Strings

Problem 1. Reverse string

- Write a JavaScript function that reverses a string and returns it.

Example:

input	output
sample	elpmas

Problem 2. Correct brackets

- Write a JavaScript function to check if in a given expression the brackets are put correctly.

Example of correct expression: `((a+b)/5-d)`. Example of incorrect expression: `)(a+b))`.

Problem 3. Sub-string in text

- Write a JavaScript function that finds how many times a substring is contained in a given text (perform case insensitive search).

Example:

The target sub-string is `in`
The text is as follows: We are liv**in**g **in** an yellow submar**in**e. We don't have anyth**in**g else. **in**side the submar**in**e is very tight. So we are dr**inkin**g all the day. We will move out of it **in** 5 days.

The result is: `9`

Problem 4. Parse tags

- You are given a text. Write a function that changes the text in all regions:

  - `<upcase>text</upcase>` to uppercase.
  - `<lowcase>text</lowcase>` to lowercase
  - `<mixcase>text</mixcase>` to mix casing(random)

Example: We are `<mixcase>living</mixcase>` in a `<upcase>yellow submarine</upcase>`.

We `<mixcase>don't</mixcase>`have `<lowcase>anything</lowcase>` else.

The expected result:

We are LiVinG in a YELLOW SUBMARINE. We dOn'T have anything else.

Note: Regions can be nested.

Problem 5. nbsp

- Write a function that replaces non breaking white-spaces in a text with ` `

Problem 6. Extract text from HTML

- Write a function that extracts the content of a html page given as text.
- The function should return anything that is in a tag, without the tags.

Example:

```
<html>
  <head>
    <title>Sample site</title>
  </head>
  <body>
    <div>text
      <div>more text</div>
      and more...
    </div>
    in body
  </body>
</html>
```

Result: Sample sitetextmore textand more...in body

Problem 7. Parse URL

- Write a script that parses an URL address given in the format: `[protocol]://[server]/[resource]` and extracts from it the `[protocol]`, `[server]` and `[resource]` elements.

- Return the elements in a JSON object.

Example:

URL	result
`http://telerikacademy.com/Courses/Courses/Details/239`	{ protocol: `http`, server: `telerikacademy.com` resource: `/Courses/Courses/Details/239`

Problem 8. Replace tags

- Write a JavaScript function that replaces in a HTML document given as string all the tags `<a href="…">…</a>` with corresponding tags `[URL=…]…[/URL]`.

Example:

input	output
`<p>Please visit <a href="http://academy.telerik.com">our site</a> to choose a training course. Also visit <a href="www.devbg.org">our forum</a> to discuss the courses.</p>`	`<p>Please visit [URL=http://academy.telerik.com]our site[/URL] to choose a training course. Also visit [URL=www.devbg.org]our forum[/URL] to discuss the courses.</p>`

Problem 9. Extract e-mails

- Write a function for extracting all email addresses from given text.
- All sub-strings that match the format @… should be recognized as emails.
- Return the emails as array of strings.

Problem 10. Find palindromes

- Write a program that extracts from a given text all palindromes, e.g. "ABBA", "lamal", "exe".

Problem 11. String format

- Write a function that formats a string using placeholders.
- The function should work with up to 30 placeholders and all types.

Examples:

```
var str = stringFormat('Hello {0}!', 'Peter');
//str = 'Hello Peter!';

var frmt = '{0}, {1}, {0} text {2}';
var str = stringFormat(frmt, 1, 'Pesho', 'Gosho');
//str = '1, Pesho, 1 text Gosho'
```

Problem 12. Generate list

- Write a function that creates a HTML `<ul>` using a template for every HTML `<li>`.
- The source of the list should be an array of elements.
- Replace all placeholders marked with –{…}– with the value of the corresponding property of the object.

Example: HTML:

```
<div data-type="template" id="list-item">
 <strong>-{name}-</strong> <span>-{age}-</span>
/div>
```

JavaScript:

```
var people = [{name: 'Peter', age: 14},…];
var tmpl = document.getElementById('list-item').innerHtml;
var peopleList = generateList(people, template);
//peopleList = '<ul><li><strong>Peter</strong> <span>14</span></li><li>…</li>…</ul>'
```