Topics in Modern Computer Science Spring 2016

Michael Kohlhase
Jacobs University Bremen

May 6, 2016

Contents

1 Assignment 1 (Character Representations) – Given Feb. 19., Due Feb. 25.

Problem 1.1 (Number Representations)
The following numbers are given to you in a base denoted by the subscript: 10_{10}, 10_2, 700_8, $ABCDEF_{16}$, $A1B2C3_{16}$, 5_{10}, 5_8 Please provide a representation for each of these numbers in base 2, base 8, base 10, and base 16. Please provide your answer by filling out this table and **SHOW**[1] all your computations: 25pt

Base 2	**Base 8**	**Base 10**	**Base 16**
		10_{10}	
10_2			
	700_8		
			$ABCDEF_{16}$
			$A1B2C3_{16}$
		5_{10}	
	5_8		

Solution: 1*: $101010111100110111101111_2$

Base 2	**Base 8**	**Base 10**	**Base 16**
1010_2	12_8	10_{10}	A_{16}
10_2	2_8	2_{10}	2_{16}
111000000_2	700_8	448_{10}	$1C0_{16}$
1*	52746757_8	11259375_{10}	$ABCDEF_{16}$
2*	50331303_8	10597059_{10}	$A1B2C3_{16}$
101_2	5_8	5_{10}	5_{16}
101_2	5_8	5_{10}	5_{16}

2*: $101000011011001011000011_2$

Problem 1.2 (UTF Encodings)
Encode the following code points into UTF-8 and UTF-16 (use binary representation) and look up the corrsponding characters: 15_{10}, $007F_{16}$, 80_{10}, $FFFF_{16}$, and 10437_{10}. 30pt

Please provide your answer by filling out this table and **SHOW**[2] all your computations/reasoning:

[1]Note that if you do not show your computations, we have to assume that you have used a calculator and do not deserve points.

[2]Note that if you do not show your computations, we have to assume that you have used a calculator and do not deserve points.

Code point	UTF-8	UTF-16	char
15_{10}			
$007F_{16}$			
80_{10}			
$FFFF_{16}$			
10437_{10}			

Solution:

Code point	**UTF-8**	**UTF-16**	**char**
15_{10}	00010101	0000000000010101	NEGATIVE ACKNOWLEDGE
$007F_{16}$	01111111	0000000001111111	DELETE
80_{10}	01010000	0000000001010000	LATIN CAPITAL LETTER P
$FFFF_{16}$	11101111 10111111 10111111	1111111111111111	NOT A CHARACTER
10437_{10}	11100010 10100011 10000101	0010100011000101	BRAILLE PATTERN DOTS-1378

Problem 1.3 (UTF-8/UTF-16)

Fill out the following table with how much disk space the specified text would occupy. Use the largest binary prefix that makes sense! (example: say 1.2 gibibytes instead of 1200 mebibytes). Justify your answers. 30pt

#	**Text description**	**Size on disk**
1	`ASCII` text of 500 characters	
2	`ASCII` text of 500 characters encoded as UTF-8	
3	`ASCII` text of 500 characters encoded as UTF-16	
4	Non-`ASCII` text of 1000 characters encoded as UTF-8	
5	Non-`ASCII` text of 1000 characters encoded as UTF-16	

Note: For non-`ASCII` text assume the worst case, i.e. a text which has an encoding of maximal length.

Solution:

Text description	**Size on disk**
ASCII text of 500 characters	500 bytes
ASCII text of 500 characters encoded as UTF-8	500 bytes
ASCII text of 500 characters encoded as UTF-16	1000 bytes = 1 kilobyte
Non-ASCII text* of 1000 characters encoded as UTF-8	4000 bytes = 4 kilobytes = 3.90 kibibytes
Non-ASCII text* of 1000 characters encoded as UTF-16	4000 bytes = 4 kilobytes = 3.90 kibibytes

Problem 1.4 (Internationalized Resource Identifiers)

15pt

Internationalized Resource Identifiers (IRIs) extend the `ASCII`-based URIs by the Universal Character Set (Unicode).

For compatibility reasons, the unicode characters are mapped to `ASCII` characters. Every non-`ASCII` character gets first mapped to its UTF-8 representation. Then each byte of the UTF-8 representation gets represented by three characters. The first character is the percent sign (%), and the other two characters are the hexadecimal representation of the byte. For example the byte 142 would be represented as `%8E`.

1. URI-encode the IRIhttp://Fußgängerübergänge.de ("Fußgängerübergänge" $\hat{=}$ "Pedestrian crossings").
2. Let's assume you want to trick someone (for a good reason of course). You provide the identifier `http://example.com`, but you replaced the `e` and `a` by the cyrillic characters, which look the same. What would be the mapping to a URI?

Hint: The cyrillic a is unicode character 1077_{10}, and the cyrillic e is unicode character 1072_{10}.

Solution: Example: ä is character number 288, which is in binary 1110 0100. This gives us the UTF-8 encoding 1100 0011 1010 0100 or C3A4. Therefore the escaped character is %C3%A4.

The cyrillic a is unicode character 1072_{10}, and the cyrillic e is unicode character 1077_{10}.

This gives us the solution

1. http://Fu%C3%9Fg%C3%A4nger%C3%BCberg%C3%A4nge.de
2. http://%D0%B5x%D0%B0mpl%D0%B5.com

2 Assignment 2 (Web Basics) – Given Feb. 27., Due Mar. 4.

Note: All the "code" (here HTML) must be submitted as text files (*.txt). Please add comments to what you (intend to) do. Please also make sure that the HTML pages run in a browser so that they can be tested.

Problem 2.1 (Web Definitions)
Answer the following questions in a clear and concise manner: 20pt

1. What is the difference between a web page and a web site? What about hyperlink and hypertext?
2. Can you use a URN to identify the location of the web resource? Why (not)?
3. What is a web browser? And what does HTTP stand for? How are browsers and HTTP connected?

Solution:

Problem 2.2 (HTTP Requests)
The following is an example of an HTTP request used to fetch a plain HTML page: 30pt

```
GET /hello.htm HTTP/1.1
User−Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)
Host: www.tutorialspoint.com
Accept−Language: en−us
Accept−Encoding: gzip, deflate
Connection: Keep−Alive
```

1. Answer the following questions:
   (a) What is the full address of the retrieved HTML page?
   (b) What version of HTTP does this request use?
   (c) What type of request is it?
2. In the lecture we learned that the HTTP protocol runs over telnet. Try sending the HTTP request above via telnet and see (i.e. protocol) what you get as a return. On `UNIX` systems (`linux` or `Mac OS X`) you can just type

   ```
   telnet www.tutorialspoint.com 80
   ```

   into the Terminal application (`Mac OS X`; called shell or bash in `linux`). you will get a response like

   ```
   Trying 93.184.220.20...
   Connected to gp1.wac.v2cdn.net.
   Escape character is '^]'.
   ```

   Then you can just paste the GET request above and see what happens.
   For `Windows`, you can use the command prompt application, but you may have to enable telnet via the control panel: goto "Programs and Features" click "Turn Windows Features on or off" and then "Enable Telnet Client and" and click OK.

3. Write another request of this type to retrieve your “Topics of Modern Computer Science” lecture notes. What happens when you run that via telnet?

Solution:

Problem 2.3 (DNA Encoding)

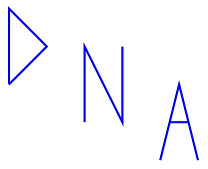
Create a simple HTML page, in which you describe a new encoding for storing DNA. 30pt

1. DNA can be stored as a sequence of the nucleotides C, G, A, and T. Start by creating a title and a short introductory paragraph.
2. Then list the nucleotides in an unordered list (<ul>).
3. Each nucleotide can be encoded as a pair of zeroes and ones. Create a HTML table that shows your encoding.
4. Of course we adhere scientific practices even on web pages, so we reference where we got all the information in the page. Conclude the page with a list of references, hyprelink the references.

Solution:

Problem 2.4 (Encoding a DNA Logo in *SVG*)

Add a logo to your previous website about DNA encoding. Your logo should roughly look like 20pt



In HTML5, the current version of HTML, we can use *SVG* for this. For the logo you only need to know that *SVG* can specify line segments by specifying their endpoints. For example the “red cross logo” on the left can be marked up by the svg tag on the right.

```
<svg width=200 height=200>
  <line x1="50" y1="50" x2="150" y2="150" stroke="red" stroke−width="2"/>
  <line x1="150" y1="50" x2="50" y2="150" stroke="red" stroke−width="2"/>
</svg>
```

Solution:

3 Assignment 3 (Cascading Style Sheets) – Given Mar. 3., Due Mar. 10.

Problem 3.1 (CSS Conversion Table)
Create an HTML page with a title (of your choice) and a table. The table will be filled with numbers. The first column should contain the decimal representation of the number, the second column should contain the binary representation, and the third column should contain the hexadecimal representation. 30pt

Put the type of the representation in the first row. Then pick 3 prime numbers, and 3 non-prime numbers enter them in the table (in an arbitrary order).

Decimal	Binary	Hexadecimal
7	111	7
12	1100	C
...		

1. Create the HTML page and use different `class`es for rows containing prime numbers and rows containing other numbers. Use a third `class` for the title row.

2. Create a stylesheet, which sets the font-sizes to relatively small fonts (e.g. 11pt for the table contents). Additionally, use different background colors for the different kinds of rows. And use bold font for the title row.

3. Create a second stylesheet, which has different colors and significantly larger fonts.

4. Now combine the two stylesheets into a third stylesheet. For large screens, use the larger font-sizes, and for smaller screens, use the smaller once. Use media queries to achieve this. The following example displays everything which has class `head` with font size 12pt, unless browser window is larger than 600 pixels.

```
.head {
    font-size: 12pt;
}

@media (min-width: 600px) {
    .head {
       font-size: 24pt;
    }

    /* ... */
}
```

Note: Please submit your files as a zip archive.

Solution: This is an example html file:

```
<html>
    <head>
        <link rel="stylesheet" type="text/css" href="style.css"/>
    </head>
    <body>
        <h1 class="title">Conversions</h1>
        <table>
            <tr class="head"> <td>Decimal</td> <td>Binary</td> <td>Hexadecimal</td> </tr>
            <tr class="nonprime"> <td>6</td> <td>110</td> <td>6</td> </tr>
            <tr class="prime"> <td>7</td> <td>111</td> <td>7</td> </tr>
            <tr class="prime"> <td>11</td> <td>1011</td> <td>B</td> </tr>
            <tr class="nonprime"> <td>12</td> <td>1100</td> <td>C</td> </tr>
            <tr class="nonprime"> <td>16</td> <td>10000</td> <td>10</td> </tr>
            <tr class="prime"> <td>17</td> <td>10001</td> <td>11</td> </tr>
        </table>
    </body>
</html>
```

And here is one of the style sheets.

```
.title {
    font-size: 2em;
}

.head {
    font-weight: bold;
    font-size: 12pt;
    background-color: lightgray;
}

.prime {
    font-size: 12pt;
    background-color: lightblue;
}

.nonprime {
    font-size: 12pt;
    background-color: lightyellow;
}

@media (min-width: 600px) {
    .title {
        font-size: 4em;
    }
    .head {
        font-size: 24pt;
    }
    .prime {
        font-size: 24pt;
    }
    .nonprime {
        font-size: 24pt;
    }
}
```

Problem 3.2 (Responsive Web Design)
Your task is to create a responsive HTML page containing 12 boxes. You can use the following code as a starting point: 30pt

```
<html>
    <style type="text/css">
        .card {
            /* Don't change these */
            float: left;
            box-sizing: border-box;

            /* Feel free to play around with these */
            text-align: center;
            border-style: double;
            border-width: 3px;
            background: lightblue;
            line-height: 100px;
            height: 100px;

            /* Change the width to get a different layout */
            width: 33.33%;
        }
    </style>
    <body style="margin:0px">
        <h1>Programming Languages</h1>
        <div class="card">Python</div>
        <div class="card">Scala</div>
        <div class="card">C</div>
        <div class="card">Java</div>
        <div class="card">C++</div>
        <div class="card">Javascript</div>
        <div class="card">PHP</div>
        <div class="card">Prolog</div>
        <div class="card">Haskell</div>
        <div class="card">Standard ML</div>
        <div class="card">Perl</div>
        <div class="card">OCaml</div>
    </body>
</html>
```

Your tasks are the following

1. The example arranges the boxes in a 3-column layout. Use media queries to switch

to a 2-column layout when the width is less than 800px, and to a 1-column layout when the width is less than 400px by changing the width property.
2. Change the topic of the page and create cards about your new topic. Add brief descriptions/comments to each card. Change the design and make sure that your descriptions/comments have a different style than the titles.

Hint: You can put `div`s inside `div`s.

Note:
1. You can test this by resizing the browser window.
2. Please submit your files as a zip archive.

Solution: For the first part: Add the following lines to your CSS.

```
@media (max-width: 400px) {
    .card {
        width: 100%;
    }
}
```

Problem 3.3 (Image Filters)

Consider the following HTML page: 40pt

```
<html>
    <head>
        <link rel="stylesheet" type="text/css" href="images.css"/>
    </head>
    <body>
        <div class="normal">
            <div class="title">Original</div>
            <div class="description">The original, unchanged image</div>
            <img class="image" src="test.jpg"/>
        </div>
        <div class="gray">
            <div class="title">Grayscale</div>
            <div class="description">The image in grayscale</div>
            <img class="image" src="test.jpg"/>
        </div>
        <div class="inverted">
            <div class="title">Inverted</div>
            <div class="description">All colors are inverted</div>
            <img class="image" src="test.jpg"/>
        </div>
    </body>
</html>
```

It displays an image called test.jpg three times. Your tasks are the following:

1. Put a jpeg image in the same directory and rename it to test.jpg (or alternatively change the HTML).

2. Create the images.css file. The descriptions should always be in italics. The titles should always be larger than the descriptions, apart from the inverted box. In the inverted box the description should be larger. Make the title in the normal div blue, and the one in the gray div gray.

3. Read about CSS filters (e.g. at `http://www.w3schools.com/cssref/css3_pr_filter.asp`). Apply a grayscale filter to the gray div element and the invert filter to the inverted div element.

Hint: If you want to change e.g. only the font size of the title in the normal div, you can use the following CSS:

```
.normal > .title {
    font-size: 22pt;
}
```

Note: Please submit your solution as a zip archive.

Solution:

```
.normal > .title {
    color: blue;
    font-size: 24pt;
}

.normal > .description {
    font-size: 14pt;
    font-style: italic;
}

.gray {
    color: gray;
    filter: grayscale(100%);
    -webkit-filter: grayscale(100%);
}

.gray > .title {
    font-size: 24pt;
}

.gray > .description {
    font-size: 14pt;
    font-style: italic;
}


.inverted {
    color: white;
    filter: invert(100%);
    -webkit-filter: invert(100%);
}

.inverted > .title {
    font-size: 24pt;
}

.inverted > .description {
    font-size: 14pt;
    font-style: italic;
}
```

---

4 Assignment 4 (Towards Web Applications) – Given Mar. 10., Due Mar. 17.

Problem 4.1 (Login)
For this problem, you will need to install and configure AMPPS (`http://ampps.com/download`) or any other PHP development stack of your choice. Please make sure to download the appropriate version for your Operating System (e.g., Windows, Mac OS, Linux). If you decide on using AMPPS, follow one of the following tutorials on how to install it based on your OS: 80pt

1. Mac OS: `http://www.ampps.com/wiki/Installation_on_Mac`
2. Windows: `http://www.ampps.com/wiki/Install`
3. Linux: `http://ampps.com/wiki/Installing_AMPPS_on_Linux`

Before working on this problem, make sure that Apache is started. You can test that by going to your browser and trying to navigate to the URL **localhost/ampps-admin**. If everything worked fine, you should see the AMPPS dashboard. Now that you have a working webserver, please follow these tasks:

1. Navigate to the installation path of AMPPS (depends on your operating system; on Linux, it should be /usr/local/ampps). Now locate the **www** folder and create a file named **index.php** inside. Add the following code inside, save the file, and then refresh **localhost**:

```
<html>
    <body>
        <?php echo 'Hello world'; ?>
    </body>
</html>
```

2. Use the **Example 5.5.1** to create a simple form asking for your username and password. The action of your form should be **login.php**. Use additional HTML and CSS to make this page look like a login page (e.g., add a title, change background color, center the login form etc.), and modify the PHP code from **Example 5.5.7** or `http://www.w3schools.com/php/php_date.asp` to insert the current time of when you accessed the webpage above the login form. Use the following code snippet for the password field of your form:

```
Password: <input type="password" name="pass" />
```

3. Go back to the **www** folder and create a file named **login.php**. By modifying the code in **Example 5.5.8**, connect to the MySQL database located at **tropaeolum.arvixe.com** with username and password both **topmodcs**. Now select the database **topmodcs** and then insert the following code snippet to query this database and redirect the user:

```
session_start();
if (!empty($_GET["user"]) && !empty($_GET["pass"])) {
    $myusername = mysql_real_escape_string($_GET['user']);
    $mypassword = mysql_real_escape_string($_GET['pass']);
    $sql = "SELECT * FROM login
                WHERE User = '$myusername' and Pass = '$mypassword'";

    $result = mysql_query($sql);
    if (mysql_num_rows($result) > 0) {
        $_SESSION["user"] = $_GET["user"];
        unset($_SESSION["error"]);
    } else {
        $_SESSION["error"] = "Unable to login!";
        unset($_SESSION["user"]);
    }
} else {
    $_SESSION["error"] = "At least one of the fields is empty!";
    unset($_SESSION["user"]);
}
mysql_close($con);
header( 'Location: index.php' );
exit();
```

4. If you go back to **localhost**, you should see your login form again. If you try to submit it, you will be redirected back to the original page. The next step that you now need to perform is to provide meaningful error messages or to display that the user logged in. In order to achieve this, we will follow the **Example 5.5.9**. First, please include the following PHP code right after the beginning of the `<body>` tag and fill in the gaps:

```
<?php
    session_start();
    if(isset($_SESSION["user"])) {
        echo "<h1> Welcome, " . $_SESSION["user"] . "! </h1>";
        echo "<a href='logout.php'>Log out</a>";
    } else {
        if(isset($_SESSION["error"])) {
            echo "<div class='error'>" . $_SESSION["error"] . "</div>";
        }

        // YOUR FORMER HTML CODE FOR THE FORM HERE THE WAY IT'S
       // IN EXAMPLE 5.5.9 MAKE SURE TO ONLY INTEGRATE WHAT YOU
       // NEED FROM THE PREVIOUS PHP CODE DISPLAYING THE CLOCK IN HERE
}
>?
```

Now please create a CSS class **error** which should be applied to error messages.

5. The last thing we need to do is make the log out link function. Go back to the **www** folder and create a file called **logout.php**. Include the following code in it:

```
<?php
    session_start();
    unset($_SESSION["error"]);
    unset($_SESSION["user"]);
    header("Location: index.php");
    exit();
?>
```

6. When sending the data between files, we use the **GET** protocol (see the HTML form you created). Why is this extremely unsecure for a login form? What can we use instead to make it more secure?

Note: To test a successful login, use **topmodcs_user** as username and **topmodcs_pass** as password!

Problem 4.2 (Clock)

JavaScript can be used to manipulate the content of an HTML page without having to refresh the page. For example, we can use JavaScript to create a clock on the page which updates every second, instead of every time we refresh the page! Look at the following JavaScript code snippet: 20pt

```
<html>
    <head>
        <script>
            function updateClock() {
                var currentTime = new Date ();
                // YOUR CODE HERE
                document.getElementById("time").innerHTML =
                        currentTime.getFullYear();
                // THIS IS JUST THE EXAMPLE;
                // IT WILL LOOK DIFFERENT IN YOUR FINAL CODE
            }
        </script>
    </head>
    <body onload="updateClock(); setInterval('updateClock()', 1000 )">
        <div id="time"></div>
    </body>
</html>
```

By doing research at `http://www.w3schools.com/js/js_date_methods.asp`, update the code above such that your **clock.html** file will update the clock all the time!

5 Assignment 5 (XML) – Given Mar. 17., Due Mar. 31.

Problem 5.1 (XML Calendar)
As you learned in class, one can easily create his/her own XML language to fit a certain purpose. After leaving the TopModCS lecture this week, you met your friend Dr. Filipidon whose current method of doing time management is pretty bad. 60pt

You quickly think that you can use your newly acquired skills to help Dr. Filipidon by designing a calendar description language that allows him to specify appointments by their start time and duration. Some of the appointments are repeating, e.g. every week. Moreover, you thought that he will need multiple types of appointments, such as teleconferences, romantic dates, client meetings etc. (you are free to make your own choices). Finally, appointments can have attendees, which can be specified by name and e-mail address and a location.

Please perform the following tasks:

1. Formalize your own tags and attributes which allow you to specify all aspects of appointments and collect multiple of them into a calendar. Briefly describe the format for each of them and how they should be used in your own words.
2. Convince Dr. Filipidon of the power of your design by writing a valid XML file with at least 10 appointments including the following:
   (a) a meeting with your academic advisor in Research I on the 24th of February at 3:30 PM which you expect to last for 1 hour and 45 minutes.
   (b) on the same day, at 7:00 PM, you have a client meeting via Skype with Bill Gates which you don't know how much will take.
   (c) you already know that you will have a midterm on March 30th at 2:15 PM in East Hall.

   Make sure that all the aspects of appointments mentioned above are used in this file.
3. Do research at `http://relaxng.org/` and write a `RelaxNG` grammar (XML-style or compact, your choice) for your newly designed language. Explain why your example from point **2** can be written using your `RelaxNG` grammar.

Hint: Developing an XML language is a design process that iterates over all three items above. You start with an initial design (item 1), test whether it looks good with an example (item 2), and see whether you can express the design in a schema (item 3).

Note: There are many ways to make sure that your `RelaxNG` grammar from item 3. is correct and that your XML file from item 2 is accepted by the grammar. One of them is to use a `RelaxNG` validator (we will certainly be using one during grading), another is to use a `RelaxNG`-aware editor to write the example for Dr. Filipidon in the first place – it will also complain about errors in the schema. Unfortunately, there are no online validators, but `http://relaxng.org` lists some that you can easily install. Many of the commercial editors have free trial versions.

Problem 5.2 (XPath)

Your task is to provide five XPath expressions for the use in XHTML documents. 30pt

1. All headings (h1 to h6)
2. The first item of any unordered list
3. All a tags that have no href attribute
4. All images with a width attribute of more than 100.
5. All lists (unordered and ordered) that contain at least three entries

Make sure that you test your XPath expressions on a large XHTML document (you can discuss where to get that on the course forum). Alternatively, you can make small XHTML documents that specifically test aspects of your expressions ("unit tests").

Explain how you did the testing, and submit the unit tests with your solution.

Hint: There are a lot of useful online tutorials, such as `http://www.w3schools.com/xsl/xpath_intro.asp`. You can check your expressions on `http://www.freeformatter.com/xpath-tester.html`. There are a lot of example documents at `http://www.w3schools.com/html/html_examples.asp` - feel free to use and modify them for checking.

Solution:

1. //h1//h2|//h3|//h4|//h5|//h6|
2. //ol/li[position()=1]
3. //a[not (@href)]
4. //img[@width>100]
5. (//ul//ol)[child::li[last()>2]]|

Problem 5.3 (Selecting appointments via XPath)

We can use XPath to query elements from an XML file, so we will try that with your example from Problem 5.1: Imagine Dr. Filipidon's calendar file has gotten very big over the years; write XPath expressions for the following applications. 10pt

1. He wants to know about all of his appointments with Bill Gates (past and future)
2. He wants to find out about his past girl friends (whom did he have romantic dates with?) in the year 2013.
3. Did he ever have appointments on weekends?

Hint: You may want to re-think your language design from Problem 5.1 if you find the expressions above too hard to write.

6 Assignment 6 (Electronic Books) – Given Mar. 31., Due Apr. 7.

Problem 6.1 (XSLT)
Consider recipes stored in XML documents, following the (compact) relaxng schema 45pt

```
start = element recipe {
    element for { text },
    element ingredients {
        element ingredient {
            attribute amount { text }?,
            text
        }+
    }?,
    element instructions { text }
}
```

An example recipe would for example be

```
<recipe>
    <for>Pancakes</for>
    <ingredients>
        <ingredient amount="500g">Flour</ingredient>
        <ingredient amount="400ml">Milk</ingredient>
        <ingredient amount="5">Eggs</ingredient>
    </ingredients>
    <instructions>
        Mix everything, then start baking.
    </instructions>
</recipe>
```

Your task is to provide an XSLT stylesheet that transforms such recipes into HTML representations that can be displayed in a browser. For the example recipe, the resulting HTML should be something like

```
<html>
    <body>
        <h1>Recipe for Pancakes</h1>
        <h2>Ingredients</h2>
        <table>
            <tr>
                <th>Ingredient</th>
                <th>Amount</th>
            </tr>
            <tr>
                <td>Flour</td>
                <td>500g</td>
            </tr>
            <tr>
                <td>Milk</td>
                <td>400ml</td>
            </tr>
            <tr>
                <td>Eggs</td>
                <td>5</td>
            </tr>
        </table>
        <h2>Instructions</h2>
        Mix everything, then start baking.
    </body>
</html>
```

Please consider that ingredients do not have to be provided! In that case, only the title *Recipe for ...* and the instructions section should be created.

Solution:

```
<xsl:stylesheet version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:template match="/">
        <html>
            <body>
                <h1>Recipe for <xsl:value-of select="/recipe/for"/></h1>
                <xsl:apply-templates select="/recipe/ingredients"/>
                <h2>Instructions</h2>
                <xsl:value-of select="/recipe/instructions/text()"/>
            </body>
        </html>
    </xsl:template>
    <xsl:template match="//ingredients">
        <h2>Ingredients</h2>
        <table>
            <tr><th>Ingredient</th><th>Amount</th></tr>
            <xsl:apply-templates select="./ingredient"/>
        </table>
    </xsl:template>
    <xsl:template match="//ingredient">
        <tr><td><xsl:value-of select="text()"/></td><td><xsl:value-of select="@amount"/></td>
    </xsl:template>
</xsl:stylesheet>
```

Problem 6.2 (EPUB and eBooks)

Please answer the following questions as precisely as possible. 25pt

1. How old is the concept of an `eBook` and what are they?
2. What is `EPUB` and what extension do `EPUB files` have?
3. Which technologies have you learned so far in the Topics of Modern CS course which help you build an `eBook` using the `EPUB` standard?
4. How should the code of an `eBook` written using the `EPUB` standard be organized like?

Problem 6.3 (EPUB Modifications)
Go to 30pt
`https://www.gutenberg.org/ebooks/1400.epub.images?session_id=59bd89d0642813811ec4e5d919596b8adc4e1d60`
in order to download the free eBook *Great Expectations by Charles Dickens* in ePub format.
Then follow these steps and document your findings:

1. Change the file's extension to `.zip` and extract the archive.
2. Go to the `OEBPS` folder and open the file named `@public@vhost@g@gutenberg@html@files@1400@1400-h@1400-h-0.htm.html` with a browser first and then with your favorite text editor.
3. After accessing the HTML code in the text editor, identify the book's name and re-write it with lowercase letters. Change the year of the edition to `2016`.
4. Download a picture of your choice and rename it to `@public@vhost@g@gutenberg@html@files@1400@1400-h@images@0012m.jpg`. Store it in the `OEBPS` folder by replacing the old one (**`!!!`** make sure that the picture is less than 1.5MB in size, otherwise you might have problems submitting on jGrader).
5. Now open the file `pgepub.css` and set the text color to `red`.
6. Create a new zip archive of the modified `OEBPS` folder together with the original `META-INF` folder and `mimetype` file.
7. Change the extension of your new archive to `.epub`
8. Download and install an **`ePub file reader`** and visualize the original ePub book and your modified one!

Note: By document your findings we mean to write a text file in which you describe at each step what you see/notice. Please submit an archive which contains your text file and the modified epub book!

---

7 Assignment 7 (GIT Basics) – Given Apr. 8., Due Apr. 15.

Problem 7.1 (Make a GitLab Account)
We will use the GitLab instance at `http://gl.kwarc.info` to manage your git repositories. Make an account there (the username should be the same as your Jacobs account name) and fill out your profile. 10pt

Note: Note that it can take some time until your account has been activated, so do this directly.

Problem 7.2 (Work on a Text File in a Private Repository)
After having installed git on your local machine (see the course notes and [CS14, section 1.5] for an introduction). 20pt

1. on GitLab make a repository called test and clone it on your local computer.
2. in the top directory of your working copy make a text file myfirst.txt and write your name and e-mail address into it.
3. add it to the repository, commit it, and push it.

Problem 7.3 (Clone/Update/Commit/Merge/Resolve)
Using your repository test from Problem 7.2 explore the revision control actions on your local machine. The trick here is that you can have multiple working copies on a single machine. 30pt

1. clone your repository test from Problem 7.2 at a different place. Let us call the old working copy W_o and the new one W_n.
2. make a change $\delta_1$to myfirst.txt in W_n, commit and push it.
3. pull W_o and make a change δ_2 to myfirst.txt and commit and push it.
4. (without pulling W_n) make a change δ_3 to myfirst.txt in a different line that δ_2 was. Can you commit? Why not? What happens if you pull?
5. (without pulling W_o) make a change δ_4 to myfirst.txt in the same line that δ_3 was. What happens if you pull now?
6. resolve the conflict in W_o and commit/push.

Give a log of what you did and what you saw. To show that you did all of this, please

1. make a screenshot of the commits page in GitLab and submit it as commits.png – mine is at `https://gl.kwarc.info/mkohlhase/test/commits/master`, but explore the GitLab user interface yourself.
2. make a screenshot of the development timeline of your project and submit it as development.png – mine is at `https://gl.kwarc.info/mkohlhase/test/network/master`.

Note: If you have messed up your working copy, you can just clone new working copies and start over. If you have messed up your repository as well, you can delete that (on GitLab; go to your project, then to "Settings" in the left menu, then to "Project Settings", and in that page all the way down to the red box with "Remove Project") and then make a new one with the same name.

8 Assignment 8 (GIT Branches) – Given Apr. 15., Due Apr. 22.

Problem 8.1 (Git Branches)

git supports working on several branches. 35pt

1. On your local machine, create a directory called gitbranches and init it in order to get an empty repository.
2. Create two files a.txt and b.txt containing each one line of text. add and commit both of them.
3. Create and checkout a new branch called crazyidea.
4. Add a line with some text to a.txt, add it, and commit it.
5. checkout the master branch.
6. What is the content of a.txt? Why?
7. Create and checkout a new branch called bugfix.
8. Add a line with some text to b.txt, add it, and commit it.
9. checkout the master branch and merge the bugfix branch.
10. checkout the crazyidea branch. What is the content of b.txt? Why?
11. merge the bugfix branch.
12. Add another line with some text to a.txt, add it, and commit it.
13. checkout the master branch and merge the changes from crazyidea.

Note:

1. Please document every step!
2. You can see the current branch using git branch.

9 Assignment 9 (A Bug Report) – Given Apr. 23., Due Apr. 30.

Problem 9.1 (Bug Report)
Oh no! While browsing the internet, Dr. Filipidon has encountered a wild bug in Internet Explorer! Because he is a good person, Dr. Filipidon wants to send a **bug report** (issue) to Microsoft. By doing some research, he gathered the following facts: 20pt

1. The website where the issue occurs is `example.com`.
2. The bug happens about every second time he loads the page, and it never happens in Google Chrome.
3. The browser either crashes completely or enters a blocked state.
4. The bug disappears if he disables JavaScript on his browser.
5. The bug disappears if he disables ActiveX, and re-enables JavaScript.
6. He has not encountered this issue on any other website before.

Help Dr. Filipidon formalize all these facts in a bug report!

Problem 9.2 (Markdown)
Being happy with his newly acquired bug report writing skills, Dr. Filipidon now wants to make it nicer in order to better catch the developer's attention. He found out that he can use a markdown language to achieve his goal! 20pt

Using **Example 7.4.9** as an example of **Markdown syntax**, format your bug report from **Problem 1**! Make sure to use at least one *heading*, two *sub-headings*, one *list*, one *link*, and one element of *text formatting* (e.g., bold, italic, underline). Of course, you are free to use as many as you want!

10 Assignment 10 (A Bug Report) – Given Apr. 30., Due May. 7.

Problem 10.1 (Pattern Matching)

Write down regular expressions matching 30pt

1. any valid binary number (e.g. 101011, 1101, and 0, but *not* 00101 or 130)
2. any even decimal number (e.g. 42, 4, and 366, but *not* 1 or 365)
3. any comma-separated sequence of increasing (decimal) digits starting with a 0 (e.g. 0,3,4,6,9, 0,3,5,9, and 0, but *not* 0,3,1 or 0,2,2)
4. wrong usage of a/an (e.g. a apple and an banana, but *not* an apple or banana shake). It is okay, if you restrict it to cases when a is followed by a vowel and an is followed by a consonant, i.e. it's okay if you also match a university.

Problem 10.2 Write a regular expression that recognizes roman numerals up to 3999 (i.e. MMMCMXCIX). Numbers of the form MDCCLIX (which stands for 1759) should be recognized, but not DMCXI or CXVIIII. 20pt

Note: We recap the symbol/value relation for reference.

Symbol	$\hat{=}$	Value	Symbol	$\hat{=}$	Value
I	$\hat{=}$	1	V	$\hat{=}$	5
X	$\hat{=}$	10	L	$\hat{=}$	50
C	$\hat{=}$	100	D	$\hat{=}$	500
M	$\hat{=}$	1,000			

Problem 10.3 (Pattern Matching)

You have offered to correct a friend's paper. Your friend is is not used to the metric system and used M instead of m for meters. You decide to use the sed stream editor to fix it. Give the sed invocation 10pt

Hint: For testing you can use `http://www.regexe.com/`.

Note: There are several special cases to consider. For example

```
26M + 23M = 49M. But don't replace the 2 Moldovans.
```

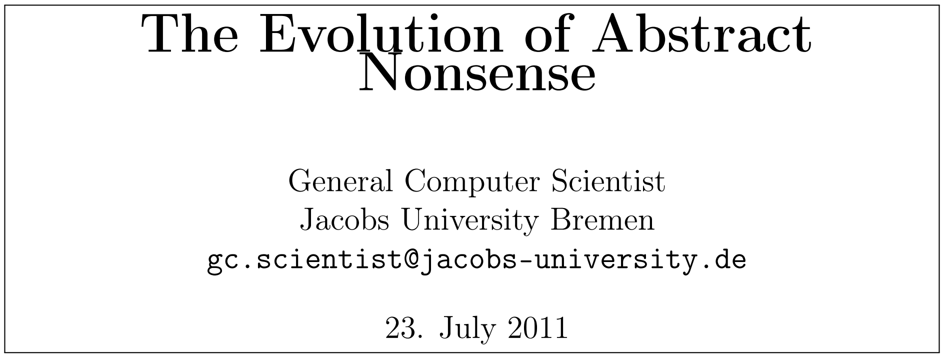
should result in

```
26 m + 23m = 49m. But don't replace the 2 Moldovans.
```

11 Assignment 11 (LaTeX) – Given May. 6., Due May. 13.

Problem 11.1 (A LaTeX with Title)

Write a document with a title, the date of today, and yourself as an author (with Jacobs University as the affiliation) It should look like this: 10pt



The Evolution of Abstract Nonsense

General Computer Scientist
Jacobs University Bremen
`gc.scientist@jacobs-university.de`

23. July 2011

Problem 11.2 (A LaTeX Document with Sections and Table of Content)

Extend the document from ?doctitle? with a couple of sections and subsections of your choice via the \section macro for sections and (correspondingly) \subsection for subsections. 30pt

Cross-reference various of the sections using the \label and \ref macros.

Hint: When you use the hyperref package (use \usepackage{hyperref} at the very end of the preamble), then the references become hyper-references (clickable in the PDF). Try this on your document!

Problem 11.3 (Simple Math Formulae)

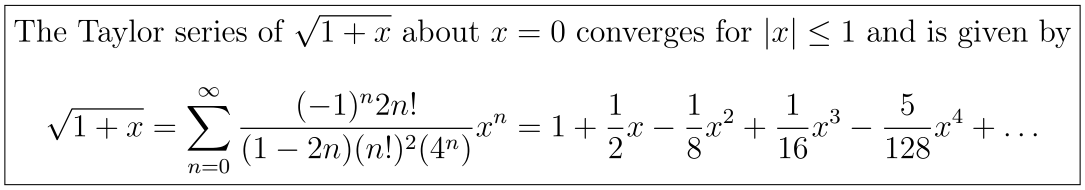
20pt

The solutions of the quadratic equation $ax^2 + bx + c = 0$ are $\dfrac{-b \pm \sqrt{b^2 - 4ac}}{2a}$.

Write this in LaTeX

Problem 11.4 (A more complex Math Formula)

20pt



The Taylor series of $\sqrt{1+x}$ about $x = 0$ converges for $|x| \leq 1$ and is given by

$$\sqrt{1+x} = \sum_{n=0}^{\infty} \frac{(-1)^n 2n!}{(1-2n)(n!)^2(4^n)} x^n = 1 + \frac{1}{2}x - \frac{1}{8}x^2 + \frac{1}{16}x^3 - \frac{5}{128}x^4 + \ldots$$

Write this in LaTeX, but note that the last multi-equation is in "display style" (i.e. centered and with bigger fonts).

References

[CS14] Scott Chacon and Ben Straub. *Pro Git*. APress, 2nd edition edition, 2014.