9

**Free-Response Question**

Free-response questions related to this lab will typically focus on text processing. One possible question is shown below.

When Web developers want to convert plain text documents to HTML, they need to make changes to the text to insert tags and replace certain characters. For this question, you will write part of the class to convert text to HTML. You will write methods to replace underscores ("\_") with tags indicating the text should be in italics.

The class definition for this problem is given below:

public class TextFormatter

{

private String line; // The line to format

public TextFormatter (String lineToFormat)

{ line = lineToFormat; }

/\*\*

* Finds the first single instance of str in line,
* starting at the position start
* @param str the string of length 1 to find.
* Guaranteed to be length 1.
* @param start the position to start searching.
* Guaranteed to be in the string line.
* @return the index of the first single instance of
* str if the string is found or -1 if it is not found.

\*/

private int findString (String str, int start)

{ /\* To be implemented in part a) \*/ }

/\*\*

* Count the number of times single instances of str appear in
* the line.
* @param str the string to find.
* Guaranteed to be length 1.
* @return the number of times the string appears in the line

\*/

private int countStrings (String str)

{ /\* To be implemented in part b) \*/ }

/\*\*

* Replace all single instances of underscores in the line given by
* line with italics tags. There must be an even number of underscores

10

* in the line, and they will be replaced by <I>, </I>, alternating.
* @param original a string of length 1 to replace
* @param replacement the string (of any length) use as a replacement
* @return the line with single instances of underscores replaced with
* <I> tags or the original line if there are not an even number of
* underscores.

\*/

public String convertItalics ()

{ /\* To be implemented in part c) \*/ }

}

a) Write the method findString. This method will take a string of length 1 to find in the line,

at a given starting point. It will return the location of the goal string. This differs from the indexOf method of the String class because it requires the string be just a single instance of the string, so if the string appears two or more times consecutively, it will not return any of those values. If there is no single instance, the method should return -1. Consider the following examples, where line has the value "aabaccb".

|  |  |  |
| --- | --- | --- |
| Call | Result | Explanation |
| findString("a", | 0) | 3 | The first single occurrence of "a" is in position 3.The "a"s in position 0 and 1 are not returned because they are not single instances. |
| findString("b", | 4) | 6 | The first single occurrence of "b" at or after position4 is in position 6. |
| findString("c", | 0) | -1 | There is no single instance of "c" in the line. |

1. Write the method countStrings. This method will take a string of length 1 to find in the line. It will return the number of times single instances of the goal string appear in the string.
2. Write the method convertItalics. If the line has an even number of single underscores, a line with those underscores converted to alternating <I> and </I> tags should be returned. The first underscore will be converted to <I>, the second to </I>, the third to <I>, and so on. If the line does not have an even number of <I> tags, the original line should be returned. Consider

the following examples.

|  |  |
| --- | --- |
| line | value returned by convertItalics |
| This is \_very\_ good. | This is <I>very</I> good. |
| \_This\_ is \_very\_ \_good\_. | <I>This</I> is <I>very</I> <I>good</I>. |
| This is \_very good. | This is \_very good. |
| This is very good. | This is very good. |