Presentation layer of Spoj problems

This document is a tutorial for creating properly formated description for problems on a spoj.com platform. We will present you a proper HTML template that fits well to the service layout.
Currently, as a Problem Setter you are able to modify a template that is presented initially during adding a new problem. However, such modifications are undesirable (except for very rare situations).
Please note that we've made an assumption that as a Problem Setter you are familiar with basics of HTML.

HTML template

The following HTML code is a backbone for every problem on Spoj.

```
<p>Problem text...</p>

<h3>Input</h3>
<p>Input description...</p>

<h3>Output</h3>
<p>Output description...</p>

<h3>Example</h3>
<pre><strong>Input:</strong>
etc.

<strong>Output:</strong>
etc.
</pre>
```

It's easy to separate four sections of the problem description:

General description

This following part of a template covers the general description of a problem.

```
<p>Problem text...</p>
```

General description is a part of problem description where the problem setter should define the problem. There are two possible approaches for a problem definition:

- formal - by the precise mathematical definition,
- fabular - by the story that is a model of the formal definition.

If the general description has a reference to the term (definition, lemma, theorem, etc.) that cannot be considered as a mathematical folklore then this term should be a part of the problem description as well. For example, you don't need to define a triangle or a prime

number. However, if you would like to refer to Bernoulie numbers, then you should provide the definition.

Input specification

This following part of a template is dedicated to describe a format of data that will be delivered to user's program.

```
<h3>Input</h3>
<p>Input description...</p>
```

You should provide a correct and very precise information. Remember that your **problem solvers are going to rely on this information**.

Please remember that you should **deliver constraints to each variable** defined in this section.

Output specification

This following part of a template is dedicated to describe a format of data should be printed by the user's program.

```
<h3>Output</h3>
<p>Output description...</p>
```

Once again you should provide precise information. Your **problem solvers are going to obey your rules**, i.e. they will print the output in accordance with the specification.

Example

The last part of presented template is dedicated for an example.

```
<h3>Example</h3>
<pre><strong>Input:</strong>
etc.

<strong>Output:</strong>
etc.
</pre>
```

The example should be delivered in a form of input and output data. The input data should be consistent with input specification. Similarly, the output data should be consistent with output specification.

Example

The following problem is an example of well formated problem on a Spoj platform:
http://www.spoj.com/EXAMPLES/problems/EX_REFERENCE/

We present this problem description in the raw HTML form:

General description

```
<p>A <em>square number</em> is an integer number that can be represented in a form of a square of the other integer number. For example, number 25 is a square number because 25 = 5<sup>2</sup>. In a contrary, number 18 is not a square number because there is no integer number <em>k</em> for which <em>k</em><sup>2</sup> = 18.</p>
<p>You are asked to find the closest square number for a given integer number. The distance between two numbers <em>n</em> and <em>m</em> is defined by the absolute value of their difference, i.e. dist(<em>n</em>,<em>m</em>) = |<em>n</em> - <em>m</em>|.</p>
```

Input specification

```
<h3>Input</h3>
<p>The first line of the input consist of a single integer number <em>t</em> which determines the number of tests.</p>
<p>In each of next <em>t</em> lines there is a single integer number <em>n</em>.</p>
<h4>Constraints</h4>
<ul>
<li>0 &amp;lt; t &amp;le; 1000</li>
<li>0 &amp;lt; n &amp;le; 50&amp;thinsp;000&amp;thinsp;000</li>
</ul>
```

Output specification

```
<h3>Output</h3>
<p>For each number <em>n</em> print its closest square number. Separate your answers with a new line character.</p>
```

Example

```
<h3>Example</h3>
<pre><strong>Input:</strong>
4
11
23
99
101

<strong>Output:</strong>
```

```
9
25
100
100
</pre>
```

Typographical requirements

For the best problem solver experience we suggest to obey the following rules:

- use ≥ (≥ in HTML) and ≤ (≤ in HTML) instead of >= and <=,
- all mathematical variables should be written in italics (or in HTML),
- all mathematical operators and function (e.g. min, sin, det) should be written in regular font,
- use superscript for powers (e.g. a^b instead of *a^b*),
- use subscript for indices (e.g. a_n instead of *a_n*),
- use bold or emphasized font for important notes (or in HTML),
- make big numbers readable (e.g. replace 1000000000 with 10^9 or 1 000 000 000;   is a thin space in HTML).

Avoid the following formatting styles:

- colors (background or font),
- overline and underline,
- headers that are not related to the template.

Optional content

The following components may be used to improve your problem description:

1. **Image for general description**
2. **Image for example**
3. **Subsection with constraints**

   There are two approaches for providing constraints for your data:

   - after every introduced variable, e.g.:

     ```
     <h3>Input</h3>
     <p>The first line of the input consist of a single integer number
     <em>t</em> (0 < t &le; 1000) which determines the number of tests.</p>
     <p>In each of next <em>t</em> lines there is a single integer number
     <em>n</em> (0 < n &le; 50 000 000).</p>
     ```

   - in a dedicated subsection (use HTML <h4></h4> header for subsections), e.g.:

     ```
     <h3>Input</h3>
     <p>The first line of the input consist of a single integer number
     ```

```
<em>t</em> which determines the number of tests.</p>
<p>In each of next <em>t</em> lines there is a single integer number
<em>n</em>.</p>
<h4>Constraints</h4>
<ul>
<li>0 &lt; t &le; 1000</li>
<li>0 &lt; n &le; 50 000 000</li>
</ul>
```

4. **Section with scoring explanation**
   This section is very important when you build your own master judge for a custom scoring algorithm.
5. **Section with test cases description**
   If your problem has many test cases each of them is probably dedicated to test some aspects of correctness of user's solution. If this information doesn't need to be a secret you may share this information with problem solvers in dedicated section. For example, every test case may have different constraints for variables.

Final notes

- If you have problem with formatting by WYSiWYG editor you can toggle the view to the raw HTML form with a checkbox under the problem description. After corrections you can go back to the editor by using the same checkbox.
- If you paste your problem description from other editor please make sure that the formatting isn't broken.
- If you have any question please don't hesitate to use contact@spoj.com.

References

http://docs.sphere-engine.com - a handbook for problem setters