

Problem 3 – NikolAngular JS

AngularJS is famous JavaScript library by Google. However, our master MVC ninja Niki decided to rewrite it for himself. He needs some junior developers so he hired you (poor him :D). Prove Niki you are the best from the best and help him build the next big thing – NikolAngular JS.

NikolAngular JS is a modern view engine adding new HTML tags which work like logic commands. Non-technically said you can use some of the additional tags to help you generate the final HTML output. This is called view and all additional tags start with "nk-".

Here is the allowed syntax for NikolAngular JS:

- **Model properties** – you can pass an object (data model) to the view and render its properties' values in between the HTML. Each property is a key-value pair. Keys must be valid JavaScript variable names. Nested objects are not allowed. Values can be:
 - **String** – examples: "Telerik Academy", "Razor View Engine", "ASP.NET MVC"
 - **Boolean** – "true" or "false"
 - **Array** – comma separated values. Examples: "JavaScript,C Sharp,C++", "1,2,3,4"

Syntax for rendering a value: `<nk-model>key</nk-model>`

Example – model is:

- **title**-Telerik Academy
- **subtitle**-Free Software Trainings

View	HTML Result
<pre><div> <nk-model>title</nk-model> <nk-model>subtitle</nk-model> </div></pre>	<pre><div> Telerik Academy Free Software Trainings </div></pre>

- **Define template** – you can define templates with content. Each template has unique name and can contain only plain HTML elements. All templates are defined at the beginning of the NikolAngular view. Templates defining is not part of the final result.

Syntax is: `<nk-template name="template-name">content</nk-template>`

Example – defining a template with name "menu" and unordered list as content

```
<nk-template name="menu">
  <ul>
    <li>Home</li>
    <li>About us</li>
  </ul>
</nk-template>
```

- **Render template** – previously defined templates can be rendered multiple times on the HTML by specifying their name. Non existing templates will result in error.

Syntax is: `<nk-template render="name" />`

Example – rendering the "menu" section inside our view

View	HTML Result
<pre><div> <h1>Telerik Academy</h1> <div> <nk-template render="menu" /> </div> </div></pre>	<pre><div> <h1>Telerik Academy</h1> <div> Home About us </div> </div></pre>

- **Conditional statement** – conditional statement depending on Boolean property in the model object. If the condition is "true", its content is rendered, otherwise it is not. The condition cannot be Boolean expressions or directly the "true" or "false" literals. It must be a key in the model.

Syntax is: `<nk-if condition="conditionValue">content</nk-if>`

Example – model is

- **masterNinja**:true
- **notNinja**:false
- **myDegree**:the best ninja

View	HTML Result
<pre> <h1>Telerik Academy</h1> <nk-if condition="masterNinja"> <h2>I am <nk-model>myDegree</nk-model></h2> </nk-if> <p>Very good developer</p> <nk-if condition="notNinja"> <h2>Not <nk-model>myDegree</nk-model></h2> </nk-if> </pre>	<pre> <h1>Telerik Academy</h1> <h2>I am the best ninja</h2> <p>Very good developer</p> </pre>

- **Loop** – loops over an array value from the model object and prints the loop content for each item in the collection.

Syntax is: `<nk-repeat for="item in collection">content</nk-repeat>`

Example – model is

- **trainers:** Ivo, Niki, Dodo
- **degree:** ninja

View	HTML Result
<pre> <h1>Telerik Academy</h1> <nk-repeat for="trainer in trainers"> <nk-model>trainer</nk-model> <nk-model>degree</nk-model> </nk-repeat> </pre>	<pre> <h1>Telerik Academy</h1> Ivo ninja Niki ninja Dodo ninja </pre>

- **Escaping** – when it is needed to escape the commands on the final result, they can be escaped by surrounding them with "{{" and "}}".

Example

View	HTML Result
<pre><div> <h1> {{<nk-model>title</nk-model>}} </h1> </div></pre>	<pre><div> <h1> <nk-model>title</nk-model> </h1> </div></pre>

Additional notes

- Using **eval()**, **external libraries** (or parts of them) and **regular expressions** is **not allowed**!
- All opening command tags will be on one line.
- All closing command tags will have only white space (" ") on the line they appear.
- There will not be any nested conditional and loop commands.
- All commands are case sensitive.
- There are no tabs in the view, only white space.
- All tests will contain only valid and formatted NikolAngular JS views.
- Some tests try only some of the commands so partial solutions may give you points. Keep trying! 😊

Input

The input data should be read from the console.

On the first line you will receive **N** – the number of key-value pairs in the model object.

On the next **N** number of lines you will receive each key-value pair from the model object. The key is separated from the value with the dash ("-") sign. All array values will be separated with the semi-colon (";") sign. There will never be dashes or semi-colons in the values themselves.

On the next line you will receive **M** - the number of lines in the view.

On the next **M** lines you will receive the view which you have to parse.

The input data will always be valid and in the format described. There is no need to check it explicitly.

Output

The output should be printed on the console.

On the first **M** lines of the output you should print the parsed HTML.

Sample solution code (in JavaScript)

```
function solve(params) {
  var s = params[0];
  var result = '';
  // Your solution here
  console.log(result);
}
```

Constraints

- **N** will be between **0** and **20**, inclusive.
- **M** will be between **1** and **200**, inclusive.
- All lines from the view will contain no more than **200** symbols.
- All views will be valid and as described.
- Allowed working time for your program: **0.10 seconds**.
- Allowed memory: **16 MB**.

Example

Input

```
6
title-Telerik Academy
showSubtitle-true
subTitle-Free training
showMarks-false
marks-3;4;5;6
students-Ivan;Gosho;Pesho
42
<nk-template name="menu">
  <ul id="menu">
    <li>Home</li>
    <li>About us</li>
  </ul>
</nk-template>
<nk-template name="footer">
  <footer>
    Copyright Telerik Academy 2014
  </footer>
</nk-template>
<!DOCTYPE html>
<html>
<head>
  <title>Telerik Academy</title>
</head>
<body>
  <nk-template render="menu" />

  <h1><nk-model>title</nk-model></h1>
  <nk-if condition="showSubtitle">
    <h2><nk-model>subTitle</nk-model></h2>
    <div>{{<nk-model>subTitle</nk-model>}} ;)</div>
  </nk-if>

  <ul>
    <nk-repeat for="student in students">
      <li>
        <nk-model>student</nk-model>
      </li>
      <li>Multiline <nk-model>title</nk-model></li>
    </nk-repeat>
  </ul>
  <nk-if condition="showMarks">
    <div>
      <nk-model>marks</nk-model>
    </div>
  </nk-if>
</body>
</html>
```

```
</nk-if>
```

```
<nk-template render="footer" />
```

```
</body>
```

```
</html>
```

Output

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Telerik Academy</title>
```

```
</head>
```

```
<body>
```

```
<ul id="menu">
```

```
<li>Home</li>
```

```
<li>About us</li>
```

```
</ul>
```

```
<h1>Telerik Academy</h1>
```

```
<h2>Free training</h2>
```

```
<div><nk-model>subTitle</nk-model> ;)</div>
```

```
<ul>
```

```
<li>
```

```
Ivan
```

```
</li>
```

```
<li>Multiline Telerik Academy</li>
```

```
<li>
```

```
Gosho
```

```
</li>
```

```
<li>Multiline Telerik Academy</li>
```

```
<li>
```

```
Pesho
```

```
</li>
```

```
<li>Multiline Telerik Academy</li>
```

```
</ul>
```

```
<footer>
```

```
Copyright Telerik Academy 2014
```

```
</footer>
```

```
</body>
```

```
</html>
```