CSCI 3300 Assignment 6

Total estimated time for this assignment: **9 hours**

When you see "Richard Ricardo" in the example screen captures, change it to **<your name>**.
When you see "Richard" in the example screen captures, change it to **<your first name>**.
If you do not put **<your name> / <your first name>** in the above mentioned fields, you will get **0 points** for the question(s).

No two students should submit webpages with exactly the same code, or same content, or same layout, or same color combination. If found, **both** students will get **0 points**.

Create a folder on your hard disk, name the folder **lastname_firstname_assignment6.** Save all the files from this assignment in this folder.

Create the following subfolders (in the folder lastname_firstname_assignment6)**: q1, q2, q3**.
As a result, you should have the following folder (directory) structure for this assignment: (-2 points if wrong)

- lastname_firstname_assignment6\q1\
- lastname_firstname_assignment6\q2\
- lastname_firstname_assignment6\q3\

Use Firefox **Web Console** (Tools > Web Developer > Web Console) to help debugging JavaScript. All html (JavaScript) files must pass JavaScript validation at Firefox Web Console without any error, without any warning **(-2 points for each error, each warning**).

- Turn on your Firefox **Menu Bar** (Hint: right-click).
- Make sure your Firefox is up to date (Help > About Firefox).
- Turn on Web Console when you code (Tools > Web Developer > Web Console).
- Fix any error/warning immediately when you see them. Do NOT accumulate errors.

When you view page source in a web browser, **`<!DOCTYPE html>`** must be at the top of every page. In other words, all pages must be written in HTML5. **(-20 points** if not)

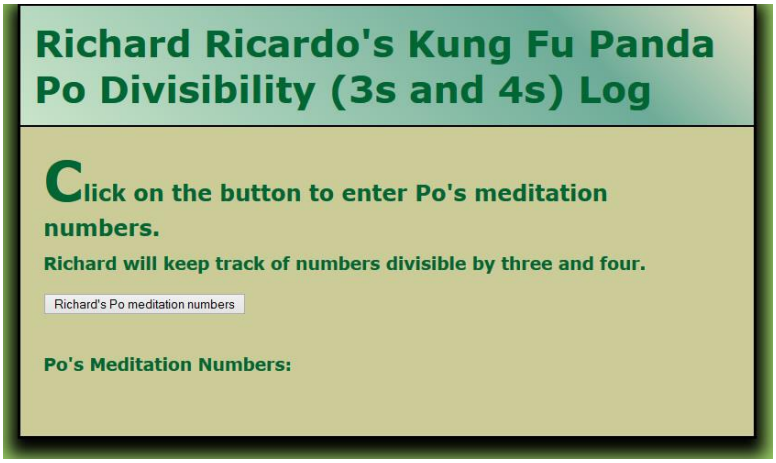
Question 1 – JavaScript Chapter 5 (33 points) **Estimated time: 3 hours**

- Save question 1 files in subfolder "**lastname_firstname_assignment6\q1**\". (-25 points if no JavaScript)
- Create a web page that displays the default information (example shown). (1 point each, total 3 points)
- The initial page and related outputs should look like the examples shown below.
- Create your page using "**<your name>**'s Kung Fu Panda Divisibility Log**"** as the page title. Save the page as **index.htm**. Remember to document the html file with html comments. (1 point each, total 3 points)
- Write a program that asks the user for Po's meditation numbers.
    - The user should be allowed to enter as many data set as desired (or enter -999 to quit). (4 points)
    - You must use **do...while** or **while** loop. (-15 points if not)
    - The output should look like the example output shown. (1 point each, total 3 points)
    - Indicate if an entered number is **divisible by three** or **divisible by four**. (6 points)

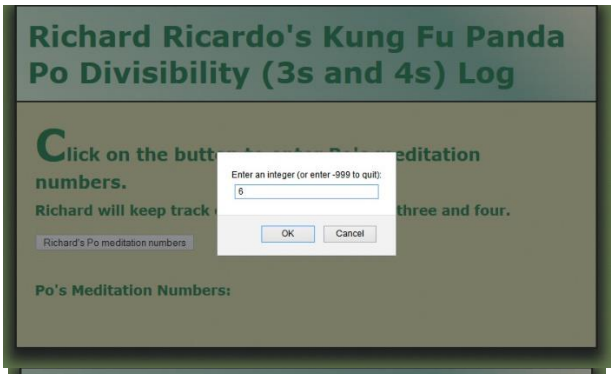
- The program should output the number of **numbers divisible by three**, their **sum** and **average**. (6 points)
- The program should output the number of **numbers divisible by four**, their **sum** and **average**. (6 points)

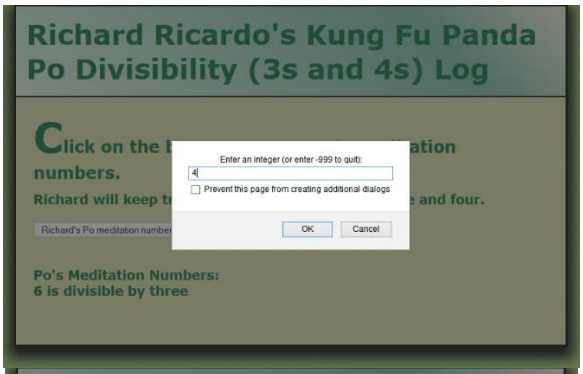
- Create a css file named **style.css** to format index.htm by creating your own layout (no two students should have the same layout). Use css comments to document the css program. You can use **the same (or similar) css file(s)** to format all questions. (1 point each, total 2 points)

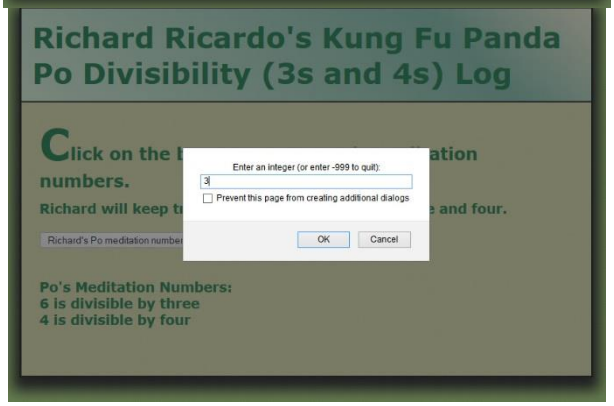
Example: Initial Page

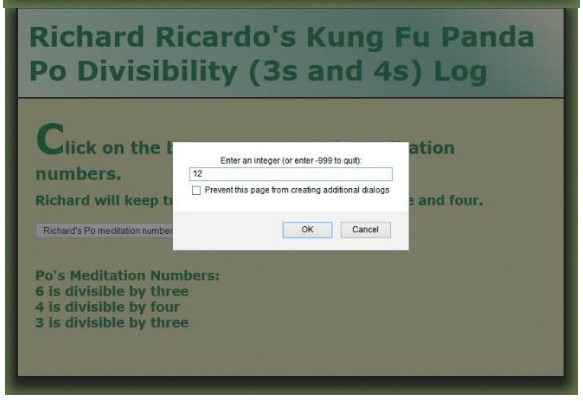


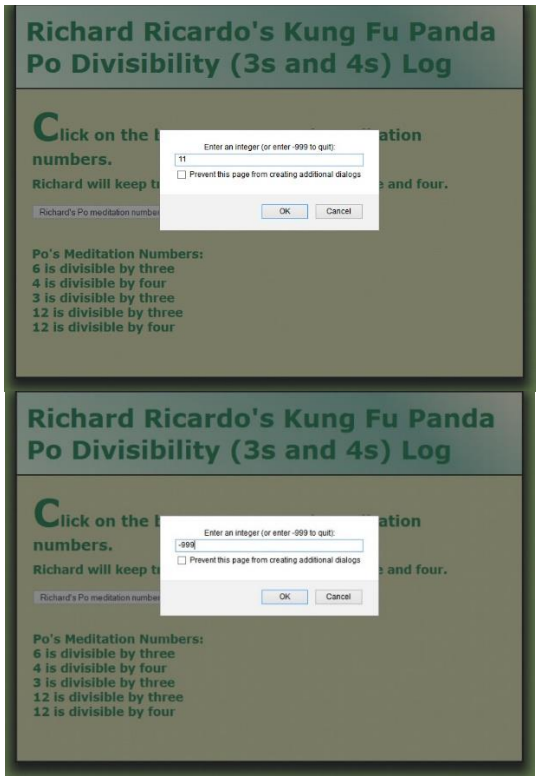
Example: Input

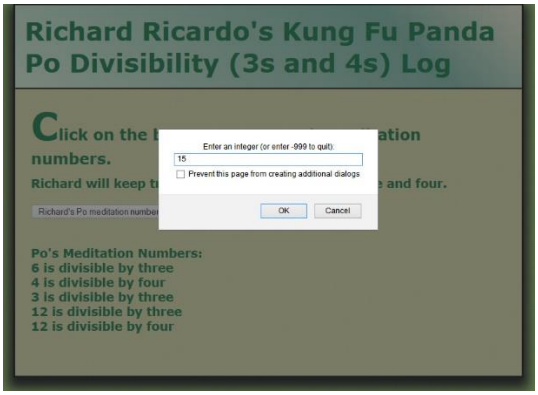




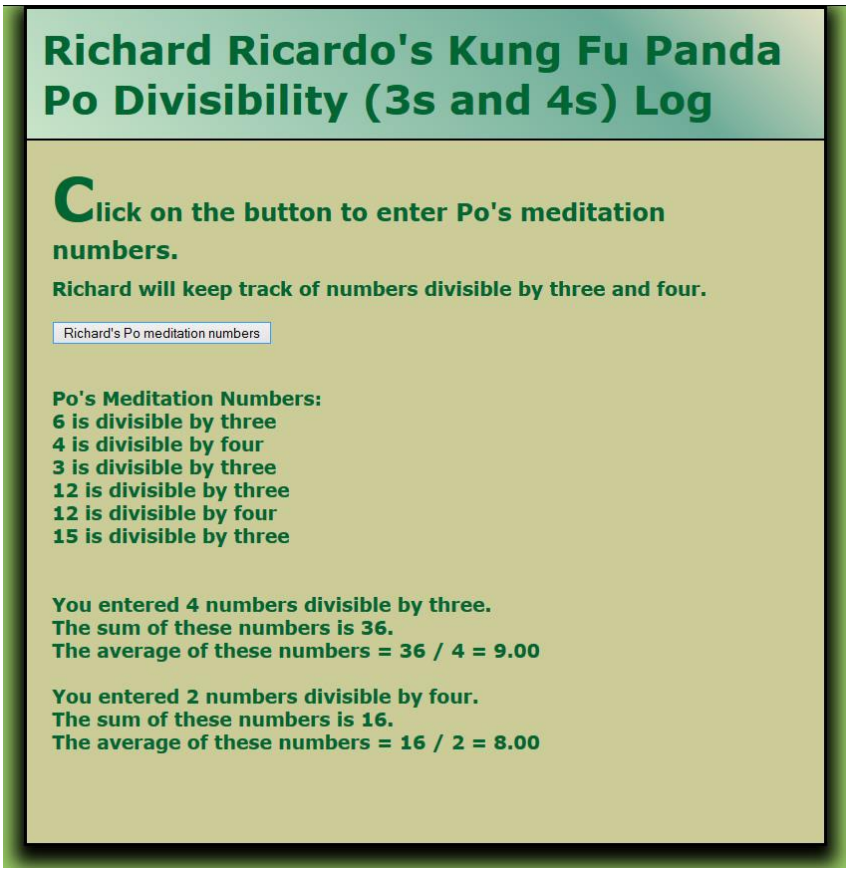








Example: Output



Question 2 – JavaScript Chapter 5 & 6 (33 points) **Estimated time: 3 hours**

- Save question 2 files in subfolder "**lastname_firstname_assignment6\q2**\". (-25 points if no JavaScript)
- Create a web page that displays the default information (example shown). (1 point each, total 3 points)
- The initial page and related outputs should look like the examples shown below.
- Create your page using "**<your name>**'s Kung Fu Panda Po Magic Rectangle**"** as the page title. Save the page as **index.htm**. Remember to document the html file with html comments. (1 point each, total 3 points)
- Write a program that asks the user for Po's row and column numbers for generating magic rectangles.
  - Use **html text fields** (`<input type="text" … >`) to capture row and column numbers. (4 points)
  - Generate the magic rectangles shown using **nested loops** (**two-dimensional loops**).
  - The first rectangle must be generated by **nested while** loops.

```
while (…)
{
      while (…) {}
}
```
(6 points for logic, 4 points for output formatting)

  - The second rectangle must be generated by **nested do…while** loops.

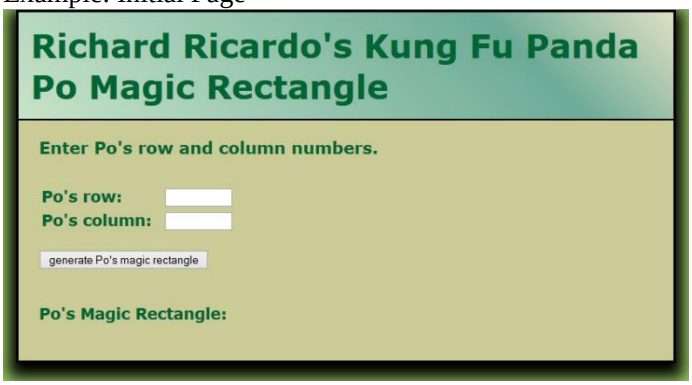
```
do
{
      do {} while (…)
}
```
(6 points for logic, 4 points for output formatting)

  - The third rectangle and related **calculation** must be generated by **nested for** loops.

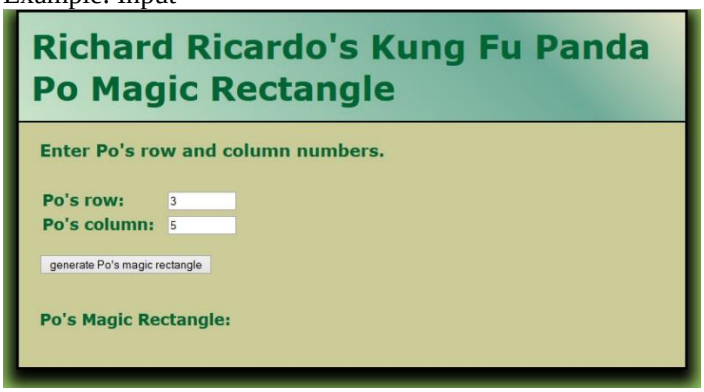
```
for (…)
{
      for (…) {}
}
```
(6 points for logic, 4 points for output formatting)

- Create a css file named **style.css** to format index.htm by creating your own layout (no two students should have the same layout). Use css comments to document the css program. (1 point each, total 2 points)

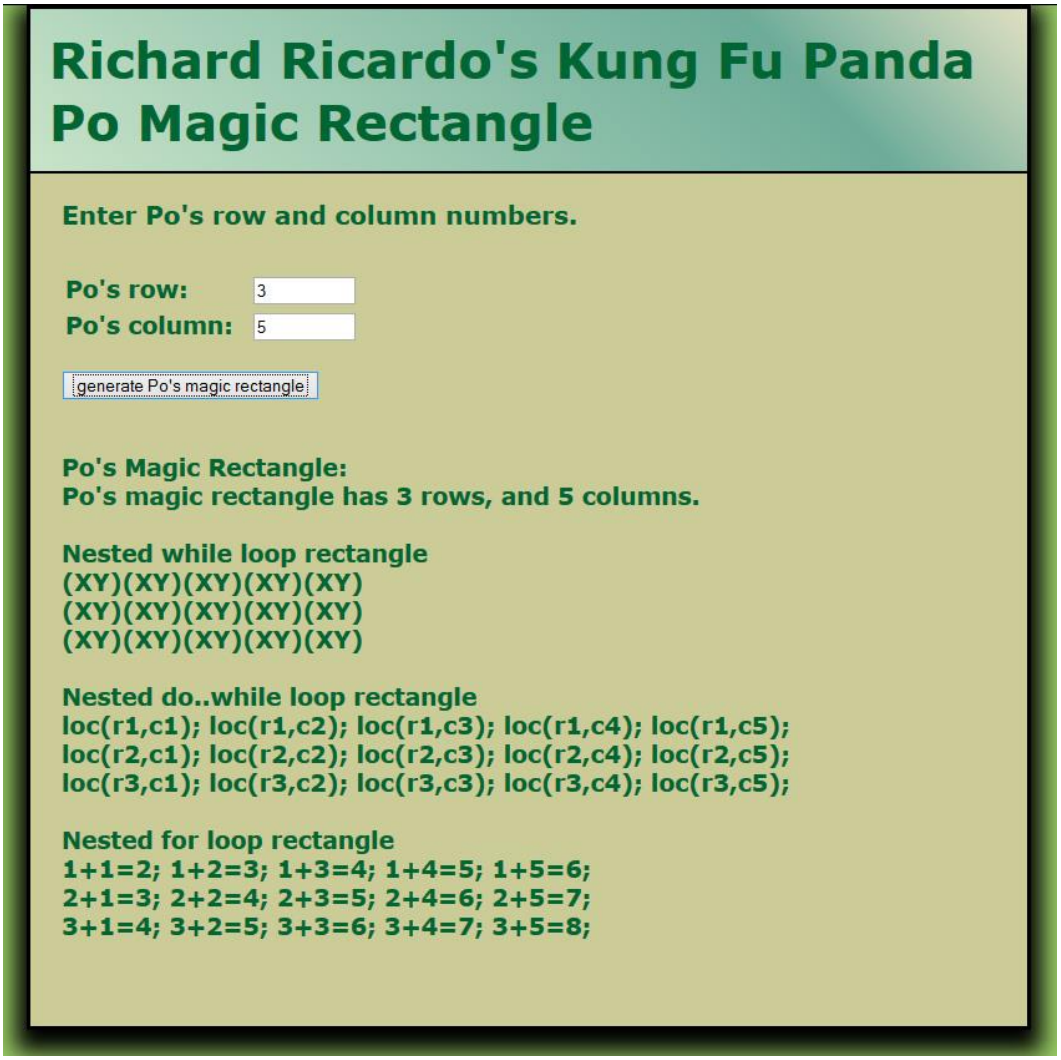
Example: Initial Page



Example: Input



Example: Output

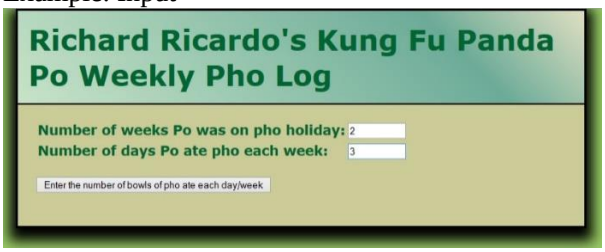


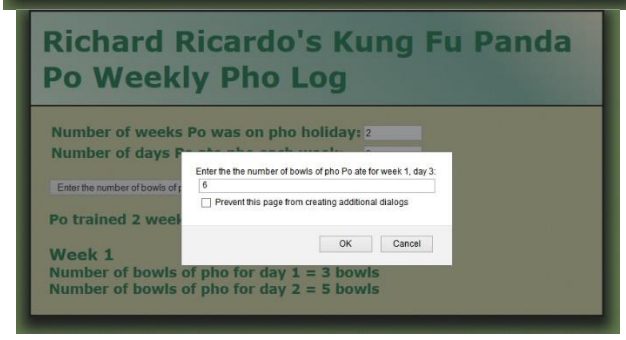
Question 3 – JavaScript Chapter 5 & 6 (34 points) **Estimated time: 3 hours**

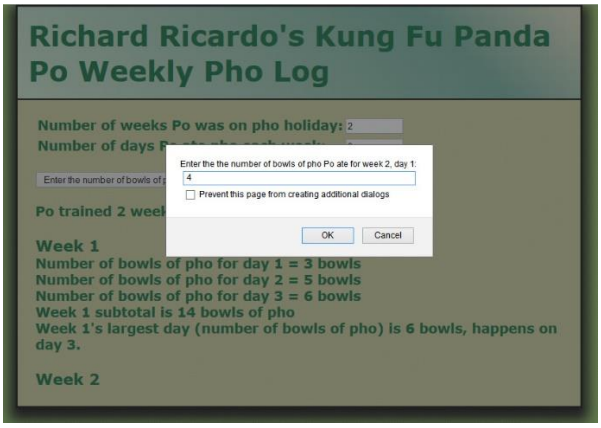
- Save question 3 files in subfolder "**lastname_firstname_assignment6\q3**". (-25 points if no JavaScript)
- Create a web page that displays the default information (example shown). (1 point each, total 2 points)
- The initial page and related outputs should look like the examples shown below.
- Create your page using "**<your name>**'s Kung Fu Panda Po Weekly Pho Log**"** as the page title. Save the page as **index.htm**. Remember to document the html file with html comments. (1 point each, total 3 points)
- Write a program that allows the user to enter the number of bowls of pho Po ate for each day and each week.
    - Use **html text fields** (`<input type="text" … >`) to capture the **number of weeks** and the **number of days** per week. (4 points)
    - The user should be allowed to enter the number of bowls of pho for each day of each week.
    - The output should look like the example **output shown**. (1 point each, total 3 points)
    - The program should also calculate Po's **total** number of bowls of pho, **subtotal** number of bowls of pho for each week, and the **day** with the **largest** number of bowls of pho for **each week**. Your output should follow the example output format given. (3 points each, total 12 points)
    - You must use **nested loops** (**for,** or **while,** or **do..while** or **mixing them**) to solve the problem. (-15 points if not)
- Create a css file named **style.css** to format index.htm by creating your own layout (no two students should have the same layout). Use css comments to document the css program. (1 point each, total 2 points)

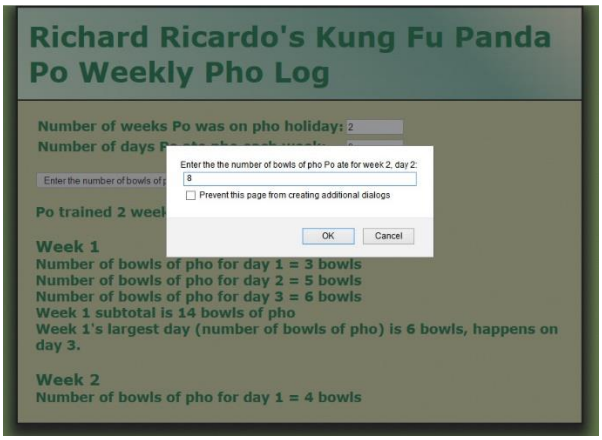
Example: Initial Page

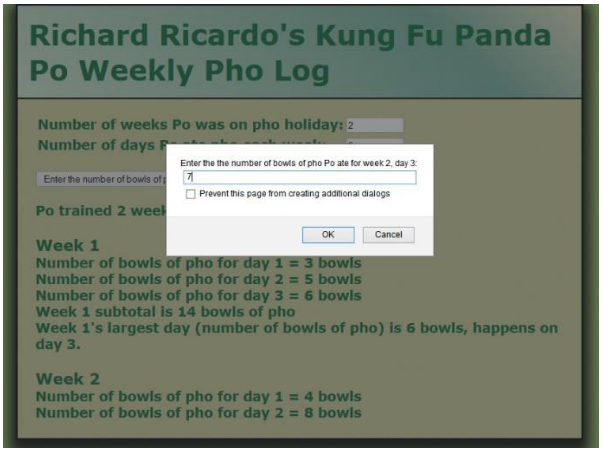
Example: Input



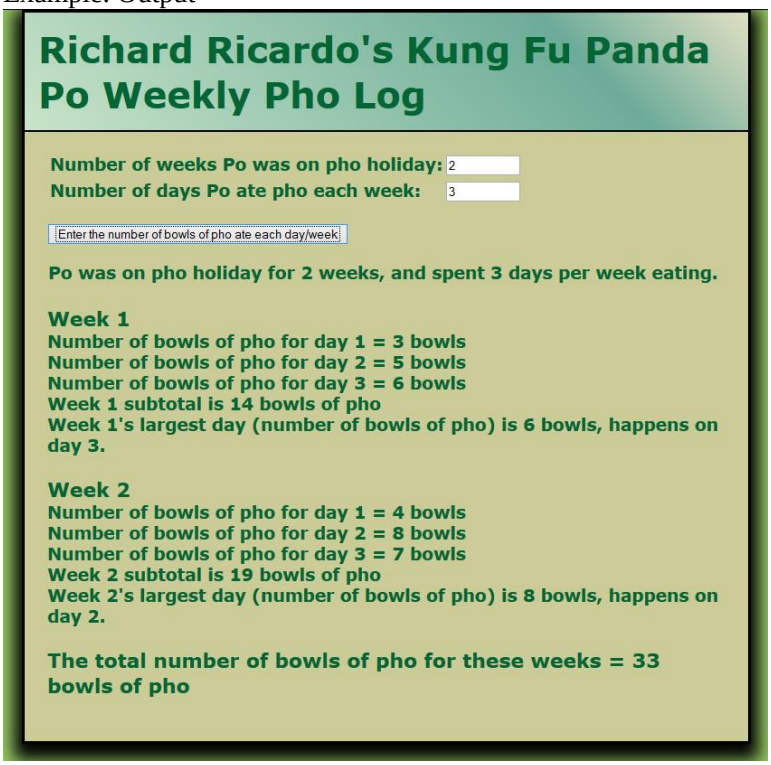








Example: Output



Important:

1. If you do not put **<your name> / <your first name>** in the above mentioned fields (as shown in the examples), you will get **0 points** for the question(s).
2. **No two students** should submit webpages with exactly the same cod, or the content, or same layout, or same color combination. If found, both students will get **0 points**.
3. When you view page source in a web browser, **`<!DOCTYPE html>`** must be at the top of every page. In other words, all pages must be written in HTML5. **(-20 points** if not)
4. All html files must pass html validation at http://validator.w3.org/ without any **error/warning** (with only 2 warnings). Use the validator's "File Upload" tab to check each file. **(-2 points** for each error/warning, **2 warnings allowed**)
5. All css files must pass css validation at http://jigsaw.w3.org/css-validator/ without any **error/warning**. **(-2 points** for each error/warning)
6. If your html file contains any css component, your html file must pass both html validation (3 above), and css validation (4 above) without any error.
7. If your files do not pass the validations, **2 points will be deducted** for **each error** (and each JavaScript warning) found.
8. Document (comment) your html files (`<!-- -->`), css files (`/* */`), and JavaScript files (`/* */`). **(-1 point** for each file with insufficient comments)

Submission instructions:

- You need to test the above document(s) in your web browser.
- Do screen capture(s) of the **initial page** and the related **output(s)**. Use any graphic editing software (e.g. Microsoft Paint, Adobe Fireworks, GIMP, or Microsoft Expression Design etc) to cut out the browser output (from the screen capture), paste them into a word document.
- For this assignment, you only need to do screen capture(s) of the **initial page** and the related **output(s)**, you do not need to do screen capture(s) of the input pages. Provide **2 different test cases** for each question. In other words, for **each question**, you need to have **1 initial page** screen capture and **2 related output(s)** screen captures. (-50 points for no test cases, -5 points for only 1 test case)
- Do screen capture(s) of html validation results and css validation results, cut and paste them into the word document. You do not need to do screen capture(s) of JavaScript validation results.
- Save the word document as a pdf file.

You need to submit the following:
1. A pdf file containing the screen capture(s) of the web browser output (all html pages) and the screen capture(s) of all html validation results (from http://validator.w3.org/) and css validation results (from http://jigsaw.w3.org/css-validator/), name the file **lastname_firstname_assignment06.pdf**.
2. All html file(s), css file(s), and other related files (e.g. image files). Zip your file folder (lastname_firstname_assignment6) into a single zip file (or rar file) **lastname_firstname_assignment06.zip**.
In the above example, the zip file should contain the following files and subfolders. If there is any image, there should be a \images\ subfolder.

- lastname_firstname_assignment6\q1\index.htm
- lastname_firstname_assignment6\q1\style.css
- lastname_firstname_assignment6\q2\index.htm
- lastname_firstname_assignment6\q2\style.css // you may put style.css in a subfolder
- lastname_firstname_assignment6\q3\index.htm
- lastname_firstname_assignment6\q3\style.css

Please submit an electronic copy (the above mentioned **two files**: .pdf and .zip) to D2L digital dropbox.

Grading guidelines (programming questions):
Your programs will be judged on several criteria, which are shown below.

- Correctness (50%): Does the program compile (run) correctly? Does the program do what it's supposed to do?
- Design (20%): Are operations broken down in a reasonable way (e.g. classes and methods)?
- Style (10%): Is the program **indented** properly? Do variables have **meaningful names**?
- Robustness (10%): Does the program handle erroneous or unexpected input gracefully?
- Documentation (10%): Do all program files begin with a **comment** that identifies the author, the course code, and the program date? Are all the classes, methods and data fields clearly **documented (commented)?** Are unclear parts of code **documented (commented)?** (Some items mentioned may not apply to some languages)

A program that does not compile (run) will get at most **50% of the possible points**.