Programming in C. Converter for Metric Measures of Length

Oleksandr Zaitsev
Inria Lille - Nord Europe
Polytech Lille

To work on the exercise, you must fork the repository `https://github.com/olekscode/MetricConverterExercises` and clone it to your personal computer. The code is located in the *src/* folder. You can compile and run it with the following commands:

```
gcc main.c converter.c -o converter
./converter
```

In this exercise, you will write functions to convert different measures of length in the metric system. There are 10 millimetres in centimetre, 100 centimetres in a metre, 1000 metres in a kilometre.

Task 1

Add three #define pragmas into *converter.h* and use them to store the constant values given above:

1. MILLIMETRES_IN_CENTIMETRE
2. CENTIMETRES_IN_METRE
3. METRES_IN_KILOMETRE

Commit and push your changes.

Task 2

Create a file *converter.c*, include file *converter.h* and perform the following steps for each function from the list below:

1. Uncomment this function in *converter.h*
2. Implement the function in *converter.c*
3. Commit your changes (do not push them!)

After all three functions are implemented, uncomment the line **#include** "tests/tests_task2.h" in *main.c* and three tests:

```
test_centimetres_to_millimetres();
test_metres_to_centimetres();
test_kilometres_to_metres();
```

Compile and run your code, make sure that tests pass. Then push your changes (do not push the executable, you can add it to *.gitignore*).

Here is the list of functions that you need to implement for this task:

- Implement function **double** centimetres_to_millimetres(**double** centimetres) using pragma MILLIMETRES_IN_CENTIMETRE.
- Implement function **double** centimetres_to_millimetres(**double** centimetres) using pragma CENTIMETRES_IN_METRE.
- Implement function **double** kilometres_to_metres(**double** kilometres) using pragma METRES_IN_KILOMETRE.

Task 3

Follow the same steps as for Task 2 for each of the functions (remember that you must commit every function separately, but push them together):

- Implement function **double** millimetres_to_centimetres(**double** millimetres) using pragma MILLIMETRES_IN_CENTIMETRE.
- Implement function **double** centimetres_to_metres(**double** centimetres) using pragma CENTIMETRES_IN_METRE.
- Implement function **double** metres_to_kilometres(**double** metres) using pragma METRES_IN_KILOMETRE.

Uncomment the line **#include** "tests/tests_task3.h" in *main.c* then uncomment and run the following tests:

```
test_millimetres_to_centimetres();
test_centimetres_to_metres();
test_metres_to_kilometres();
```

If all the tests pass (should be 6 tests by now), push your changes.

Task 4

Follow the same steps as for Task 2 for each of the functions (remember that you must commit every function separately, but push them together):

- Implement function **double** metres_to_millimetres(**double** metres) using functions metres_to_centimetres and centimetres_to_millimetres.

- Implement function **double** millimetres_to_metres(**double** millimetres) using functions millimetres_to_centimetres and centimetres_to_metres.
- Implement function **double** kilometres_to_millimetres(**double** kilometres) using functions kilometres_to_metres and metres_to_millimetres.
- Implement function **double** millimetres_to_kilometres(**double** millimetres) using functions millimetres_to_metres and metres_to_kilometres.
- Implement function **double** kilometres_to_centimetres(**double** kilometres) using functions kilometres_to_metres and metres_to_centimetres.
- Implement function **double** centimetres_to_kilometres(**double** centimetres) using functions centimetres_to_metres and metres_to_kilometres.

Uncomment the line **#include** "tests/tests_task4.h" in *main.c* then uncomment and run the following tests:

```
test_metres_to_millimetres();
test_millimetres_to_metres();
test_kilometres_to_millimetres();
test_millimetres_to_kilometres();
test_kilometres_to_centimetres();
test_centimetres_to_kilometres();
```

If all the tests pass (should be 12 tests by now), push your changes.