Programming assignment 3.

Basic programming for solving calculation problems

Purpose of the exercise

This exercise will help you do the following:

1. Develop skills in C programming with different data types, operators and expressions.
2. Develop familiarity with formatted console output.

Overview

Temperature is measured with a thermometer, and thermometers are calibrated in various temperature scales. The most common scales are:

- the Celsius scale (formerly called *centigrade*, denoted °C),
- the Fahrenheit scale (denoted °F),
- the Kelvin scale (denoted K).

To perform conversion of temperatures written in Fahrenheit to Celsius:

- Subtract 32 from the *Fahrenheit* temperature
- Multiply this number by 5
- Divide the result by 9

To convert Celsius to Kelvin, simply add 273.15 to the Celsius temperature.

In this assignment, you are to write a function `temperature_convertor()` that takes in a temperature in Fahrenheit and displays it in Celsius and Kelvin on the screen. Following shows the output related to a input `0` in Fahrenheit:

```
Farenheit      Celsius        Kelvin
---------------------------------------------
0              -17.78         255.37
```

In this assignment the only data types you should use are `int`, `double`, and `void`.

While formatting output for displaying the numbers use left justification, set the width of a column of text to 15 characters, and use 2 decimal places for floating-point numbers.

Task

1. Download the assignment source code files:

   Go to the course page in *Moodle* - DigiPen (Singapore) online learning management system, click the link **Assignment 3 files**, download a zipped archive that contains the source file and extract them into a Microsoft Windows directory *c:\sandbox\* (adjust the path as suitable for your system). The included files are:

   - *q.c*
   - *qdriver.c*
   - *out.txt*
2. Using the Microsoft Windows command prompt navigate to the *sandbox* folder. Then type `wsl` to open Linux bash shell in the same current directory.

3. From Linux open *q.c* in Microsoft Visual Studio Code:

   ```
   code q.c
   ```

4. Begin editing by formatting a file-level documentation at the top of the source file *q.c*. Make modifications to the template by replacing `@todo` with your information. Providing a file-level documentation is mandatory for any submissions in this course.
5. Devise a complete algorithm; go through all the steps of the algorithm formulation process to make sure it is clear, well defined, and correct. Then implement the function `temperature_convertor()` inside the file *q.c*:

   ```
   #include <stdio.h>  // printf

   void temperature_convertor(int fahrenheit)
   {
       // TODO: Complete the code
   }
   ```

   Do not forget to save the source file after editing!

   You will **not** submit *qdriver.c*; any changes you make to it will not be seen during grading.
6. Add function-level documentation right before the function definition. Providing function-level documentation is mandatory for all functions for all submissions in this course.
7. Compile and link the edited source file *q.c* and the given file *qdriver.c*:

   ```
   gcc -Wall -Wextra -Werror -Wconversion -Wstrict-prototypes -pedantic-errors -std=c11 qdriver.c q.c -o main
   ```

   If the compilation succeeded, you should see the executable file appear in the directory. If you made any mistakes in the code, the compiler will print out a list of errors and warnings that need to be fixed; edit your code and try again.
8. Run the executable *main* and redirect the output to a file *myout.txt*:

   ```
   ./main > myout.txt
   ```

9. The given file *out.txt* contains the expected output. Your output, which was redirected into a file *myout.txt*, must **exactly** match the contents of *out.txt*. Use the `diff` command in the Linux bash shell to compare your implementation's actual output with the expected output provided to you, like this:

   ```
   diff --strip-trailing-cr -y --suppress-common-lines myout.txt out.txt
   ```

Submitting the deliverables

You have to upload a complete file *q.c* to *Moodle* - DigiPen (Singapore) online learning management system, where the file will be automatically evaluated.