Assignment 1 - Introduction to Server Side Programming

Due Sep 29, 2023 by 11:59p.m. **Points** 100 **Submitting** a file upload **File Types** txt
Available until Oct 2, 2023 at 11:59p.m.

This assignment was locked Oct 2, 2023 at 11:59p.m..

Please note that all assignments must be submitted both to CSUNIX and also to Canvas. Except for the filename difference noted below, your files in both submissions must be identical or your work will be graded 0.

Please follow the submission instructions carefully to prevent a grade of 0 for incomplete submission.

For this course programs like ChatGPT or Co-pilot will be treated exactly the same as code copied from an internet source. Any evidence of their use will be treated as a level 2 academic integrity violation.

Purpose

The goal of this assignment is to get our heads back into computer aided problem solving with some fairly routine problems and to engage with the client-server model. The problems are chosen to engage loops, conditional statements, math functions, and string manipulation. If you find them difficult I recommend that you practice general programming skills as these problems should be relatively simple for 3rd semester.

Starter Code

Please download the starter code for assignment 1 **from this link (https://mycanvas.mohawkcollege.ca/courses/92989/files/17296059?wrap=1)** **(https://mycanvas.mohawkcollege.ca/courses/92989/files/17296059/download?download_frd=1)** . As indicated previously, we aren't assessing your abilities in producing client side code this term, and we want you to engage fully with the er side code. As such you are forbidden from altering the front end code. If we download your solutions and put it with the starter code file, it must work without alteration. To help you understand whether this is just some academic nonsense or if it's a real world restriction that could occur: during team programming one of the first steps is to agree on an interface which allows the teams to work in parallel without frequent interactions. Neither team is permitted to alter the interface once it is created. If a change is required, it must come from the project manager/team lead, which in this setting is your professor. This is a pattern that occurs in many kinds of programming and allows for teams to build software in parallel much quicker than an individual could generate the code. It also allows for the creation of standard tests at the time the specification is written so that we can validate that the created code meets the requirements. If we are going to strive to help you improve your skills as you move through the program with the goal of a professional level of practice, it feels reasonable to set up a restriction such as this.

Getting Started

Please review the **documentation standards (https://mycanvas.mohawkcollege.ca/courses/92989/pages/documentation-standards)** for the homework in this course.

Download 10260-a1.html starter code and put it in your /xampp/10260/a1 folder.

Create a new file called **me.php** that outputs your name and student number and place it in the same folder.

Question 1

Bank note collectors sometimes look at the serial number of the bills to see if it is a particularly interesting, rare combination of numbers. You may assume that the user will input serial numbers that is formatted as 3 letters followed by 7 numbers. You will process the serial number and state if it falls into any of the following categories:

- radar note - the numeric portion of the serial number forms a palindrome (reads the same backwards and forwards). For example: 1112111, 1223221, 1234321. If the radar note has only 1 digit (e.g. 1111111) call it a "solid serial number" instead.
- ladder note - the numeric portion of the serial number is a series of strictly increasing consecutive numbers (ladder up, e.g. 3456789), a series of strictly decreasing consecutive numbers (ladder down, e.g. 8765432), or a series of numbers that begins increasing and then changes to decreasing in the middle digit (ladder up-down 4567654) or vice versa (ladder down-up 7654567). Be sure to specify the specific pattern matched.
- rotator note - the numeric portion o f the serial number can be read the same upside down and right side up. For example: 0681890. The digits that can rotate are 0, 1, 6, 8, and 9 - you may ignore minor font based irregularities for this problem.
- binary note - if all the digits of the bank note are either 0 or 1 call it a binary note.

Create a function for each test that accepts the numeric portion of the serial number as input and returns a string containing any matched patterns. If the pattern is not matched return Boolean false. Output the list of all matching serial number groups as list items (<li></li>) as the calling AJAX will be placing your list items into an existing unordered list. If none of the categories are matched then classify the note as "uninteresting serial number".

At the start of your main code include the line `$serial_number = $_GET['serial_number'];` which will give you the user's input serial number.

Don't worry about data validation for this assignment.

Name your solution `group1-1.php`

Question 2

Generate the rows and columns of an HTML table with 2 columns using PHP. In the first column we'll be creating something like a pyramid. In row 1 place 1 "1". In row 2 place 2 "2"'s and so on, as shown below. In the second column put the sum of the numbers in column 1 of the same row.

Generate each row in a function that accepts the row number as an input and returns a string. Choose an appropriate name for your function.

At the start of your main code include the line `$rows = $_GET['rows'];` This will tell you how many rows of output to generate. Don't worry about data validation for this assignment.

Other than the code for the rows and cells of the table, do not output any HTML code - remember that your PHP program will be responding to an AJAX request.

Name your solution `group2-1.php`

1	1
22	4
333	9
4444	16

Submission Instructions

Programs are meant to be run, not read. Therefore in this course you must submit your work to CSUNIX so that your instructor can grade the functional performance of your work. This assignment must be uploaded to your **public_html/private/10260/a1** directory on CSUNIX (you will probably have to create this directory). If you submit this to the wrong folder it will be treated as though it was not submitted. Searching around your directory structure to find the folder is not a good use of your professor's assignment grading time. If your professor cannot find your work on CSUNIX you will be given 1 week to correct the folder name, and to email them to alert them that you have fixed the problem. After the 1 week period the grade of 0 will be permanent.

Your program must also be submitted to Canvas. First and foremost this establishes a time that you submitted your work that we can all agree on. Secondly it allows your work to be inspected for plagiarism. You will not be judged based on the simple score from TurnItIn, but on your professor's interpretation of whether the identified code is common and reasonably should appear in many student's work (i.e. **<?php** will show up a lot) or whether the code suggests copying from an unauthorized source. We are suggesting you shouldn't fully trust AI tools, and we won't either.

You will submit your files to Canvas in the following way:

For each .php file (not the .html starter code):

- Copy the file and add .txt to the end of the filename. For example: question1.php gets copied to question1.php.txt
- Upload the .txt files to Canvas

Do not archive (zip) your files, or upload files that are not requested.

You should be submitting **me.php.txt**, **group1-1.php.txt**, and **group2-1.php.txt** to Canvas and uploading the HTML starter code and your 3 .php files to CSUNIX.

Assignment 1 Rubric

Criteria	Ratings				Pts
Submitted to CSUNIX account Student has submitted assignment to CSUNIX in their "private/10260/a1" directory.	**0 pts** **NO** I attempted to grade your work but could not find it in the correct directory on CSUNIX. I am willing to regrade the work if you can put it in the correct location and email me to notify me that you have done so within 7 days of this feedback, otherwise the 0 will be permanent. Be forewarned that except for the extension of the file name (.php.txt vs .php) the file on CSUNIX must be identical to the one on CANVAS or you will be cited for academic dishonesty.	**0 pts** **Yes**			0 pts
Starter HTML is unmodified If the student has altered the front end code in any way, a penalty will be applied for not following the requirements of the assignment.	**0 pts** **Modified** Apply penalty of -10 to assignment. This is in addition to any 0's that will be applied for missing functionality.	**0 pts** **Unmodified**			0 pts
me.php is present and functional me.php must output the student's name and student number. It must contain a statement of authorship, @author, @version, and @package tags.	**10 pts** **Meets all requirements** me.php exists in the same directory as the starter HTML file. It outputs the student's name and student number. The comment block is present and contains the statement of authorship, as well as the phpdoc tags @author, @version, and @package with suitable values.	**7.5 pts** **Meets some requirements** me.php exists in the same directory as the starter HTML file. It outputs the student's name and student number. The comment block is present and contains the statement of authorship. The phpdoc tags @author, @version, and @package are incomplete, or do not have suitable values.	**4.5 pts** **Functional requirements not present** me.php exists in the same directory as the starter HTML file but does not output the student's name and student number. The comment block exists and contains at least the statement of authorship.	**0 pts** **Requirements not met** Some or all of the following criteria is true: - me.php does not exist - me.php is not in the same directory as the starter HTML file - the statement of authorship is missing	10 pts
Question 1 Nonfunctional Elements File is named group1-1.php, located in the same directory as the starter HTML file, contains a comment block that includes the statement of authorship and the phpdoc tags @author, @version, and	**7.5 pts** **Meets all requirements** File is named group1-1.php, located in the same directory as the starter HTML file, contains a comment block that includes the statement of authorship and the phpdoc tags @author, @version, and @package with suitable values. All functions (if any) are commented with phpdoc @param and @return tags that	**5 pts** **Meets some requirements** File is named group1-1.php, located in the same directory as the starter HTML file, and contains a comment block that includes the statement of authorship and the phpdoc tags @author, @version, and @package are present but may not have suitable values. Functions (if any) are not all commented with phpdoc	**3 pts** **Needs Improvement** File is named group1-1.php, located in the same directory as the starter HTML file, and contains a comment block that includes the statement of authorship. PHPDOC may be missing or incomplete.	**0 pts** **Requirements not met** Some or all of the following criteria is true: - group1-1.php does not exist - group1-1.php is not in the same directory as the starter HTML file - the statement of authorship is missing	7.5 pts

Criteria	Ratings				Pts
@package with suitable values. All functions (if any) are commented with phpdoc @param and @return tags that describe the function's input(s) and output(s).	describe the function's input(s) and output(s).	@param and @return tags that describe the function's input(s) and output(s), or the tags contain incomplete values.			
Question 1 Inputs Solution works with the specified format of @@@####### where @ is an alpha character and # is a numeric digit. The solution in some way manages separating the alpha and numeric portions for processing (string manipulation).	**7.5 pts** **Meets all requirements** Solution works with the specified format of @@@####### where @ is an alpha character and # is a numeric digit. The solution in some way manages separating the alpha and numeric portions for processing.	**3.75 pts** **Meets some requirements** Solution changes the assumed specified format of @@@####### where @ is an alpha character and # is a numeric digit to something else to avoid string manipulation requirements.	**0 pts** **Requirements not met** Solution does not correctly process the input from the starter HTML file and instead relies on hardcoded values, or is non-functional in some other manner.		7.5 pts
Question 1 Modularity The solution breaks processing of radar/ladder/rotator notes into modules in a manner that is consistent with program expectations; at least 3 functions are present and each has a clear, logical purpose. Each function accepts the numeric portion of the serial number only and returns	**15 pts** **Meets all requirements** The solution breaks processing of radar/ladder/rotator notes into modules in a manner that is consistent with program expectations. Each function accepts the numeric portion of the serial number only and returns either a string or Boolean false.	**10 pts** **Meets some requirements** The solution breaks processing of radar/ladder/rotator notes into at least 1 module separate from the creation of the final output. The function accepts the the serial number in some format and returns at least a string but may be missing the Boolean false value.	**7.5 pts** **Requirements minimally met** The solution breaks processing of radar/ladder/rotator notes into at least 1 module. The function's input and output parameters may be missing: i.e. it relies on a global variable and/or produces output using print/echo statements.	**0 pts** **Requirements not met** The solution does not break the processing steps into individual functions or the functions provided are non-function/trivial.	15 pts

Criteria	Ratings				Pts
either a string or Boolean false.					
Question 1 Functionality The solid, radar, ladder up, ladder down, ladder up-down, ladder down-up, rotator, and binary note patterns are all present and functional for 7 digit serial numbers. The pattern identifications are not mutually exclusive except solid/radar, ladder-up/ladder-down/ladder up-down/ladder down-up. The pattern identifications are expressed as strings that are parsed as correct HTML list item elements (e.g. "<li>pattern</li>", with one pattern per list item. If none of the patterns are met the message "uninteresting serial number" is returned instead.	**30 pts** **Meets all requirements** The solid, radar, ladder up, ladder down, ladder up-down, ladder down-up, rotator, and binary note patterns are all present and functional for 7 digit serial numbers. The pattern identifications are not mutually exclusive except solid/radar, ladder-up/ladder-down/ladder up-down/ladder down-up. The pattern identifications are expressed as strings that are parsed as correct HTML list item elements (e.g. "<li>pattern</li>", with one pattern per list item. If none of the patterns are met the message "uninteresting serial number" is returned instead.	**22.5 pts** **Meets some requirements** At least 5 of the solid, radar, ladder up, ladder down, ladder up-down, ladder down-up, rotator, and binary note patterns are present and functional for 7 digit serial numbers. The pattern identifications are not mutually exclusive except solid/radar, ladder-up/ladder-down/ladder up-down/ladder down-up. The pattern identifications are expressed as strings that are parsed as correct HTML list item elements (e.g. "<li>pattern</li>", with one pattern per list item. If none of the patterns are met an appropriate failure message is returned instead.	**15 pts** **Requirements minimally met** Some of the solid, radar, ladder up, ladder down, ladder up-down, ladder down-up, rotator, and binary note patterns are functional for 7 digit serial numbers. The pattern identifications may not correctly be identified as mutually exclusive. The pattern identifications are expressed as strings but might not be parsed as correct HTML list item elements (e.g. "<li>pattern</li>", with one pattern per list item. If none of the patterns are met either no output or an appropriate failure message is returned instead.	**0 pts** **Requirements not met** Code provided is substantially ineffective, including inappropriate use of regular expressions, etc. No meaningful success towards completing more than 1 pattern match is demonstrated.	30 pts
Question 2 Nonfunctional Elements File is named group2-1.php, located in the same directory as the starter HTML	**5 pts** **Meets all requirements** File is named group2-1.php, located in the same directory as the starter HTML file, contains a comment block that includes the	**3.75 pts** **Meets some requirements** File is named group2-1.php, located in the same directory as the starter HTML file, and contains a comment block that	**2.5 pts** **Needs Improvement** File is named group2-1.php, located in the same directory as the starter HTML file,	**0 pts** **Requirements not met** Some or all of the following criteria is true: - group2-1.php does not exist - group2-1.php is not	5 pts

Criteria	Ratings					Pts
file, contains a comment block that includes the statement of authorship and the phpdoc tags @author, @version, and @package with suitable values. All functions (if any) are commented with phpdoc @param and @return tags that describe the function's input(s) and output(s).	statement of authorship and the phpdoc tags @author, @version, and @package with suitable values. All functions (if any) are commented with phpdoc @param and @return tags that describe the function's input(s) and output(s).	includes the statement of authorship and the phpdoc tags @author, @version, and @package are present but may not have suitable values. Functions (if any) are not all commented with phpdoc @param and @return tags that describe the function's input(s) and output(s), or the tags contain incomplete values.	and contains a comment block that includes the statement of authorship. PHPDOC may be missing or incomplete.	in the same directory as the starter HTML file - the statement of authorship is missing		
Question 2 Inputs Solution works with the specified input of a single positive integer.	**2.5 pts** **Meets all requirements** Solution works with the specified input of a single positive integer.	**0 pts** **Requirements not met** Solution does not correctly process the input from the starter HTML file and instead relies on hardcoded values, or is non-functional in some other manner.				2.5 pts
Question 2 Modularity The solution breaks processing of each row into a function. The function accepts the integer row number and returns the row as a valid HTML string.	**7.5 pts** **Meets all requirements** The solution breaks processing of each row into a function. The function accepts the integer row number and returns the row as a valid HTML string.	**4 pts** **Meets some requirements** The solution breaks processing of the required output into a function by some means. The function accepts the integer row number and returns the row as a valid HTML string.	**2.5 pts** **Requirements minimally met** The solution breaks processing of the required output into a function by some means. The function's input and output parameters may be missing: i.e. it relies on a global variable and/or produces output using print/echo statements.	**0 pts** **Requirements not met** The solution does not break the processing steps into individual functions or the functions provided are non-function/trivial.		7.5 pts
Question 2 Functionality The program creates the number of HTML rows (tr element) containing two columns (td elements) specified by the nput. For each row the first column contains the input	**15 pts** **Meets all requirements** The program creates the number of HTML rows (tr element) containing two columns (td elements) specified by the nput. For	**12.5 pts** **Meets most requirements** The program creates the number of HTML rows (tr element) containing two columns (td elements) specified by the nput. For	**7.5 pts** **Meets some requirements** The program creates some sort of HTML table output output related to the input but and the format/implementation of the solution is substantially correct, but might have slight issues.	**4.5 pts** **Requirements minimally met** The program creates some HTML output related to the input but despite some effort being apparent the format/implementation of the solution may be incorrect.	**0 pts** **Requirements not met** Code provided is substantially ineffective. No meaningful success towards the solution is demonstrated.	15 pts

Criteria	Ratings					Pts
row repeated that many times (e.g. 1 "1", 2 "2"'s, ..., 10 "10"'s, etc) in a string. The second column contains the sum of the repeated number (not digits, e.g. 1x1, 2x2, 10x10). The output string does not contain any table element tags (<table>, </table>).	each row the first column contains the input row repeated that many times (e.g. 1 "1", 2 "2"'s, ..., 10 "10"'s, etc) in a string. The second column contains the sum of the repeated number (not digits, e.g. 1x1, 2x2, 10x10). The output string does not contain any table element tags (<table>, </table>).	each row the first column contains the input row repeated that many times (e.g. 1 "1", 2 "2"'s, ..., 10 "10"'s, etc) in a string. The second column contains the sum of the repeated number (not digits, e.g. 1x1, 2x2, 10x10). Extra HTML table element tags might be present.				
						Total Points: 100