104818 – COMPUTER ORGANIZATION AND PROGRAMMING – SPRING 2025 ..... ASSIGNMENT C WARM UP

ACADEMIC INTEGRITY. You are reminded that you should understand and comply with basic standards of academic integrity –in particular, in matters related to misrepresentation by deception or by other fraudulent means of submitted work. Lack of academic integrity can result in serious consequences – the minimum of which is a grade of 0 (zero) on the assignment. The following illustrates only three forms of academic dishonesty: (1) plagiarism, e.g., the submission of work that is not one's own or for which other credit has been obtained; (2) improper collaboration; (3) copying or using unauthorized material.

SUBMISSION. You must submit your solution using C files with *.c* extension, C header files with *.h* extension, and a *makefile* file. Your solution must also include any provided files, even if you didn't modified them. As for any additional files you want to submit, the only accepted formats are: plain text file (*.txt*); markdown files (*.md*); and PDF files (*.pdf*).

Context

This Assignment is meant to warm up with programming (specifically in C). We will see simple problems and solve them on a high level language. Later on the course will start programming in Assembler, and we will revisit many of these problems, so it is important to have a more abstract grasp on how to approach them so later on we will be able to solve them in a low level language.

Accompanying this guide, I'll provide code including C (header and implementation) files, and a *makefile* which we will use to compile the code. Organization of provided code is as follow:

makefile Containing incomplete compilation instructions to be completed.

standard.h Header file containing function profiles, types, structure names, and constants. This file also contains documentation for all declarations.

boolean.c C file containing implementation of all boolean functions and related structures.

math.c C file containing implementation of all math functions.

strings C file containing implementation of all string functions.

main.c Containing code to test the implemented functions of *standard.h.*

Exercises

1. Complete the *Makefile* with the following tasks:
   (a) A compilation task that generates an executable called *assignment1*, this task requires *boolean.o*, *math.o*, and *strings.o*.
   (b) A compilation task to generate each *boolean.o*, *math.o*, and *strings.o* (a task for each one).
2. Make the necessary modifications needed to have *expression*s involving *XOR*. These modifications may include changes in *standard.h* and *boolean.c*. If you modify *standard.h* make sure to add the necessary comments/documentation.
3. In *boolean.c* complete the implementation of function *eval* for *expression*s involving *AND*, *OR*, and *XOR*.
4. In *math.c* implement the *maximum* function.

5. In *math.c* implement the *is_prime* function.
6. In *math.c* implement the *fibonnaci* function.
7. In *strings.c* implement the *lenght* function.
8. In *strings.c* implement the *equals* function.
9. In *strings.c* implement the *last_index_of* function.
10. In *strings.c* implement the *to_lower_case* function.
11. In *strings.c* implement the *to_upper_case* function.
12. In *strings.c* implement the *substring* function.
13. In *strings.c* implement a private function (use the *static* keyword) which will switch a range of characters by a certain constant value, the profile will be:
    static char shift_in_range(const char c, unsigned int range_start, unsigned int range_end, const int shift)

    For example, we can define the range of lower case letters (97 - 122), and a shift of -32 to shift any lower case letter to it's upper case counterpart while keeping any other symbol unchanged.

    Now re-implement *to_lower_case* and *to_upper_case* functions to use this new *shift_in_range* function.