## Programming Exercise B - HTML Tag Detection

### Assignment Objectives

* Practice using a C program to read characters from a file
* Practice using command-line redirection of files
* Practice identifying HTML tags in a sequence of characters
* Practice using character arrays
* Practice writing character information to a file

### Assignment Summary

In this assignment, you will create a C program that prints a list of the HTML tags in an HTML file, followed by the total number of tags found. The program may assume that the input file is a properly-formatted HTML file and that an HTML tag begins and ends on the same line. Your finished program should compile, link and run with no error messages. Submit your completed C source code file on the Blackboard course website. Do not submit any other files.

### Software Requirements

The program shall read through the HMTL file by means of standard input. This means that the user of the program shall use file redirection on the command line in order for the program to read an existing HTML file. The program shall not prompt the user for any information.

Before printing any tags, the program shall first print "**\*\*\* START OF HTML TAGS \*\*\***", followed by a blank line. The program shall then search sequentially through the file for HTML tags.

An HTML tag starts out with a left (or opening) angle bracket. The bracket is then followed by a tag name and zero or more attributes, which all may be in uppercase letters, lowercase letters, or some mix of both. A tag ends with a right (or closing) angle bracket.

When the program detects a left angle bracket, it has found the start of a tag. When the program then finds a right angle bracket, it has found the end of the tag. By definition, tags may not be nested within each other. When the program finds an HTML tag, it shall print the tag and its contents (including the angle brackets) all on one line, with one tag per line.

After the final tag is printed, the program shall print a blank line followed by "**\*\*\* TOTAL NUMBER OF TAGS FOUND: xx \*\*\***", where xx represents the actual count. The program shall then exit normally. See the sample-run.txt file for more details on the required format of the output data.

### Design and Implementation Constraints

* Name your source code file ProgramB.c
* Use a modular approach in your program consisting of two or more user-defined functions (including the main function)
* Keep your implementation simple and easy to understand
* Declare all variables at the top of a block
* Use variables names and function names that contain well-known acronyms or words commonly found in a dictionary
* You may declare global constants in your program, but do not declare any global variables
* Declare no user-defined classes
* Use no dynamic memory allocation (e.g., new, malloc(), calloc(), etc.)
* Do not use or create a string tokenizer. Also, do not read the HTML file line by line. Instead, read through the file character by character
* Use only the C I/O functions shown and demonstrated in class
* Add no other functionality to the program other than what is requested in this assignment
* Use the following as the basic layout of your source code file:

Include file names

User-defined types and constants

Function prototypes

Definition of the main function

Definitions of all other user-defined functions