published on Novermber 3, 2015

Programming Languages: Assignment #2

Due Novermber 15, 2015 at 11pm

Description

LaTeX is a word processor and a markup language that is used to format a source Latex text to an output file (e.g. PDF). In this assignment, you will implement simple automata using the flex tool. The input (stdin) to your flex will contain a Latex-like program and your output should be an HTML translation of the Latex source. The translation from a Latex source to an HTML output will be directed by a set of text processing rules to be followed.

Notice, you are not prerequisite to have any prior knowledge in neither Latex nor HTML.

- An example for a Latex-like source:

```
\title{My document}

\section{Introduction}
This is the introduction section.

\section{My First Section}\label{S-First}
This is a word in \emph{italic}. \\This is a word in \textbf{bold}.
This is a formula: $a = x + 5$.
See the following list of (ordered) items:
\begin{enumerate}
\item first item
\item second item
\end{enumerate}

\section{Another Section}
For more information about how to style a text in Latex see \ref{S-First}.
```

Definitions

Whitespace (WS):

`' ' (space), \t or (tab)`

End of line (EOL):

`\n or \r\n`

Commands

Your input will be a Latex program source that contains a mix of content (the text to display) and commands (the directives for how the content should be display). The table below lists Latex commands and their corresponding expected HTML output (given by example).

Command	Latex	HTML	Notes
Title	`\title{My Title}`	`<h1>My Title</h1>`	
Section	`\section{My Section}`	`<h2>1 My Section</h2>`	For each subsequent section, the section-number is incremented by 1. There is a single space after the number.

Lemma	`\begin{lemma}`
	`my lemma text`
	`\end{lemma}`
		`<b>Lemma 1. </b>`
		`<i>my lemma text</i>`
			For each subsequent lemma, the lemma-number is incremented by 1. The pattern for the first part: `"Lemma <NUM>. "`.

Unordered List	`\begin{itemize}`
	`\item first text`
	`\item second text`
	`\end{itemize}`
		`<ul>`
		`<li>first text</li>`
		`<li>second text</li>`
		`</ul>`
			There must be at least one whitespace or EOL after `\item`.

Ordered List	`\begin{enumerate}`
	`\item first text`
	`\item second text`
	`\end{enumerate}`
		`<ol>`
		`<li>first text</li>`
		`<li>second text</li>`
		`</ol>`
			There must be at least one whitespace or EOL after `\item`.

Italic text	`\emph{italic text}`	`<i>italic text</i>`	
Bold text	`\textbf{Bold text}`	`<b>bold text</b>`	
Break line	`aa\\bb\\cc`	`aa</br>bb</br>cc`	

Formulas

A formula is defined inside `$...$` and is translated to `<code>...</code>`.
For example: `$a = x + 5$` is translated to `<code>a = x + 5</code>`.
The text inside a formula is processed by the following substitution rules.

- `\alpha` → `&alpha;`

- `\beta` → `&beta;`
- `\eps` → `&epsilon;`
- `\le` → `&le;`
- `\ge` → `&ge;`

For example: `$\alpha + \beta = 6$` is translated to `<code>&alpha; + &beta; = 6</code>`.

Comments

A comment starts with `%` and ends with an EOL (or EOF) (similar to C "\\").
Example: `% this is my comment`.
Comments should be ignored and not to be printed.

Labeling

In order to allow a text to refer to the number of a section or a lemma, a section or lemma can be optionally decorated with a label. To add a label to a section or to a lemma we use the `\label` command.
For example:

```
\section{About Latex}\label{Section-About-Latex}
\begin{lemma}\label{Lemma-My-Lemma}
lemma text
\end{lemma}
```

To refer a section or lemma number inside a text, we use `\ref`.
For example, the following text refers to the lemma labeled "Lemma-My-Lemma".

```
We use here Lemma \ref{Lemma-My-Lemma} to prove that...
```

Notice that all labels must be distinct.

Content Text

A content text is the text we actually want to display. It may appear outside of commands, inside of commands or inside formulas. For example, in the following code

```
\section{AAA}
BBB \emph{CCC}
\begin{lemma}
DDD
\end{lemma}
$YYY$
```

"AAA", "CCC" and "DDD" are inside of command, while "BBB" is outside of command and "YYY" is inside of a formula.

In a content text, the character '\' is allowed only if it is part of command, otherwise it is an error. The characters '`{`' and '`}`' if appeared in a text (i.e. not part of command) are ignored.
The content inside commands is further restricted and only `a-z, A-Z, 0-9`, `-` (hyphen), WS and EOL are allowed. Content inside formulas has the same limitation as for content inside command, however it also

allows `+,*,/,^,=,!,(,)`. Also, the character '\' is allowed only if it is part of the substitution rules for formula (otherwise it is an error).

Notice that the given limitation of the allowed text content inside commands or formula, prevent nested commands (recall that recursive structures are better to be handled by bison, not flex).

Output

Every output is wrapped with:

<**html**>
<**body**>
...
</**body**>
</**html**>

In case of an error, the program stops and prints `error`, so the closing `body` and `html` will not appear in such case (see also "Handling Errors"). The content should be printed as is (except for the special rules described above).
WS and EOL: End of lines and whitespaces at the beginning or at the end of lines are not printed (i.e. when EOL occurred, every WS before it and every EOL/WS after it are not printed). Between two non-WS sequences, a sequence of whitespaces (i.e. 1 or more whitespaces) are reduce to a single space. All the heading/trailing whitespaces/EOLs in a content inside curly braces of command including "`\label`" and "`\ref`" are eliminated. Same applied to the content after "`item`".
Examples (`"\n"` and `"\t"` represent breakline and tab and **not** the strings `"\n"` and `"\t"`):

- "`aa\nbb`" → "`aa\nbb`".
- "`aa\t\t\n  \n    bb`" → "`aa\nbb`".
- "`aa \t\t   bb`" → "`aa bb`".
- "`\emph{ \nmy text  } `" → "`<i>my text</i>`".
- "`\item \t\txx \t\n yy\n\t\n`" → "`<li>xx\nyy</li>`".

Specific output formatting: in the output of the Lemma command, after "`<number>.`" there is a single space and there is no space between `<b>` and `<i>`. In lists there is `\n` *after* `<ul>`, `<ol>` and `<\li>` and `\t` before `<li>`.

Handling Errors

In case of an error, everything until the first error is outputted as defined. When an error occurred for the first time, the word `error` is printed and the program must immediately exit. A command and its content are outputted only if the command and its content are all written correctly (i.e. if there is an error nothing of the command gets printed). If there is illegal token in a text outside of a command (i.e. `\cat` appears in the middle of the text), all the text until the illegal token is printed.

Example:

input:

```
Line 1
Line 2
\begin{lemma}
aaaa
\end{lema}
```

Output:

```
Line 1
Line 2
error
```

Notes:

1. Command name must not contain whitespaces or EOL (e.g. "`\ti tle{`" or "`\ title{xxx}`" are errors). Also "`{lemma}`"/ "`{itemize}`"/"`{enumerate}`" must not contain WS or EOL. However WS and EOL may appear elsewhere in command (e.g. "`\title  {   xx yy zz}`" is OK).
2. A content inside ordered/unordered list must preceded by `\item` and separated by at least one whitespace or EOL. A list might not have items (empty list).
3. If EOF is reached before command ended, it's an error.
4. `$aaa$xxx$yyy$` is not an error, but `$aaa$xxxyyy$` is (assuming the third `$` is the last one).

Examples

Input:

```
\title{  My document
    }

\section{Introduction}
This is the introduction section.



\section{My First Section}\label{S-First}
This is a word in \emph{italic}. \\This is a word in \textbf{bold}.
This is a formula: $  \alpha \le x + 5 + \eps$.
See the following list of (ordered) items:
\begin{enumerate}
\item first     item
\item second item
\end{enumerate}

\section{Another Section}
For more information about how to style a text in Latex see Section \ref{S-First}.
```

Output:

```
<html>
<body>
<h1>My document</h1>
<h2>1 Introduction</h2>
This is the introduction section.
<h2>2 My First Section</h2>
This is a word in <i>italic</i>. </br>This is a word in <b>bold</b>.
This is a formula: <code>&alpha; &le; x + 5 + &epsilon;</code>.
See the following list of (ordered) items:
<ol>
    <li>first item</li>
    <li>second item</li>
</ol>
<h2>3 Another Section</h2>
For more information about how to style a text in Latex see Section 2.
</body>
</html>
```

Input:

```
\title{  My document
    }

\section{Introduction}
This is the introduction section.




\section{My #First Section}\label{S-First}
This is a word in \emph{italic}. \\This is a word in \textbf{bold}.
This is a formula: $  \alpha \le x + 5 + \eps$.
See the following list of (ordered) items:
\begin{enumerate}
\item first     item
\item second item
\end{enumerate}

\section{Another Section}
For more information about how to style a text in Latex see Section \ref{S-First}.
```

Output:

```
<html>
<body>
<h1>My document</h1>
<h2>1 Introduction</h2>
This is the introduction section.
error
```

Submission Notes

- Assignment name: Ex2
- The assignment should run on the department computer: u2
- Submission via 'submit' system (submit.cs.biu.ac.il).
- You must supply (at least) the following files: makefile, ex2.lex
- The running executable must bear the name: a.out

General Notes

1. You must use Flexs states functionality.
2. Your code will probably be checked manually. In order to achieve a high grade, you should make sure the action (of the pattern) is as short as possible. Long or complex code (e.g., string manipulation) implies you did not use the full power of Flex (which will reflect in a lower grade).

3. You can get some preliminary help at:
   (a) Flex Examples: [2] → Example `#6`: http://comsci.liu.edu/ murali/lexyacc/Main.htm
   (b) Compiler Design Using Flex And Yacc, ch. 3.7, p.30
   (c) `http://dinosaur.compilertools.net/flex/flex_11.html#SEC11`
4. You can look at the make file at the end of the PPT `#2`
5. You may consult others (friends, internet, etc...) but copying or communal work is forbidden. Please do not plagiaries. We make all effort to catch such cases.
6. Some characters are different between Linux and Windows OS (e.g., EOL). Thus, when you write your own test files, dont forget to 'dos2unix' when migrating them from your Win platform to Planets Linux.
7. Remember that the submit system accepts ZIP files
8. At the head of your Ex2.lex file, add a remark with your: name, ID, group, user-name
9. Remember that your work will be checked (after the deadline) against tougher test files. Therefore, make sure to write your own (malicious) test files
10. Please don't send me mails on Shabbat