CIS 1152 - Lab #2 PHP Logic and Loops
S. Ruegsegger
Modified by Peter Chapin (with permission)

Objective

To explore the use of PHP logic (Boolean) conditionals and loops.

Task 1: Multiplication table

Student skill: *nested* loops

Use a nested loop to build a multiplication table in HTML of size n *x* n; where a single PHP variable $n determines the (square) size. For example, here are two tables.

This is a **5 x 5** multiplication table.

x	1	2	3	4	5
1	1	2	3	4	5
2	2	4	6	8	10
3	3	6	9	12	15
4	4	8	12	16	20
5	5	10	15	20	25

This is a **10 x 10** multiplication table.

x	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10
2	2	4	6	8	10	12	14	16	18	20
3	3	6	9	12	15	18	21	24	27	30
4	4	8	12	16	20	24	28	32	36	40
5	5	10	15	20	25	30	35	40	45	50
6	6	12	18	24	30	36	42	48	54	60
7	7	14	21	28	35	42	49	56	63	70
8	8	16	24	32	40	48	56	64	72	80
9	9	18	27	36	45	54	63	72	81	90
10	10	20	30	40	50	60	70	80	90	100

You may be creative with CSS or other formatting. Here are 4 <u>requirements</u>:
- There is a first column of *row-headers* and a first row of *column-headers*
- Row and column headers have a different color background.
- Row and column header text is **bold**.
- Color the row=column diagonal cells with a different, highlighting background color.

Task 2: Collatz Sequence

Student skill: *while* loops (unknown sequence ending) with if-then-else logic.

What is the Collatz Sequence? Read this: https://en.wikipedia.org/wiki/Collatz_conjecture

This is an algorithm which can start with *any* (very large) integer and it <u>always</u>, and eventually (and rather quickly), ends at the number "1". I find that amazing! I love it.
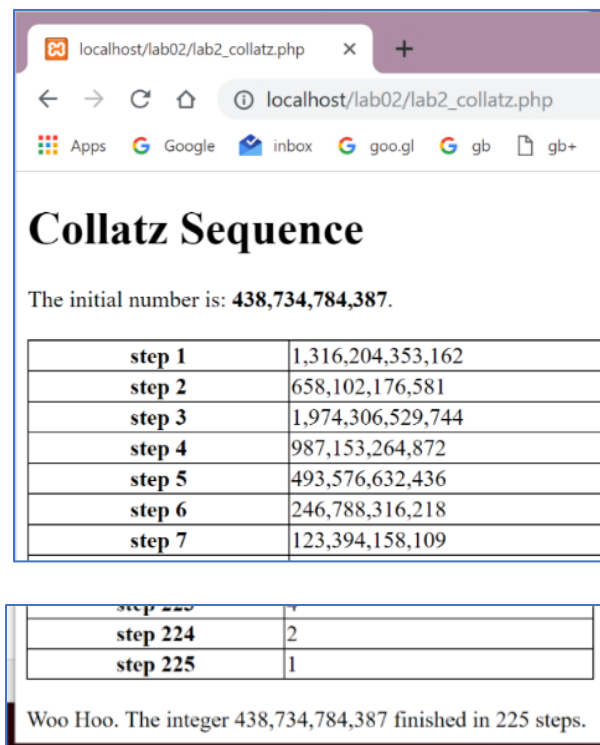
Here is the algorithm:
- If the number is even, divide it by 2
- If the number is odd, then return (3 * number + 1)
- Keep doing this loop until the number is 1.

Notice that evens get <u>smaller</u>, but odds get <u>bigger</u>!

<u>Requirements</u>:
- Print out the initial, **very large** number – which is at least **12 digits** long.
- Print all numbers in a readable "comma" format.
- Make a two-column table where the first column is the step number and the second column is the resultant sequential number of the algorithm.
- Finally, after the table is ended, print out the number of steps.