

## 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 - adding 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: **1/3**

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

Send

Hide

In this chapter, we create independent programs that are closely associated with each other. All the programs are components of a phone directory program. The intention is that you can continue your programming exercises by programming a phone directory. The exercises in this chapter can easily be converted into functions for a larger entity.

The name and number data for the phone directory are contained in the file "phonedir.txt" which has the following format:

```
4
Firstname Lastname 050-3500980
John Doe 041-3478924
Brian Smith 040-3980982
Brita Smith 05-4567393
```

The first line of the file indicates the number of names added to the directory as an integer. The first name, last name and telephone number are processed in text format. The data items are separated by spaces in the file, and the length of an individual item (such as a first name) may be no more than 20 characters. Each line includes the information for one person only. The directory may contain a maximum of 50 people. Your first task is to write a program for adding a new telephone number to the directory. Upon execution, the program must first ask for a first name, after which it prompts for a last name and a telephone number. When data is saved, the value on the first line of the file must increase by one. Note that in order to simplify the exercise, the data to be entered may not include a newline character, for example. Each personal data item is stored on its own line. (Remember the newline character.)

*Example output:*

```
Enter first name:Thomas
Enter last name:Jefferson
Enter telephone number:05-6243370
Successfully saved the data.
```

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

- program.c

```
1 #include <stdio.h>
2 int main()
3 {
4     int i;
5     FILE *amend_file;
6     char phonedir[] = "phonedir.txt";
7
8     FILE *write_file;
9     char phonedir2[] = "phonedir.txt";
10
11    struct person
12    {
13        char firstname[15];
14        char lastname[25];
15        char number[25];
16    };
17
18    struct person person_list;
19
20    printf("Enter first name:");
21    scanf("%s", &person_list.firstname[0]);
22    printf("Enter last name:");
23    scanf("%s", &person_list.lastname[0]);
24    printf("Enter telephone number:");
25    scanf("%s", &person_list.number[0]);
26
27    amend_file = fopen(phonedir, "a");
28
29    if ((amend_file = fopen(phonedir, "a")) == NULL)
30    {
31        printf("Error opening file\n");
32    }
33    else
34    {
35        fprintf(amend_file, "%s %s %s\n", person_list.firstname, person_list.lastname, person_list.number);
36    }
37
38    fclose(amend_file);
39}
```

Position: Ln 67 Ch 2

Full screen (Esc to exit)

Reset

Save Run