's data done](#)

[Hide](#)

Write a program for removing a specific person's data from the directory. The program asks for a first name followed by a last name. The program checks if the person's data can be found. If the person is in the directory, the data is removed. Otherwise print an error message.

Hint:

You should carefully design the program before writing any actual code. The use of structures may make the task easier. It is most obvious that when you save an updated list, the existing data in the file phonendir.txt is overwritten.

You can compare name information using the C language function strcmp() in the library file string.h. You can find information about using the function in the C manual.

Example output:

```
Enter first name:Tim
Enter last name:Franklin
Data removed from the directory.
```

The verification of program output does not account for whitespace characters like "\n", "\t" and " "

program.c

```
1 #include <stdio.h>
2 #include <string.h>
3
4 int main()
5 {
6     /*1. Read file into an array*/
7     char string[5][100];
8     char line[100];
9     int i;
10    i = 0;
11    int size;
12    FILE *read_file;
13    char phonendir[] = "phonendir.txt";
14
15    read_file = fopen(phonendir, "r");
16
17    if ((read_file = fopen(phonendir, "r")) == NULL)
18    {
19        printf("Error opening file.");
20        return 0;
21    }
22    else
23    {
24
25        while (fgets(line, sizeof line, read_file) != NULL)
26        {
27            strcpy(string[i], line);
28            i++;
29        }
30        size = i;
31    }

```

Position: Ln 105 Ch 2

Full screen (Esc to exit)

Save

Run