A computer program can be used to draw banners consisting of squares and triangles. The program makes use of the following five instructions:

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>Draw a large square</td>
</tr>
<tr>
<td>s</td>
<td>Draw a small square</td>
</tr>
<tr>
<td>T</td>
<td>Draw a large triangle</td>
</tr>
<tr>
<td>t</td>
<td>Draw a small triangle</td>
</tr>
<tr>
<td>N[I]</td>
<td>Repeat the instructions, I, exactly N times</td>
</tr>
</tbody>
</table>

For example, the program `s 2[T t] S` draws the following banner:

Questions:

1. Given the program `t 4[s] T 3[t S]`, draw the corresponding banner.

2. Create two different programs that will draw the following banner:

3. Given the program `2[2[s S] t T]`, draw the missing shapes in the following banner:

4. Given the incomplete program `?[2[?] t ?[s T ?]]`, complete the missing instructions in order to draw the following banner:
5. Suppose you want to draw the following banner:

![Banner Image]

You create the program \[2[S \ T \ t] \ 2[T \ S \ s]\] which incorrectly draws this banner:

![Incorrect Banner Image]

What are the mistakes in your program?

6. A new instruction named \texttt{if} is now available to you. The instruction \((a:b/c)\) means that \textit{if} the previous shape drawn was \(a\), then the next shape drawn is \(b\). \textit{If} the previous shape drawn was \textbf{not} \(a\), then the next shape drawn is \(c\).

For example, the program \(s \ (s:S/t) \ (t:T/s)\) draws the following banner:

![Example Banner Image]

For each program in parts (a) to (f), decide whether or not it will draw the following banner:

(a) \[2[T \ (t:T/t)]\]  
(b) \(T \ (T:t/s) \ (t:T/S)\)  
(c) \(T \ 2[(t:T/t)]\)  
(d) \(t \ (t:T/s) \ (s:S/t)\)  
(e) \(T \ (T:t/S) \ (S:s/T)\)  
(f) \(3[(T:t/T)]\)

7. Try creating your own new instructions. Perhaps add new shapes, or new capabilities such as chaining shapes vertically. Swap programs with a friend or family member and try to draw each other’s banners.

More Info:  
Check out the CEMC at Home webpage on Wednesday, April 1 for the solutions to these questions. This task exercises your computational thinking muscles! For more information on how this task relates to computer science, check out Chain on the 2016 Beaver Computing Challenge.