CSCI 3901 Assignment 1

Due date: 11:59pm Friday, January 18, 2019 in Brightspace

Problem 1

Goal

Get practice in decomposing a problem, creating a design for a program, and implementing and testing a program.

Background

One of the products that opened the web to the general public is a content management system (CMS). A CMS allowed non-technical people to post formatted information on the web without knowing the hyptertext markup language (HTML) used by browsers.

We're going back in time to early CMSs and the first version of HTML. Early CMSs asked writers to put simple notation around their text to denote formatting. For example, one might use underscrores before and after text to indicate italics, such as _italicized_. The CMS would then translate that notation to the appropriate HTML coding; version 1.0 of HTML would have encoded our example as <i>italicized</i>. The <i> notation is called an HTML tag and almost always appear in bracketing pairs.

All tags are identified by starting with the < symbol and ending with the > symbol. With bracketing tags, the starting tag begins with < and ends with > while the ending tag begins with </ and ends with >, just like <i> and </i> above. A stand-alone tag begins with < and ends with />. For example, an empty line might be represented by the "break" tag of <br />.

HTML allows us to nest tags within tags. For example, if we wanted bolded (<b> tag) italicized (<i> tag) text then we would say <b><i>some text</i></b>. Notice that the tags must be closed in the opposite order that they were opened to nest properly. There are some specific instances where this nesting is absolutely necessary:

- All the document must be nested inside a <html> tag
- All metadata about the document, like the title, must be nested inside a <head> tag
- All the document information must be nested inside a <body> tag
- All text of one paragraph must be nested inside a <p> tag
- All items of an unordered list must be nested inside a <ul> tag

Problem

Write a program that translates the contents of a text file into an HTML 1.0 web page and prints that HTML to the screen.

The text file will have the following special annotation characters:

- If the first non-blank line starts with “title: “ (no quotes) then the text that follows in that line should be reported in a <title> tag within the <head> tag of the HTML
- A blank line in the text represents a new paragraph. Paragraphs are captured with the <p> tag.
- Text between underscores (_) should be italicized using the <i> tag.
- Text between asterisks (*) should be bolded using the <b> tag.
- Text between percentage signs (%) should be underlined using the <u> tag.
- The appearance of an ! (alone or at the start of a word) indicates that the next word should be bolded using the <b> tag. The ! at the end of a word should be treated as punctuation.
- Text that starts with a minus sign (-) as the first non-space character is a list item that should be enclosed in a list <li> tag.

Inputs

Your program will read a filename from the keyboard and assess the content of the file.

For testing simplicity, prompt for a filename with the text “filename? “ on a line of its own.

The contents of the file will be regular text. It will require the translations mentioned above and summarized below:

Text annotation	Meaning	Tag
First text line starts “title: “	Title of the document	<title>
Blank line	New paragraph	<p>
some text	Italicize “some text”	<i>
more text	Bold “more text”	<b>
! even more text	Bold next word “even”	<b>
%last text%	Underline “last text”	<u>
- List info	List item “list info”	<li>

Outputs

Output is to the screen. It is a properly nested HTML 1.0 file that is ready for a web browser to read. Each line of input text should be printed as its own single line of output text.

Assumptions

You may assume that

- The input text does not contain any HTML tags or any special symbols that need to be translated for HTML.
- None of the special symbols used to annotate the text formatting appear in the text except the exclamation point as proper punctuation.
- No line in the file is more than 80 characters.
- List items are a single line
- List will not be inside paragraphs: they will have a blank line separating them from the surrounding text, unless starting or ending the body of the file
- There will be no formatting annotations inside the title line

Constraints

- You may not use an HTML or XML library.
- Write your solution in Java. You are allowed to use data structures from the Java Collection Framework.
- If in doubt for testing, I will be running your program on bluenose.cs.dal.ca. Correct operation of your program shouldn't rely on any packages that aren't available on that system.
- Have the following tags / tag blocks on their own lines for both opening and closing tags: html, head, body, p, ul
- Have the "title" and "li" tags entirely on one line with their text

Marking scheme

- Documentation (internal and external) – 3 marks
- Program organization, clarity, modularity, style – 5 marks
- Ability to process file contents – 5 marks
- Ability to translate the file to HTML 1.0 for "normal" uses of the annotations – 10 marks
- Ability to handle convoluted annotation cases by sloppy text users – 5 marks
- Estimated effort it would take to add in more block tags – 2 marks
  - 0 – we can't re-use anything without careful review
  - 1 – there is an obvious pattern that can be used, but new code must be created
  - 2 – adding the block tags amounts to updating some data. No code to change

The majority of the functional testing will be done with an automated script, so stick to the output format.

Sample interaction

Input file contents

```
title: a first programming exercise
first line
second *bold me* line
_third_ italicized line

Try some %underline%

- One
- Two

Something !bold to come
```

Output

```
<html>
<head>
<title>a first programming exercise</title>
</head>
<body>
<p>
first line
second <b>bold me</b> line
<i>third</i> italicized line
</p>
<p>
Try some <u>underline</u>
</p>
<ul>
<li>One</li>
<li>Two</li>
</ul>
<p>
Something <b>bold</b> to come
</p>
</body>
</html>
```

Test cases

Files:

- Blank line (empty) file name
- File name that doesn’t exist
- File with no lines in it
- File with 1 line in it
- File with 2 lines in it
- File with many lines (so a middle line)

Metadata:

- File with no title metadata
- File with title metadata on the first line
- File with blank lines before the title metadata
- File with text after the title metadata line
- File with a blank line after the title metadata line
- File with multiple blank lines after the title metadata line

Tags:

- File with a metadata line and no body information
- File with a single paragraph
- File with 2 paragraphs
- File with several paragraphs
- File with several blank lines between two paragraphs
- File with no paragraph but a list
- File with both paragraphs and lists, paragraphs first and last in the file
- File with both paragraphs and lists, lists first and last in the file
- For each of the annotations _, *, !, and %
  - Annotation is the first character in the line
  - Annotation is the first non-space character in the line
  - Annotation is the last character in the line
  - Annotation is the last non-space character in the line
  - Annotation is in the middle of the line
  - Annotation is being applied to a single character
  - Annotation is being applied to a single word
  - There is no space between the annotation and the text to which it is applied
  - There are spaces between the annotation and the text to which it is applied
  - The text to which it is applied is on the next line
  - There is a single annotation in an input line
  - There are multiple different annotations in an input line
  - There are multiple instances of the same annotation in an input line
  - Only one annotation is being applied to a bit of text
  - Multiple annotations are being applied to a bit of text
  - Annotations at the start of the text for list items
  - Annotations at the end of the text for list items
  - Annotations in the middle of text for list items
- For each of the annotations _, *, and %
  - Annotation is being applied to 2 or more words
  - Annotation open and close on the same text line
  - Annotation open and close are on different text lines
  - Opened annotation closed before the end of the paragraph
  - Opened annotation not closed before the end of the paragraph
- Annotations closed in a correctly-nested order
- Annotations closed in an incorrectly-nested order
- Several different formatting characters in a row
- Several of the same formatting characters in a row
- Annotations all closed before the end of the input text file
- Annotations not all closed by the end of the input text file
- The ! appears as punctuation at the end of some word
- The ! annotation precedes another annotation item
- The ! annotation comes right after another annotation item

- Two ! annotations appear one after the other
- ! and * are applied to the same text