CSE 190 M, Spring 2008
Practice Final Exam 1

1. HTML/CSS Interpreting (20 points)

Draw a picture of how the following HTML and CSS code will look when the browser renders it onscreen. Indicate a background coloring by shading lightly or by drawing repeated diagonal lines like this. Assume that `stickman.png` is a generic picture of a stick man.

HTML

```
<h1>Scene</h1> <h1>#7</h1>

<div id="a">Elizabeth: Captain Barbosa, I am here to negotiate
the cessation of hostilities against Port Royal.</div>

<div>Barbossa: <img src="stickman.png" alt="stick man" />
There are a lot of long words in there,
Miss.<br />What is it that you want?</div>

<div class="b">
  <div class="c">Elizabeth: I want you to leave</div>
  <div class="c">and never come back.</div>
</div>

<div class="c" id="d">Barbossa: I'm disinclined to acquiesce
to your request. Means &quot;no.&quot;</div>
```

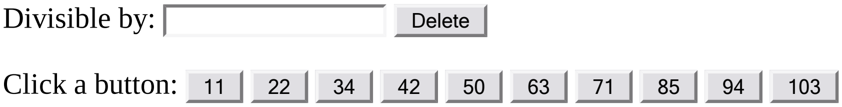
CSS

```
img     { vertical-align: top; width: 10em; height: 10em; }
#a      { border: 2px solid black; float: left;
          padding-right: 3em; width: 33%; }
.a #a   { background-color: yellow; }
.b div  { display: inline; }
.b, .c  { text-decoration: underline; }
.b > .c { background-color: yellow; }
#d      { float: right; }
```

2. Javascript/DOM (20 points)

Write the Javascript code to accompany the following HTML code, so that when the Delete button is clicked, any button whose text value is divisible by the number written in the text field is removed from the page. You may assume that a valid number has been typed in the text field. The HTML code is the following:

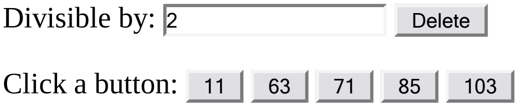
```
<div id="q2controls">
    Divisible by:
    <input type="text" id="divisor" />
    <button id="del">Delete</button>
</div>

<div id="q2buttons">
    Click a button:
    <button>11</button> <button>22</button>
    <button>34</button> <button>42</button>
    <button>50</button> <button>63</button>
    <button>71</button> <button>85</button>
    <button>94</button> <button>103</button>
</div>
```



Divisible by: Delete

Click a button: 11 22 34 42 50 63 71 85 94 103

For example, after typing the number 2 into the text field and pressing Delete, the following should be the page appearance:



Divisible by: 2 Delete

Click a button: 11 63 71 85 103

3. Ajax/XML (20 points)

Write the Ajax Javascript code to fetch and display XML data from the file named `movie.xml` (in the same directory as your code). This file contains lines spoken by a character in a movie. Your code should process the XML and display the character's lines, each in its own paragraph, in the format shown below. Assume that the code will execute on an HTML page containing a `div` with the CSS ID of `q3html`, and insert the paragraphs into this `div`.

The XML data will be in a format that matches the following abbreviated example:

```
<character name="Captain Jack Sparrow" actor="Johnny Depp" />
    <line time="02:55">Captain Jack Sparrow, if you please, sir.</line>
    <line time="3:21">I'm in the market as it were.</line>
    ...
    <line time="2:41:38">On deck, you scabrous dogs! Man the braces!
    Let down and haul to run free. Now... bring me that horizon.  And
    really bad eggs?  Drink up, me 'earties, yo ho.</line>
</character>
```

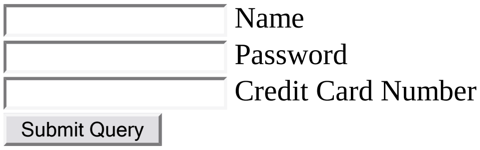
For the XML data above, your code would produce the following content on the HTML page:

4. PHP (20 points)

Write PHP code that processes the following form:

```
<form action="q4.php" method="post">
    <fieldset>
        <input type="text" name="name" /> Name <br />
        <input type="text" name="pw" /> Password <br />
        <input type="text" name="cc" /> Credit Card Number <br />
        <input type="submit" />
    </fieldset>
</form>
```

(Onscreen, the form looks like this:)



Your code should examine the name, password, and credit card number submitted, and verify that they are valid. A valid name is any non-empty string. A valid password is any string that is at least 6 characters long. A valid credit card number contains exactly 16 digits. Optionally, the credit card number can contain dashes between some or all groups of four digits. No other characters may be part of a credit card number. For example, the following are some examples of valid and invalid credit card numbers:

Valid	Invalid
1234567812345678	2457.1543.4367.4093
2457-1543-4367-4093	foo1234567812345678
39485098-81902375	12345678123456789
9834-34256678-9827	1234-5678-1234-5678-
	12-34-56-78-12-34-56-78

Your PHP code's output should be a two lines of text, the first stating whether the data was valid or invalid, and the second containing the data itself separated by commas. Replace the password by a string of `*` characters of equivalent length. Strip any dashes out of the credit card number while displaying it. For example, here are some outputs of your script for various form input:

Form Input	Output
Marty booyah! 1234-5678-1234-5678	Successful. Marty, *******, 1234567812345678
Kenneth hulk 11112222-33334444	Denied! Invalid data. Kenneth, ****, 1111222233334444
Jeff quailman 4321-4321x-4321-43210	Denied! Invalid data. Jeff, ********, 43214321x432143210

Use regular expressions for pattern matching and replacement.

5. SQL (20 points)

Write an SQL query that will return the names of all characters that appeared in **two or more** of the *Pirates of the Caribbean* movies; that is, movies whose name contains the substring "Pirates of the Caribbean". Ensure that the results are returned in alphabetical order. If it helps you, you may assume that the character is played by the same actor in both movies. The following is a subset of the results returned:

```
+-------------+
| role        |
+-------------+
| Himself     |
| Marine      |
| Will Turner |
+-------------+
3 rows in set (2 min 38.73 sec)
```

Recall that the `imdb` database contains the following tables:

actors

id	first_name	last_name	gender
433259	William	Shatner	M
797926	Britney	Spears	F
...			

movies

id	name	year	rating
112290	Fight Club	1999	8.5
210511	Memento	2000	8.7
...			

roles

actor_id	movie_id	role
433259	313398	James T. Kirk
797926	342189	Herself
...		