CO222: Programming Methodology: Project 1 - Specification

Deadline: March 4th 2016 @ 11.55PM
Marks: 10%

Command Line Converter (an extension to lab4)

In this project, you are expected to build a **complete** command line conversion tool. The converter should take inputs as command line arguments and produce outputs to the standard output (**stdout**). The tool should take three arguments as shown below (assuming that the tool is going to be called **clc**):

```
./clc -<input format> -<output format> <input>
```

Input/output format can be one of the followings:

Format	Description
B	A 32-bit or 64-bit representation
H	An 8-digit or 16-digit hexadecimal representation
I	Integer (32-bit)
F	Single precision floating point number
D	Double precision floating point number

You are **not allowed to use** any library functions except **fprintf()** from **stdio**.

Following aspects should be considered when you develop your solution:
1. The program should be modularized as much as possible using proper programmer defined functions.
2. Your program should have the minimum possible amount of code.
3. Identifiers (names of variables and functions) should have proper names.

One of the following error (and warning) messages should be produced to the standard error (**stderr**). When there is an error, the program should stop running and return a 1. When there is a warning, the program should continue running after outputting the warning. Please note that the error/warning messages should be printed __identical__ to what you see below as we will be marking your answers automatically (our auto-marker is not intelligent enough to ignore spaces, understand capital/simple letters, etc.).

1. Wrong number of arguments
   ERROR: The number of arguments is wrong.
   Usage: ./clc -<input format> -<output format> <input>

2. Wrong input argument
   ERROR: The input argument is wrong.
   Possible input arguments are -B, -H, -I, -F and -D.

3. Wrong output argument
   ERROR: The output argument is wrong.
   Possible output arguments are -B, -H, -I, -F and -D.

4. Wrong input size
   ERROR: The input size is wrong.

5. Mistake in the input format at location x from left to right (x is a number, which starts from 0)
   ERROR: Input format error at location x.

6. If there are conversions that are not possible, you should print
   ERROR: This conversion is not possible.

7. Conversions that will lose/reduce precision
   WARNING: There is a possibility for a precision loss.

If you find any ambiguities in the error/warning messages you are encouraged to raise them in the project forum, where we will give our interpretation. We will not take responsibility for your wrong interpretation of error/warning messages.

Submit your answer (without compile error or warning) in a single file with the filename E13yyyclc.c, where yyy is your registration number.

46 69 6e 61 6c 6c 79 2c 20 48 61 70 70 79 20 43 6f 64 69 6e 67 21 ☺