Ankara University
Computer Engineering
COM2067 LAB 4

In this lab, you are asked to check the accuracy of an HTML code received from the user by using stacks. If there is no error in the HTML code, "correct" is printed. If there is an error, first "error" and then the incorrect tag with a space are printed on the screen.

An HTML document generally looks like this (there are exceptions):

<HTML> <!-- This tag is compulsory for any HTML document. -->
<HEAD>
<!-- The Head tag is used to create a title of web page, CSS syntax for a web page-->
</HEAD>
<BODY>
<!--The Body tag is used to display the content on a web page-->
</BODY>
</HTML>

HTML tags can be classified in many different ways. One classification includes opening and closing tags. These tags contain the name of the HTML element. For example,

<html> </html> : creates the HTML source document.
<title> </title>: names the source document.
<h1> </h1>: prints the largest title.

Submission:

Name your source file as <StudentID>.c. For example, if your ID is 22290777, then you will submit 22290777.c file.

Testing:

We provide a sample input/output text file pairs for you to test your codes at Ubuntu. Please carefully review the sample input and output files given to you for the correct output format.

We recommend you to use input redirection mechanism of your operating system to test your programs. For example, if your executable is called as Lab4, redirect the input.txt file to standard input using < operator and redirect your outputs to a file using > operator such as:

> ./Lab4 < input.txt > output.txt

This kind of execution enables your programs to read inputs from a file without writing any file related functions. In other words, scanf reads data from the redirected files instead of the std. input in this way (e.g. keyboard).

Automatically compare your own output with the expected output by using the diff myOutput1.txt output1.txt command. If a warning as shown below does not appear on the screen after executing this command, this means that your program is working correctly. If you see a warning in the command system after executing the command, this indicates that there is a problem with your output.

Test your program for different inputs that you will create yourself. Please note that the input files given to you and the input files used during the evaluation may differ from each other.