Homework #3

Question 1 (3 pt.)

Consider the following fragment of a C program:

```
float limitedSquare(float x)
{
    /* Returns x-squared, but never more than 100 */
    return (x <= -10.0 || x >= 10.0) ? 100 : x * x ;
}
```

a) Apply a manual lexical analysis pass on the given code, by splitting it into lexems. For each identified lexeme, write the associated token in the format $\langle TokenName, AttributeValue \rangle$, where the attribute value should only be specified when strictly necessary.

b) For each of the token types identified above, give a regular definition that identifies them. If you need to use a symbol of the alphabet overlapping a regular expression operator (such as `*`, `?`, `(`, `)`, or `+`), use double quotes (`"*"`, `"?"`, `"("`, `")"`, or `"+"`). If you need to use the double quote itself as a symbol of the alphabet, you can escape it by preceding it with a backslash: `\"`.

Question 2 (4 pt.)

Consider the following HTML code:

```
Hello, how are <b>you</b>?
<p>
  This is an image: <img src="pic.jpg">
  <br>
  And this is a hyperlink: <a href="site.html">Click here</a>
</p>
```

a) Split it into lexemes and specify the associated tokens. There are significantly different approaches to scan HTML code, all of which are valid options.

b) Again, associate regular definitions to each different token type identified above.

Question 3 (3 pt.)

Write regular definitions for the following languages:

a) All strings of lowercase letters that contain the five vowels in order, and each vowel occurs only once.

b) All strings of lowercase letters from a to g in which the letters are in ascending lexicographic order.

c) All strings of digits where digit 0 appears at most once, not including the empty string.