

```

1  /*****
2  | ***** For Loops of Varying Size *****
3  | *****/
4
5  Consider that we would like to print the following pattern for a given value of
6  SIZE, where 3 < SIZE < 9:
7
8  SIZE 3:          SIZE 6:
9  //////////////\  //////////////\
10 //*****\  //*****\
11 *****  //*****\
12          //*****\
13          *****
14
15 Step 1: Determine the different patterns associated with the problem.
16 This will indicate the number of for loops that we need to have in our solution.
17 Ask yourself: "How many different symbols are incorporated into the problem?"
18
19 When creating patterns, it's a good idea to separate the problem line by line.
20 When doing this, create an outer for loop to manage the number of lines that we
21 would like to print out.
22
23 //      +-----+
24 //      | This part of the for loop tells  |
25 //      | the program to print three lines. |
26 //      | This number will change depending |
27 //      | on the number of lines we would  |
28 //      | like to print out, which will    |
29 //      | vary depending on the SIZE that  |
30 //      | we are printing out. For now, we |
31 //      | can leave this at three, however, |
32 //      | we will need to change it later  |
33 //      | so that it prints the correct    |
34 //      | number of lines based on the SIZE |
35 //      | of the figure we have chosen.    |
36 //      +-----+
37 //      |
38 //      |
39 //      \|/
40 //      V
41 for(int line = 1; line <= 3; line++)
42 {
43     // OTHER CODE WILL GO HERE
44
45     // Including this println on the end will
46     // allow the program to move to the next line.
47     System.out.println();
48 }
49
50 We will then need a for loop for each symbol, "/", "*" and "\",
51 to control the number of times each character is printed per line.
52 These for loops will go inside of the line number for loop, like so:
53
54 // Line
55 for(int line = 1; line <= 3; line++)
56 {
57     // Print each "/"
58     for()
59     {
60         System.out.println("/");
61     }
62
63     // Print each "*"
64     for()
65     {
66         System.out.println("*");
67     }

```

```

68
69 // Print each "\"
70 for()
71 {
72     System.out.println("\\");
73 }
74 }
75
76 Step 2: Construct a table for at least one of the SIZES.
77 Let's use SIZE 3. SIZE 6 is included for reference as well.
78
79 +-----+ +-----+
80 |          SIZE = 3          | |          SIZE = 6          |
81 |=====| +=====+
82 | line  /      *      \  | | line  /      *      \  |
83 |-----| |-----|
84 | 1      8      0      8  | | 1      24     0      24  |
85 | 2      4      8      4  | | 2      16      8      16  |
86 | 3      0     16      0  | | 3      12     16     12  |
87 +-----+ | 4      8      24     8  |
88 |          | | 5      4      32     4  |
89 |          | | 6      0     40     0  |
90 +-----+ +-----+
91
92 Step 3: Determine the mathematical expressions
93 Each character in the table is going to need it's own expression.
94 This will tell the computer how many times it needs to print that
95 character for the line number that it's currently at.
96
97 For this reason, we need an expression dependent upon the line number.
98
99 Let's start with the slashes.
100 We need to find an equation that satisfies the following requirements:
101
102 When line is 1, the # of slashes = 8
103 When line is 2, the # of slashes = 4
104 When line is 3, the # of slashes = 0
105 +-----+
106 | line  Expression  /  |
107 |-----|
108 | 1      line * _____ = 8  |
109 | 2      line * _____ = 4  |
110 | 3      line * _____ = 0  |
111 +-----+
112 A possible solution: 12 - (4*line)
113
114 Check if the expression satisfies the conditions:
115 +-----+
116 | line  Expression          /  |
117 |-----+
118 | 1      12 - (4*line) = 8  | -> 12 - (4*1) = 8
119 | 2      12 - (4*line) = 4  | -> 12 - (4*2) = 4
120 | 3      12 - (4*line) = 0  | -> 12 - (4*3) = 0
121 +-----+
122
123 Therefore, the expression, 12 - (4*line), gives us the
124 number of slashes that we need to print out for each line.
125
126 Set up the for loop accordingly:
127
128 // Print each "/"
129 for(int f = 1; f <= 12 - (4*line); f++)
130 {
131     System.out.print("/");
132 }
133
134 Now for the stars.
135 We need to find an equation that satisfies the following requirements:

```

```

136
137 When line is 1, the # of stars = 0
138 When line is 2, the # of stars = 8
139 When line is 3, the # of stars = 16
140 +-----+
141 | line      Expression      * |
142 |-----|
143 | 1      line * _____ = 0 |
144 | 2      line * _____ = 8 |
145 | 3      line * _____ = 16 |
146 +-----+
147 A possible solution: 8 * (line-1)
148
149 Check if the expression satisfies the conditions:
150 +-----+
151 | line      Expression      * |
152 |-----|
153 | 1      8 * (line-1) = 8 | -> 8 * (1-1) = 0
154 | 2      8 * (line-1) = 4 | -> 8 * (2-1) = 8
155 | 3      8 * (line-1) = 0 | -> 8 * (3-1) = 16
156 +-----+
157
158 Therefore, the expression, 12 - (4*line), gives us the
159 number of "/" that we need to print out for each line.
160
161 Set up the for loop accordingly:
162
163 // Print each "/"
164 // +-----+
165 // | This part of the for loop will      |
166 // | change depending on the SIZE that |
167 // | we are working on. In this case, |
168 // | Starting with 12 works when SIZE |
169 // | is equal to 3; however, this will |
170 // | will not work for any other SIZE, |
171 // | and so we need to change this.    |
172 // +-----+
173 //          |
174 //          |
175 //          \|/
176 //          V
177 for(int f = 1; f <= 12 - (4*line); f++)
178 {
179     System.out.print("/");
180 }
181
182
183 Incorporating SIZE:
184 Since the SIZE will end up changing the number of slashes that we will be
185 starting with, we need to include it into our formulas above for slashes.
186
187 Consider the tables again. Notice how we start with 8 slashes in SIZE 3, while
188 we start with 16 in SIZE 6:
189 +-----+          +-----+
190 | SIZE = 3 |          | SIZE = 6 |
191 |=====+          |=====+
192 | line  / |          | line  / |
193 |-----|          |-----|
194 | 1      8 |          | 1      20 |
195 | 2      4 |          | 2      16 |
196 | 3      0 |          | 3      12 |
197 +-----+          | 4      8  |
198                   | 5      4  |
199                   | 6      0  |
200                   +-----+
201
202 We need to change our initial starting number in the slashes equation:
203

```

```
204 When SIZE is 3: We start with 12 slashes
205 When SIZE is 6: We start with 20 slashes
206 +-----+
207 |  SIZE      Expression  /  |
208 |-----|
209 |  3          SIZE * _____ = 12 |
210 |  6          SIZE * _____ = 20 |
211 +-----+
212 Possible Solution: SIZE*4
213
214 This will change the slashes equation to as follows:
215 Originally:      12      - (4*line) <-- This only works for SIZE = 3
216 New Equation: (SIZE*4) - (4*line) <-- This works for all SIZES
217 \*****/
218 public class StarFigure
219 {
220     // SIZE is a constant int
221     public static final int SIZE = 6;
222
223     public static void main(String[] args)
224     {
225         // Print out each line
226         for(int line = 1; line <= SIZE; line++)
227         {
228             // Within each line...
229             // Print out each forward slash
230             for(int f = 1; f <= (4*SIZE)-(4*line); f++)
231             {
232                 System.out.print("/");
233             }
234
235             // Print out each star
236             for(int s = 1; s <= 8*(line-1); s++)
237             {
238                 System.out.print("*");
239             }
240
241             // Print out each backslash
242             for(int b = 1; b <= (4*SIZE)-(4*line); b++)
243             {
244                 System.out.print("\\");
245             }
246
247             // Drop down to the next line
248             System.out.println();
249         }
250     }
251 }
```