CSE 190 M, Spring 2011
Final Exam, Thursday, June 9, 2011

Name:	____		
Quiz Section:	____________________	**TA:**	____________________
Student ID #:	____________________		

Rules:

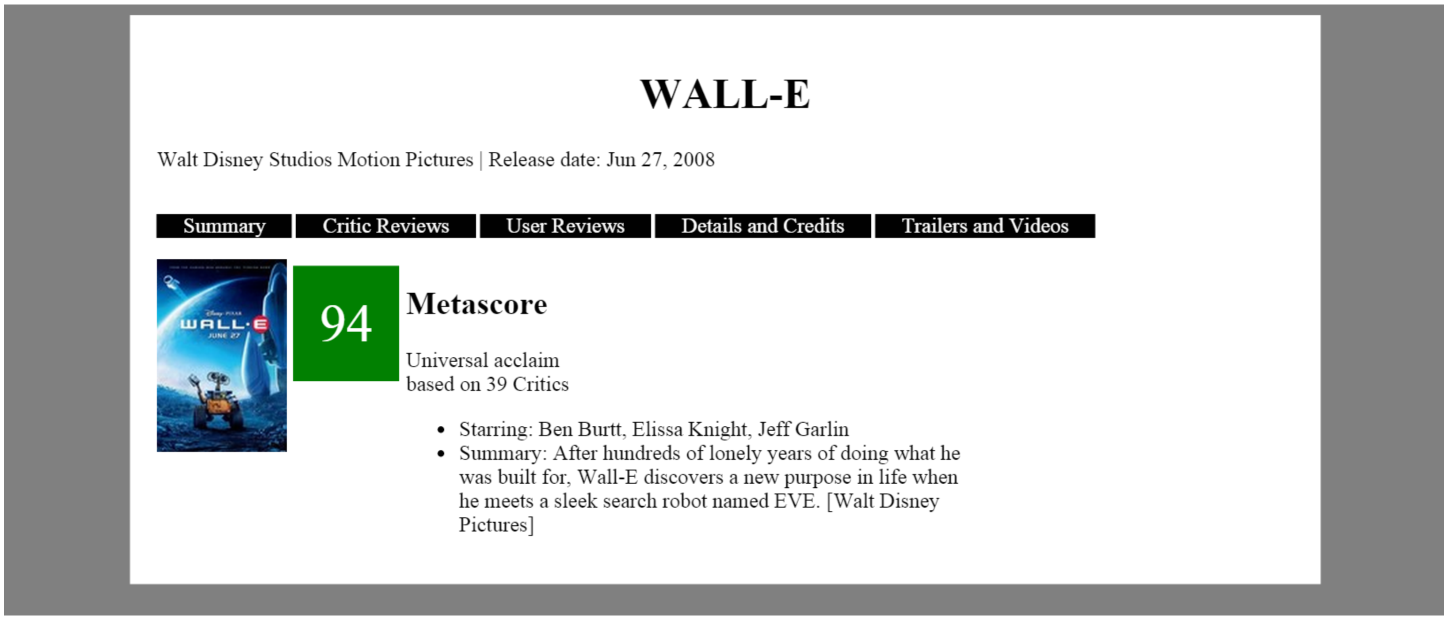
- You have **110 minutes** to complete this exam.
  You may receive a deduction if you keep working after the instructor calls for papers.
- This test is open-book/notes. You may use any paper resources other than practice exams.
- You may *not* use any computing devices, including calculators, cell phones, or music players.
- Unless otherwise indicated, your code will be graded on proper behavior/output, not on style.
- Please do not abbreviate code, such as writing ditto marks ("") or dot-dot-dot marks (...).
- If you enter the room, you must turn in an exam and will not be permitted to leave without doing so.
- You must show your **Student ID** to a TA or instructor for your submitted exam to be accepted.

Good luck!

Problem	**Description**	**Earned**	**Max**
1	CSS		
2	PHP		
3	PHP/JSON		
4	Ajax		
5	Regular Expressions		
6	SQL		
TOTAL	**Total Points**		**100**

1. CSS

Write the **CSS code** necessary to recreate the following appearance on-screen, exactly as shown.
The page uses the HTML code below. You are **not allowed to modify the HTML**.



WALL-E

Walt Disney Studios Motion Pictures | Release date: Jun 27, 2008

Summary | Critic Reviews | User Reviews | Details and Credits | Trailers and Videos

94

Metascore

Universal acclaim
based on 39 Critics

- Starring: Ben Burtt, Elissa Knight, Jeff Garlin
- Summary: After hundreds of lonely years of doing what he was built for, Wall-E discovers a new purpose in life when he meets a sleek search robot named EVE. [Walt Disney Pictures]

```
<body>
   <div id="all">
      <h1>WALL-E</h1>
      <p>
         Walt Disney Studios Motion Pictures
         | Release date: Jun 27, 2008
      </p>
      <div id="menu">
         <p>Summary</p>
         <p>Critic Reviews</p>
         <p>User Reviews</p>
         <p>Details and Credits</p>
         <p>Trailers and Videos</p>
      </div>
      <div id="pic">
         <img src="walle.jpg" alt="walle" />
      </div>
      <div id="rating">94</div>
      <div id="score">
         <h2>Metascore</h2>
         <p>
            Universal acclaim <br />
            based on 39 Critics
         </p>
         <ul>
            <li>
               Starring: Ben Burtt, Elissa
               Knight, Jeff Garlin
            </li>
            <li>
               Summary: After hundreds of
               lonely years of doing what he
               was built for, Wall-E
               discovers a new purpose in
               life when he meets a sleek
               search robot named EVE. [Walt
               Disney Pictures]
            </li>
         </ul>
      </div>
   </div>
</body>
```

The menu items (*Summary* to *Trailers and Videos*) are white text on a black background. There is 20px space between the words and the right and left sides of their black boxes. There is 3px of blank space to the right of each.

The score information should be 50% wide.

The movie rating has a green background and white text. The number is in 30pt font. There is 20px space between the number and the edges of the green background. There is 5px blank space around the green background.

The background color of the page is gray but the background color of the area containing everything is white. The white area is 80% wide and centered. There is 20px of space between the edge of the white background and everything inside it.

2. PHP

Write the PHP code for a partial web page that searches for names that contain a given letter a given number of times, case-insensitively. Your web service would be located in a file named `q4.php` on the server. In this same directory is a file `peeps.txt`; each line of this file contains a name. For example:

```
Helene Martin
Robert Purple
Marty Stepp
Stuart Reges
Bob Loblaw
```

The names are guaranteed to be non-empty and unique.

Your web service accepts a query parameter named `letter` indicating the character to be searched for, and a parameter named `times` indicating how many times that character must occur, case-insensitively. For each name that contains the letter at least the given number of times, an HTML paragraph should be printed indicating this with the name in bold along with the number of times it contains the letter. For example, if the following query is given, is given, this output is produced:

```
q4.php?letter=R&times=2
```

Robert Purple contains 'R' exactly 3 times.

Stuart Reges contains 'R' exactly 2 times.

If no names contain the given character the given number of times, output a message saying so:

```
q4.php?letter=x&times=1
```

No name contained 'x' enough times.

If one or both of the required query parameters is not passed, your service must produce an HTTP 400 Invalid Request error. You should also generate an HTTP 400 error if the `letter` parameter is not a one-letter string or if the `times` parameter is not a positive number. (Any `times` value ≤ 0 is invalid and anything else is valid.)

```
q4.php?letter=thingy&times=-4
q4.php?letter=t
q4.php?times=7
q4.php
```

HTTP/1.1 400 Invalid Request

Your code should *not* output a complete HTML page. Output a partial page containing only the paragraphs indicated. Do not use any `print` or `echo` statements in your code.

Write your answer on the next page.

3. PHP/JSON

Write the code for a web service `books.php` that outputs `JSON` data about books available at a book store in a particular category. This service should take a `GET` parameter named `category` and search a data file for all books in that category.

Your `PHP` code will read a data file named `books.txt`. Each line in that file matches the following format, with each token of information separated by a single vertical bar:

`name|author|category|date|price`

All tokens of data are guaranteed not to contain any vertical bars in them. You can assume all data in the file is valid although it may contain extra blank lines and these shouldn't crash your program or alter your output.

The `JSON` that is output should contain a number labeled "`count`". This should represent a count of all books at the book store. It should also contain a list called "`books`" that contains a list for each book in the passed in category. The list for each book should contain the year labeled as "`year`", the price labeled as "`price`", the title labeled as "`title`", and the author labeled as "`author`".

If the category parameter is not passed return a 400 status and stop execution immediately. If no books match the category you should send back the same data as usual, just leave the course array empty.

An example input file:

```
21 Burgers for the 21st Century|Stuart Reges|cooking|2010|75.00
Harry Potter and the Sorceror's Stone|J.K. Rowling|children|1998|19.99
Breakfast for Dinner|Amanda Camp|cooking|2009|22.00
```

Output using the above input file and `books.php?category=cooking`

```
{ "count":3,
  "books":[ {"year" : "2010",
              "price" : 75.00,
              "title" : "21 Burgers for the 21st Century",
              "author" : "Stuart Reges"
            },
            {"year" : "2009",
              "price" : 22.00,
              "title" : "Breakfast for Dinner",
              "author" : "Amanda Camp"
            }
          ]
}
```

4. Ajax/XML

Suppose that there is a web service named `adventure.php`, located on your web server in the same directory as your code. This service outputs XML data describing a situation and options a player can pick in a choose your own adventure style game. In this problem you will use Ajax to contact the web service (using a GET request), examine its XML data, and display the questions and options in the game.

The XML data returned by the web service consists of an overall document element called `adventure` that contains either a `situation` tag and an `answers` tag or an `end` tag. The `answers` tag contains a series of one or more `answer` elements inside it.

The format matches the following example (though it might have more/fewer than 4 answers):

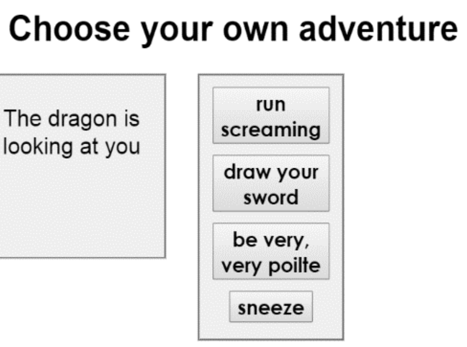
```
<adventure>
    <situation>The dragon is looking at you</situation>
    <answers>
            <answer>run screaming</answer>
            <answer>draw your sword</answer>
            <answer>be very, very polite</answer>
            <answer>sneeze</answer>
    </answers>
</adventure>
```

It may also match the following example:

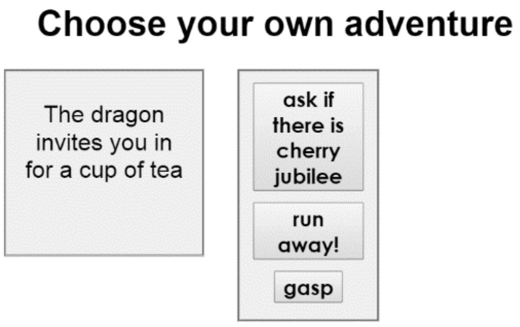
```
<adventure>
    <end reason="Uh-oh! The dragon accidentally stepped on you" />
</adventure>
```

Assume the code you are writing will go into `adventure.js`, linked from the following page:

```
<body>
      <h1>Choose your own adventure</h1>
      <div id="container">
            <fieldset id="situation">
                  <p id="situationparagraph"></p>
            </fieldset>
            <fieldset id="answer"></fieldset>
      </div>
</body>
```

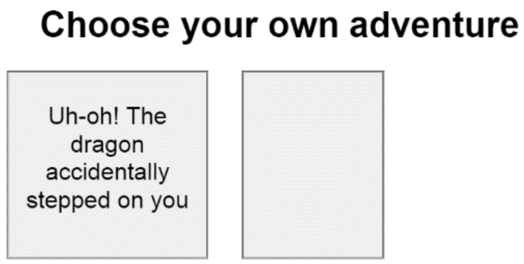


When the page loads send an AJAX request to the `adventure.php` service and pass it a parameter named `situation` with the value `begin`.


Choose your own adventure
The dragon invites you in for a cup of tea
ask if there is cherry jubilee
run away!
gasp

Insert the questions you get back into the `situationparagraph`. Add a `button` to the answer `fieldset` for each `answer` tag you receive. Set these buttons so that if they are clicked your page will send an Ajax request to `adventure.php` with the `situation` parameter set to the text on the button. This request will return the same sort of data which should be dealt with in the same way. The buttons displayed should only ever be the current options, no old options should show.

If instead of receiving XML back that contains a `question` tag you receive XML with an `end` tag, place the text of the `reason` attribute of the `end` tag in the `situationparagraph` and make sure no buttons are showing.


Choose your own adventure
Uh-oh! The dragon accidentally stepped on you

You can see three possible appearances of the page at left but there are many more. You do not have to write any CSS or apply any styles in Javascript for this problem.

You may assume that the XML data is valid in the format described previously, and that the .php service is reachable. If it returns a status other than 200 insert a descriptive message into the `situationparagraph` and make sure no buttons are showing.

4. Regular Expressions

a) Write a regular expression to match a US zipcode. Zipcodes contain 5 numbers. Sometimes there is a dash after the 5 numbers which is followed by 4 more numbers.

Valid:	Invalid:
`99999`	`abc12345`
`98115`	`123456789`
`67241-1234`	`12345-6`

b) Write a regular expression to match a price. A price always starts with a dollar sign. Any amount of numbers can come before the decimal. Two numbers should always follow the decimal.

Valid:	Invalid:
`$14.99`	`$14`
`$1234567.00`	`$134213.89money`
`$.90`	`$1.1a`

c) Write a regular expression to match a self closing HTML tag. A tag must start with "<" and end with "/>". It must have a name and can have 0, 1 or more properties. You do not need to wory about white space inside the tag.

Valid:	Invalid:
`<img src=”foo.jpg” />`	`</>`
`<img />`	`<img src= />`
`<a href=”foo.html” id=”stuff” />`	

[abc]	A single character of: a, b, or c	.	Any single character
[^abc]	Any single character except: a, b, or c	\s	Any whitespace character
[a-z]	Any single character in the range a-z	\S	Any non-whitespace character
[a-zA-Z]	Any single character in the range a-z or A-Z	\d	Any digit
^	Start of line	\D	Any non-digit
$	End of line	\w	Any word character (letter, number, underscore)
\A	Start of string	\W	Any non-word character
\z	End of string		
(...)	Capture everything enclosed	a+	One or more of a
(a|b)	a or b	a{3}	Exactly 3 of a
a?	Zero or one of a	a{3,}	3 or more of a
a*	Zero or more of a	a{3,6}	Between 3 and 6 of a

5. SQL

Write an SQL query to search the `imdb` database for all directors who have appeared in one of their own movies, playing two or more characters in th same movie. Show only the names of the director, in **alphabetical order** by last name ascending, breaking ties by first name in ascending order. Each director should be listed only once. Recall the `imdb` database tables:

actors

id	first_name	last_name	gender
433259	William	Shatner	M
797926	Britney	Spears	F
831289	Sigourney	Weaver	F
...			

movies

id	name	year	rank
112290	Fight Club	1999	8.5
209658	Meet the Parents	2000	7
210511	Memento	2000	8.7
...			

roles

actor_id	movie_id	role
433259	313398	Capt. James T. Kirk
433259	407323	Sgt. T.J. Hooker
797926	342189	Herself
...		

directors

id	first_name	last_name
24758	David	Fincher
66965	Jay	Roach
72723	William	Shatner
...		

movies_directors

director_id	movie_id
24758	112290
66965	209658
72723	313398
...	

movies_genres

movie_id	genre
209658	Comedy
313398	Action
313398	Sci-Fi
...	

When run on the `imdb` database, your query would produce the following results:

```
+------------+-----------+
| first_name | last_name |
+------------+-----------+
| LeVar      | Burton    |
| Santo      | Cilauro   |
| Claus Theo | Gartner   |
| Tom        | Gleisner  |
| Jane (I)   | Kennedy   |
| Tony (III) | Martin    |
| Bill       | Melendez  |
| Mick (I)   | Molloy    |
| Yves       | Renier    |
| Rob        | Sitch     |
| Jason      | Stephens  |
| James      | Tolkan    |
+------------+-----------+
12 rows in set
```

Note that actor IDs and director IDs are not equal, but you may assume that only one actor and director share the same first/last name. If you join too many tables together that are not needed for the query, you will not receive full credit. You should solve this problem using only the SQL syntax taught in class.