

Phone number conversion

Table of Contents

- 1. Problem
 - 1.1. 2 Tips for the programming assignment
- 2. Solution

1 Problem

- Write a program that prompts the user to enter a telephone number in the form (xxx) xxx-xxxx, and then displays the number in the form xxx.xxx.xxxx.
- Example input/output of the first program, phone1.c:

```
Enter phone number [(xxx) xxx-xxxx]: (870) 456-7890
You entered: 870.456.7890
```

- Write another program that asks for the input format in the form xxx\xxx\xxxx, and then displays the number in the form (xxx)xxx-xxx.
- Example input/output of the second program, phone2.c:

```
Enter phone number [xxx\xxx\xxxx]: 870\456\7890
You entered: (870) 456-7890
```

- Submit one Emacs Org-mode file phone.org with both programs in it as code blocks that can be **tangled** as phone1.c and phone2.c, resp.
- The header information of your Org-mode file should look like this:

```
##+TITLE: Phone number conversion
##+AUTHOR: [your name]
##+HONOR: pledged
```

- Tip: some characters, like \ are protected because they are part of the file PATH. If you want to use them, you have to "escape" them with an extra \, like the newline character \n. So to print (or to scan) the character \, you use \\.

1.1 2 Tips for the programming assignment

1. (02/25/2022) If you want to test and run your file **inside** Emacs, which saves a lot of time, just add this to the header arguments (after the c):

```
:cmdline < input
```

where input is a file that contains the phone number as requested for input, (xxx) xxx-xxxx or xxx\xxx\xxxx. Now C-c C-c will run and execute the program. Alternatively, you have to tangle the code block, compile and run it by hand on the command line.

2. (02/27/2022) the formatting in scanf() needs to match the input pattern. E.g. if the input is given as

```
foo==+//bar
```

(with `foo` and `bar` declared as `int` variables) then only the following command will pick the input up properly:

```
scanf("%d==+//%d", &foo, &bar);
```

irrespective of the output, which can be formatted in any way we like. `scanf()` only looks for two integers separated by all that junk between the numbers.

Example: the file `inputfile` contains only: `444==+//555`. We want to only print out 444 and 555.

```
int a,b;
scanf("%d==+//%d", &a, &b);
printf("%d %d", a, b);
```

2 Solution

- To run the first program, I took the input number from `input1`:

```
(501) 422 4725
```

- Program 1

```
int area, prefix, suffix;
printf("Enter phone number [(xxx) xxx-xxxx]:\n");
scanf("(%d) %d-%d", &area, &prefix, &suffix);
printf("You entered: %d.%d.%d\n", area, prefix, suffix);
```

- To run the first program, I took the input number from `input1`:

```
501\422\4275
```

- Program 2: in both functions, the `\` character has to be escaped as `\\`.

```
int area, prefix, suffix;
printf("Enter phone number [xxx\\xxx\\xxxx]:\n");
scanf("%d\\%d\\%d", &area, &prefix, &suffix);
printf("You entered: (%d) %d-%d\n", area, prefix, suffix);
```

Author: Marcus Birkenkrahe

Created: 2022-03-17 Thu 17:03

[Validate](#)