CMSC 350 Project 1
Tokenizing the input

1. Tokenizing the input using regular expressions

For tokenizing the input, I recommend using regular expressions (**regex**).
By using regular expressions we may define String patterns that can be further used (i) for identifying patterns in texts and (i) for other text manipulations. To use regular expressions in Java you have to include the classes java.regex.Pattern and java.regex.Matcher defined in the java.regex package.

For example, for tokenizing Project 1 input we may use the following regular expression:

```
String regexPattern = "\\d+|(\\+|-|\\*|=|:)";
```

If the string to be tokenized is either of the following:

```
String prefix = "* 2 + 2 -+ 12 9 2"; // using spaces between tokens
String prefix = "*2+2-+12 9 2";      // spaces only between operands
```

then to get the tokens that should be considered by the Project 1 may be obtained as follows:

```
Pattern pattern = Pattern.compile("\\d+|(\\+|-|\\*|=|:)");
Matcher matcher = pattern.matcher(prefix);
while(matcher.find()) {
    String s = matcher.group();
    // ... process the token (string) s according to your algorithm
    System.out.println(s);
}
```

Below are the tokens generated as a result of executing the above code:

```
*
2
+
2
-
+
12
9
2
```

This way you may get each token of the input string and process it as required by the pseudocode.

Below please find three good tutorials about regular expressions in Java.

https://www.baeldung.com/regular-expressions-java
http://tutorials.jenkov.com/java-regex/index.html

https://beginnersbook.com/2014/08/java-regex-tutorial/
https://www.javatpoint.com/java-regex

2. Tokenizing the input using the StringTokenizer

An alternative way of tokenizing the input is to use the StringTokenizer class (although its usage is no more recommended by Oracle).

For example, if the prefix input string is stored in the location *prefix* of type String

```
String prefix = "* 2 + 2 -+ 12 9 2";
```

then you may use:

```
StringTokenizer tokenizer = new StringTokenizer(prefix, "+-*/() ",
true);
```

Using this StringTokenizer in a loop:

```
while(tokenizer.hasMoreTokens() {
   System.out.println(tokenzier.nextToken());
}
```

you will get the following tokens:

*
<space>
2
<space>
+
<space>
2
<space>
-
+
<space>
12
<space>
9
<space>
2

If the input postfix expression is using spaces only between numbers, i.e.

```
String postfix = "*2+2-+12 9 2";
```

then the resulted tokens will be:

*
2
+
2
-
+
12
<space>
9
<space>
2