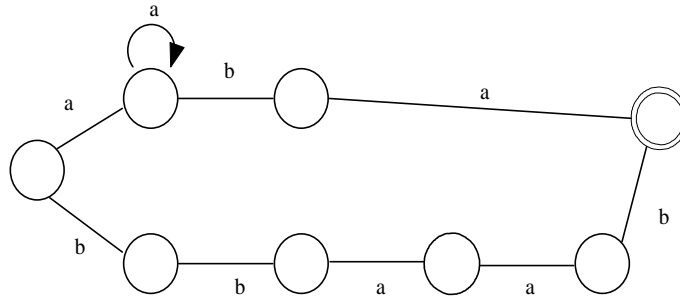


All-Star Contest  
FSA

**PROBLEM:** Given a valid Regular Expression, draw the FSA. The FSA must stay on the screen long enough to be graded. If it scrolls off the screen prior to grading, it will be graded as incorrect. The string (aa\*ba)U(bbaab) produces the FSA drawing below and which your program will print as the diagram below the drawing:



<b>O</b>	<b>aO</b>	<b>a*</b>	<b>-bO</b>	<b>a</b>	
	<			>	<b>@</b>
	<b>bO</b>	<b>-bO</b>	<b>-aO</b>	<b>-aO</b>	<b>b</b>

Start with upper case O. End with @. Union splits with a < and ends with a >. The two segments that form the < and > symbols are considered edges. Note that the start and end transition rules for union must align as shown above. Transition rules (the letters a and b) are written above and below the < and >. Nodes are noted by an upper case O. The other transition edges are dashes followed by the rule followed by a node. Note, the transition rule going into the end state is not followed by a node (O). A circular node is a transition rule and an asterisk. Terms must align vertically and be left justified except as shown above for >.

**INPUT:** There will be 10 lines of input. Each line will be a string representing a Regular Expression. There will be at most 1 union operator per string.

**OUTPUT:** For each line of input print its FSA according to the rules above. Do not simplify the input expressions.

SAMPLE INPUT

SAMPLE OUTPUT

1. abab
2. ab\*aba
3. (a a\*ba)U(bbaab)

1. O -aO -bO -aO -b @
2. O -aO b\* -aO -bO -a @

3.						
	aO	a*	-bO		a	
O	<				>	@
	bO	-bO	-aO	-aO	b	