

**CSE 154, Spring 2013**  
**Final Exam, Thursday, June 13, 2013**

**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, but closed notes. You may *not* use printed/written notes or 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 (...).  
You may write **id** for `document.getElementById` and **qs** for `document.querySelectorAll`.
- You may not use JavaScript frameworks such as jQuery or Prototype when solving problems.
- 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! You can do it!*

<b>Problem</b>	<b>Description</b>	<b>Earned</b>	<b>Max</b>
1	HTML / CSS Tracing		20
2	CSS		20
3	PHP		20
4	JS / Ajax / JSON		20
5	SQL		20
X	Extra Credit		1
<b>TOTAL</b>	<b>Total Points</b>		<b>100</b>

## 1. HTML / CSS Tracing

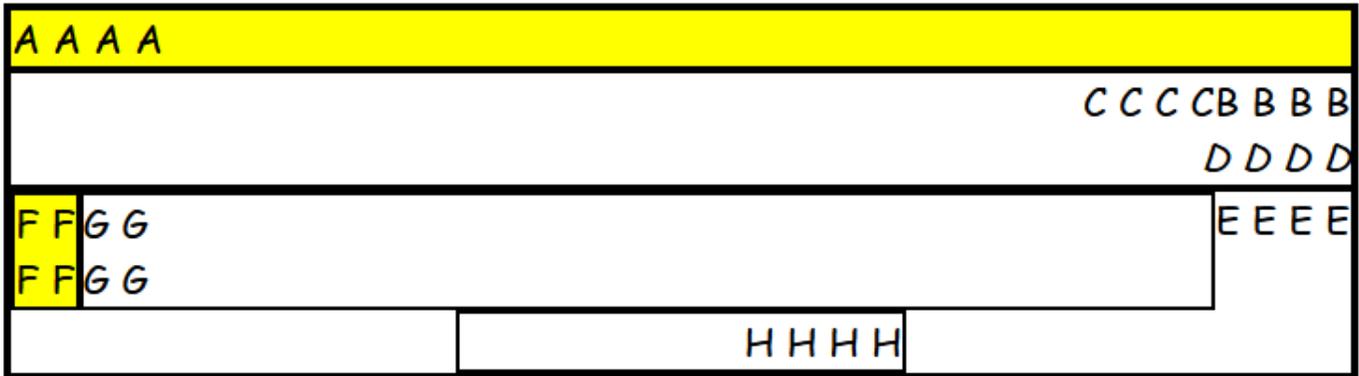
Draw a picture of how the following HTML/CSS code will look when the browser renders it on-screen. Assume that the HTML is wrapped in a valid full page with a head and body. Indicate a non-white background by shading lightly or by drawing diagonal lines like *this*. It is possible that some CSS rules shown will not apply to any elements.

<pre>&lt;div&gt; &lt;div class="id"&gt;A A A A&lt;/div&gt; &lt;div&gt;   &lt;span&gt;B B     B B&lt;/span&gt;   &lt;span id="id"&gt;C C C C&lt;/span&gt;   &lt;span class="class"&gt;D D D D&lt;/span&gt; &lt;/div&gt;  &lt;div class="class"&gt;   &lt;span&gt;E E E E&lt;/span&gt;   &lt;div class="id"&gt;F F&lt;br /&gt;F F&lt;/div&gt;   &lt;div id="class"&gt;G G&lt;br /&gt;G G&lt;/div&gt;   &lt;div id="H"&gt;H H H H&lt;/div&gt; &lt;/div&gt; &lt;/div&gt;</pre>	<pre>div { border: 2px solid black; padding: 1em; background-color: white; overflow: hidden; } div div &gt; div { background-color: red; } div &gt; div .class { text-decoration: underline; }  #id { padding: .9em; border: 2px dashed black; } .id { float: right; } .class #class { float: left; }</pre>
--	---

## 2. CSS

Write the **CSS code** necessary to recreate the following appearance on-screen, exactly as shown.

The page uses the same HTML code as in the previous problem. You are **not allowed to modify the HTML**.



```
<div>
  <div class="id">A A A A</div>
  <div>
    <span>B B
      B B</span>
    <span id="id">C C C C</span>
    <span class="class">D D D D</span>
  </div>

  <div class="class">
    <span>E E E E</span>
    <div class="id">F F<br />F F</div>
    <div id="class">G G<br />G G</div>
    <div id="H">H H H H</div>
  </div>
</div>
```

Text now uses Comic Sans MS font at the default size.

All borders shown are 2px thick and black in color.

The elements "A" and "F" have yellow background.

The element "D" now has *italic* text.

The element "H" is now each exactly one third (1/3) of the page width, in the page center; the "H H H H" text appears on the right edge of the element.

### 3. PHP

Write the PHP code for a web page `books.php` that searches for books in a bookstore for sale under a given maximum price. Assume that there is a provided page with a form where the user can type information about the category of books to search and the maximum price (inclusive) of book to show. The form will submit to your `books.php`. Here is the relevant HTML from that provided page, and a screenshot of its appearance:

```
<h1>Book Search</h1>
<form action="books.php" method="get">
  <div><input id="ab" type="text" name="category" /> Category</div>
  <div><input id="cd" type="text" name="max" /> Maximum Price</div>
  <div><input id="ef" type="checkbox" name="prime" /> Prime Club Member?</div>
  <div><input id="gh" type="submit" value="View Books" /></div>
</form>
```

#### Book Search

Each category has a corresponding text file in the current directory that contains all books for sale in that category; for example the "fantasy" category is found in `fantasy.txt`. Each line of these files contains a book name followed by a colon (:) and the price of the book. For example, below are `scifi.txt` and `fantasy.txt`:

I, Robot:10.00 Hitchhiker's Guide to the Galaxy:5.58 Do Androids Dream of Electric Sheep?:22.95 Ringworld:34.25 Tunnel in the Sky:16.75	Harry Potter:15.00 A Game of Thrones:15.89 Fellowship of the Ring:27.50 Dragonlance:12.95
---	--

Your PHP page should accept the query parameters from the above form, open the text file for the requested category, and search it for books whose price are less than the given maximum, inclusive. Match case-insensitively; the user might submit the category in any capitalization, but the input files are always in lower case. If the user types a category for which there is no corresponding text file, instead of the bulleted list, just display a paragraph saying, "Category not found." Your page should not produce any errors or warnings in such a case.

Display each matching book as a bullet in an unordered list, showing the price and the title separated by a dash (-). The form also has a "Prime Club Member" checkbox; if it is checked, you should reduce the price of every item by 20% before comparing it to the maximum and show the discounted prices in the list.

The following screenshots show the PHP page output after the user submits the form with various values:

<p><u>scifi, max price 20:</u></p> <p><b>Matching Books:</b></p> <ul style="list-style-type: none"><li>• \$10.00 - I, Robot</li><li>• \$5.58 - Hitchhiker's Guide to the Galaxy</li><li>• \$16.75 - Tunnel in the Sky</li></ul> <p>3 match(es).</p>	<p><u>scifi, max price 20, prime club:</u></p> <p><b>Matching Books:</b></p> <ul style="list-style-type: none"><li>• \$8 - I, Robot</li><li>• \$4.464 - Hitchhiker's Guide to the Galaxy</li><li>• \$18.36 - Do Androids Dream of Electric Sheep?</li><li>• \$13.4 - Tunnel in the Sky</li></ul> <p>4 match(es).</p>	<p><u>category "silly":</u></p> <p><b>Matching Books:</b></p> <p>Category not found.</p>
<p><u>fantasy, max price 15:</u></p> <p><b>Matching Books:</b></p> <ul style="list-style-type: none"><li>• \$15.00 - Harry Potter</li><li>• \$12.95 - Dragonlance</li></ul> <p>2 match(es).</p>	<p><u>fantasy, max 15, prime club:</u></p> <p><b>Matching Books:</b></p> <ul style="list-style-type: none"><li>• \$12 - Harry Potter</li><li>• \$12.712 - A Game of Thrones</li><li>• \$10.36 - Dragonlance</li></ul> <p>3 match(es).</p>	<p><u>fantasy, max 10:</u></p> <p><b>Matching Books:</b></p> <p>0 match(es).</p>

You may assume that the user will submit the form properly; assume all parameters are passed properly and have appropriate values (e.g. a non-negative max). You may assume that if the category file exists, it is valid in the format described above. You can write just the code that would go inside the page `body`; you don't need to output a `head` section or a complete page. Use the browser's default styling; do not write any CSS for this problem.

*Write your answer on the next page.*

### 3. PHP (writing space)

## 4. JavaScript / DOM

Write the **JavaScript code** to add behavior to the following page for manipulating strings. The page UI allows the user to type a phrase into a text box. The user can click the "Go!" button to display the words in that phrase in reverse order. Each word in the phrase should be inserted as a `span` with a class of `word`, inside a `div` with the `id` of `words`. (The `word` class gives the `spans` their appearance in the screenshots below, giving them a border and a background and so on.) Every other word (the first, third, fifth, etc.) should also be underlined.

The user can optionally specify a "filter" text by typing into a text box with the `id` of `filter`. If a non-blank filter is specified, you should exclude any words from the phrase that contain that filter text, case-insensitively. For example, if the filter text is "abc", exclude any words containing abc, ABC, aBc, etc. If any words are excluded, under the list of words you should modify the `div` with `id` of `count` to display text of the form, "5 word(s) filtered out".

The code should work for multiple clicks of the button. On each click it should clear any previous information you injected. You may assume that words in the phrase are separated by single spaces. Do not use any JavaScript libraries such as jQuery or Prototype. Here is the relevant HTML code for the page:

```
<h1>Sentence Reverser!</h1>
<div> Phrase: <input id="phrase" type="text" size="40" /> </div>
<div> Filter: <input id="filter" type="text" size="10" /> </div>
<div> <button id="go">Go!</button> </div>
<div id="words"></div>
<div id="count"></div>
```

These screenshots show the initial state, and after phrases have been typed and "Go!" is clicked.

### Sentence Reverser!

Phrase:

Filter:

you? ARE How hello

### Sentence Reverser!

Phrase:

Filter:

shore sea the by seashells sells she

### Sentence Reverser!

Phrase:

Filter:

you? ARE

2 word(s) filtered out.

### Sentence Reverser!

Phrase:

Filter:

shore the by she

3 word(s) filtered out.

Write/finish your answer on the next page.

#### 4. JavaScript / DOM (writing space)

## 5. SQL

Write an SQL query to search the `imdb` database for the names/years of all movies that had at least 2 directors, one of whom was Woody Allen, where Allen also acted in the movie. Show the movies in alphabetical order. Each actor/film combination should be listed only once. Woody Allen's first and last name in the database are "Woody" and "Allen" respectively, and you may assume that he is the only actor or director in the database with that exact first/last name pairing. Note that actor IDs do not match director IDs. Recall the `imdb` 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
...	

```

+-----+-----+
| name          | year |
+-----+-----+
| New York Stories | 1989 |
| What's Up, Tiger Lily? | 1966 |
+-----+-----+
2 rows in set (0.58 sec)

```

When run on `imdb` database, your query produces the results at left.

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 shown in class and the textbook.

**X. Extra Credit**

Draw a picture of what your TA would look like if he/she became a gangsta rapper or pop music star.

*(This is just for fun; any picture that appears to reflect more than a few moments' work will receive credit.)*