

2. This question involves methods that distribute text across lines of an electronic sign. The electronic sign and the text to be displayed on it are represented by the `Sign` class. You will write the complete `Sign` class, which contains a constructor and two methods.

The `Sign` class constructor has two parameters. The first parameter is a `String` that contains the message to be displayed on the sign. The second parameter is an `int` that contains the *width* of each line of the sign. The width is the positive maximum number of characters that can be displayed on a single line of the sign.

A sign contains as many lines as are necessary to display the entire message. The message is split among the lines of the sign without regard to spaces or punctuation. Only the last line of the sign may contain fewer characters than the width indicated by the constructor parameter.

The following are examples of a message displayed on signs of different widths. Assume that in each example, the sign is declared with the width specified in the first column of the table and with the message "Everything on sale, please come in", which contains 34 characters.

Width of the Sign	Sign Display
15	<div>Everything on s ale, please com e in</div>
17	<div>Everything on sal e, please come in</div>
40	<div>Everything on sale, please come in</div>

In addition to the constructor, the `Sign` class contains two methods.

The `numberOfLines` method returns an `int` representing the number of lines needed to display the text on the sign. In the previous examples, `numberOfLines` would return 3, 2, and 1, respectively, for the sign widths shown in the table.

The `getLines` method returns a `String` containing the message broken into lines separated by semicolons (;) or returns `null` if the message is the empty string. The constructor parameter that contains the message to be displayed will not include any semicolons. As an example, in the first row of the preceding table, `getLines` would return "Everything on s;ale, please com;e in". No semicolon should appear at the end of the `String` returned by `getLines`.

GO ON TO THE NEXT PAGE.

The following table contains a sample code execution sequence and the corresponding results. The code execution sequence appears in a class other than `Sign`.

Statement	Method Call Return Value (blank if none)	Explanation
<code>String str;</code>		
<code>int x;</code>		
<code>Sign sign1 = new Sign("ABC222DE", 3);</code>		The message for <code>sign1</code> contains 8 characters, and the sign has lines of width 3.
<code>x = sign1.numberOfLines();</code>	3	The sign needs three lines to display the 8-character message on a sign with lines of width 3.
<code>str = sign1.getLines();</code>	"ABC;222;DE"	Semicolons separate the text displayed on the first, second, and third lines of the sign.
<code>str = sign1.getLines();</code>	"ABC;222;DE"	Successive calls to <code>getLines</code> return the same value.
<code>Sign sign2 = new Sign("ABCD", 10);</code>		The message for <code>sign2</code> contains 4 characters, and the sign has lines of width 10.
<code>x = sign2.numberOfLines();</code>	1	The sign needs one line to display the 4-character message on a sign with lines of width 10.
<code>str = sign2.getLines();</code>	"ABCD"	No semicolon appears, since the text to be displayed fits on the first line of the sign.
<code>Sign sign3 = new Sign("ABCDEF", 6);</code>		The message for <code>sign3</code> contains 6 characters, and the sign has lines of width 6.
<code>x = sign3.numberOfLines();</code>	1	The sign needs one line to display the 6-character message on a sign with lines of width 6.
<code>str = sign3.getLines();</code>	"ABCDEF"	No semicolon appears, since the text to be displayed fits on the first line of the sign.
<code>Sign sign4 = new Sign("", 4);</code>		The message for <code>sign4</code> is an empty string.
<code>x = sign4.numberOfLines();</code>	0	There is no text to display.
<code>str = sign4.getLines();</code>	null	There is no text to display.
<code>Sign sign5 = new Sign("AB_CD_EF", 2);</code>		The message for <code>sign5</code> contains 8 characters, and the sign has lines of width 2.
<code>x = sign5.numberOfLines();</code>	4	The sign needs four lines to display the 8-character message on a sign with lines of width 2.
<code>str = sign5.getLines();</code>	"AB;_C;D_;EF"	Semicolons separate the text displayed on the four lines of the sign.

GO ON TO THE NEXT PAGE.

Write the complete `Sign` class. Your implementation must meet all specifications and conform to the examples shown in the preceding table.