Project 1: scanners

1. Introduction.

The purpose of this assignment is to write a simple "translator" from HTML to text. The main goal is to make yourself familiar with the scanner generator *flex.*

Your task is to use the scanner generator *lex* or *flex* to write a program that "translates" HTML to text. Since the main goal of this assignment is to learn how to use *flex*, we won't pay too much attention to all the peculiarities of HTML in a manner that should be followed had we been working on a commercial project.

Please find the documentation for *lex/flex* here:
`http://flex.sourceforge.net/manual/`

1.1. Description of the program.

Your program should take data from *stdin* on input, remove all HTML tags (including comments, see description below), recognize and handle a certain set of "special characters" and write output to *stdout.*

1.2. HTML tags.

By defaulft HTML uses a broad array of tags. Since we are mostly interested in learning *flex* here, we shall simplify our task a little and accept the following definition of a tag:

A tag is any sequence of characters of the form $< S >$*, where* S *is a sequence of printable characters that doesn't start with a white space and doesn't include a* $>$ *character.*

By printable characters we mean characters as specified by the *isprint()* function in C, and by white spaces all characters as specified by *isspace().* They correspond to classes of characters defined in *flex* by *[:print:]* and *[:blank:]*, respectively.

For example, each of the following sequences of characters is a tag:

- $< b >$
- $< br >$
- $< ahref = "http://www.math.us.edu.pl/\ pgladki/" >$
- $< /b >$
- < !– this is an HTML comment, and has the structure of a tag as described above – >

1.3. Special characters.

HTML defines certain "special characters" that your program should recognize and change according to the following scheme:

- &*amp*; change to &
- &*lt*; change to <
- &*gt*; change to >
- &*quot*; change to "

For the full list of special characters that your program ought to handle, see:
`http://www.utexas.edu/learn/html/spchar.html`

2. Running your program.

Your executable file should be called *myhtml2txt* and should read from *stdin* and write output to *stdout.* In other words, a translation from *foo.html* to *foo.txt* shoul be executed as follows:

myhtml2txt < *foo.html* > *foo.txt*