Submission Worksheet

Course: IT202-007-F2024
Assigment: [IT202] Module 2 PHP Problems
Student: Alanor S. (ass89)

Submissions:

Submission Selection

1 Submission [submitted] 9/23/2024 11:21:40 AM

Instructions

^ COLLAPSE ^

Overview video: https://youtu.be/Vvx_cfdLqqM

Guide:

1. Make sure you're in the dev branch locally and `git pull origin dev` any pending changes.
2. Make a new branch per the recommended branch name below (`git checkout -b ...`).
3. Grab the template code from
   https://gist.github.com/MattToegel/48b48377eaa1937c886b7840c449750a.
4. Create individual PHP files for each problem and save the files inside your `public_html` folder in a subfolder of your choice.
   1. If you don't have this folder yet, refer to the setup lessons (you'll need a few files for the deployment to work).
5. Move the unedited template files to GitHub.
   1. `git add .`
   2. `git commit -m "adding template files"`
   3. `git push origin branch_name` (see below)
   4. Create and open a pull request from the homework branch to main (leave it open until later steps).
6. Note: As you work, it's recommended to add/commit at least after each solution is done (i.e., 3+ times in this case).
   1. Make sure the files are saved before doing this.
7. Fill in the items in the worksheet below (save as often as necessary).
8. Once finished, export the worksheet.
9. Add the output file to any location of your choice in your repository folder (i.e., a `Module2` folder).
10. Check that git sees it via `git status`.

11. If everything is good, continue to submit.
    1. Track the file(s) via `git add`.
    2. Commit the changes via `git commit` (don't forget the commit message).
    3. Push the changes to GitHub via `git push` (don't forget to refer to the proper branch).
    4. Create a pull request from the homework related branch to dev (i.e., dev <- "homework branch").
    5. Open and complete the merge of the pull request (it should turn purple).
    6. Locally checkout dev and pull the latest changes (to prepare for future work).
12. Take the same output file and upload it to Canvas.

Branch name: M2-PHP-Problems

Group

100%

Group: Problem 1
Tasks: 1
Points: 3

^ COLLAPSE ^

Task

100%

Group: Problem 1
Task #1: Problem 1 Evidence
Weight: ~100%
Points: ~3.00

^ COLLAPSE ^

ⓘ Details:

Only make edits where the template code mentions.

Solution should ensure that any passed in array will have only the odd values output.
Requires at least 2 screenshots (code + output from heroku dev)

Columns: 1

Sub-Task

100%

Group: Problem 1
Task #1: Problem 1 Evidence
Sub Task #1: Show the output from heroku dev (url must be visible)

🖼 Task Screenshots

Gallery Style: 2 Columns

4   2   1

prob 1 deployed from heroku

Caption(s) (required) ✓
Caption Hint: *Describe/highlight what's being shown*

🔗Task URLs

URL #1

https://asa89-it202-007-dev-bef7e2e4f18c.herokuapp.com/phpProbfolder/problem1.php

URL
https://asa89-it202-007-dev-bef7e2e4f18c.herol

Sub-Task 100%

Group: Problem 1
Task #1: Problem 1 Evidence
Sub Task #2: Show the code solution (ucid/date as comment must be present)
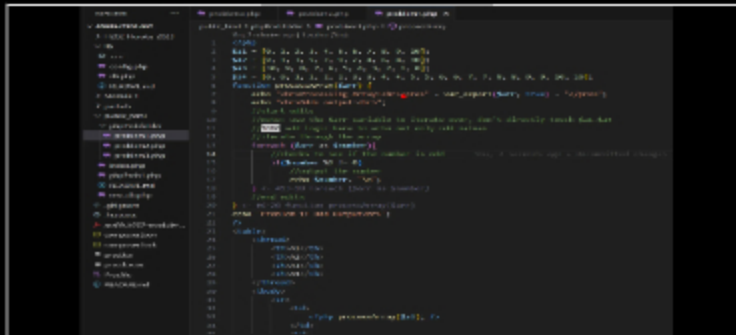

🖼 Task Screenshots

Gallery Style: 2 Columns

4   2   1

code evidence of prob 1

Caption(s) (required) ✓
Caption Hint: *Describe/highlight what's being shown*

≡ Task Response Prompt

Explain in concise steps how this logically works

Response:

First I iterate through the array. I check each element in the array, denoted as the variable $number, and check to see if it's odd (using modulus). If it is odd, it will print the number.

End of Task 1

End of Group: Problem 1
Task Status: 1/1

Task

100%

Group: Problem 2
Task #1: Problem 2 Evidence
Weight: ~100%
Points: ~3.00

^ COLLAPSE ^

ⓘ **Details:**

Only make edits where the template code mentions.

Solution should ensure that any passed in array will have the numerical values summed and rounded to two decimals (similar to currency)

Columns: 1

Sub-Task

100%

Group: Problem 2
Task #1: Problem 2 Evidence
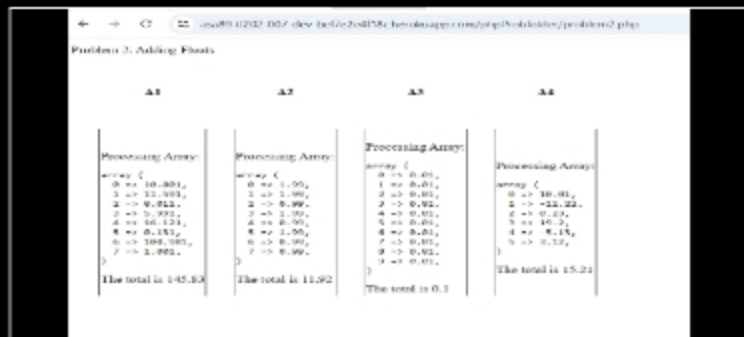Sub Task #1: Show the output from heroku dev (url must be visible)

🖼 Task Screenshots

Gallery Style: 2 Columns

4   2   1

prob 2 deployed from heroku

Caption(s) (required) ✓
Caption Hint: *Describe/highlight what's being shown*

🔗 Task URLs

URL #1
https://asa89-it202-007-dev-bef7e2e4f18c.herokuapp.com/phpProbfolder/problem2.php

URL
https://asa89-it202-007-dev-bef7e2e4f18c.hero|

Sub-Task

Group: Problem 2
Task #1: Problem 2 Evidence
Sub Task #2: Show the code solution (ucid/date as comment must be present)

🖼 Task Screenshots

Gallery Style: 2 Columns

4   2   1

code evidence of prob 2

Caption(s) (required) ✓
Caption Hint: *Describe/highlight what's being shown*

≡ Task Response Prompt

Explain in concise steps how this logically works

Response:

I iterate through the array with a foreach loop. It checks each element in the array. I notiiced a variable ($total) was initialized to 0 in the beginning. This variable will be the bucket to hold the total values. As it iterates through the loop, it adds the value of the element($number) to the total ($total). As for the rounding, I use the round function and make it round the total to two decimal points. It then prints the total as part of a string.

End of Task 1

End of Group: Problem 2
Task Status: 1/1

Group

100%

Group: Problem 3
Tasks: 1
Points: 3

^ COLLAPSE ^

Task

100%

Group: Problem 3
Task #1: Problem 3 Evidence
Weight: ~100%
Points: ~3.00

ⓘDetails:

Only make edits where the template code mentions.

Solution should ensure that any passed in array will have its values converted to a positive version of the value AND converted back to the original data type.

Columns: 1

Sub-Task

100%

Group: Problem 3
Task #1: Problem 3 Evidence
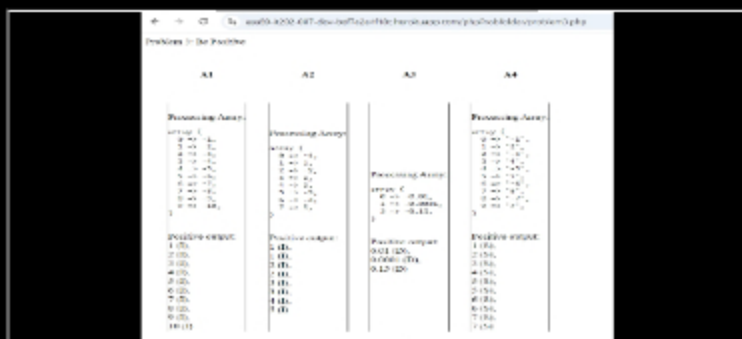Sub Task #1: Show the output from heroku dev (url must be visible)

🖼 Task Screenshots

Gallery Style: 2 Columns

4   2   1

prob 3 heroku deployment

Caption(s) (required) ✓
Caption Hint: *Describe/highlight what's being shown*

⑆Task URLs

URL #1
https://asa89-it202-007-dev-bef7e2e4f18c.herokuapp.com/phpProbfolder/problem3.php

URL
https://asa89-it202-007-dev-bef7e2e4f18c.hero|

Sub-Task
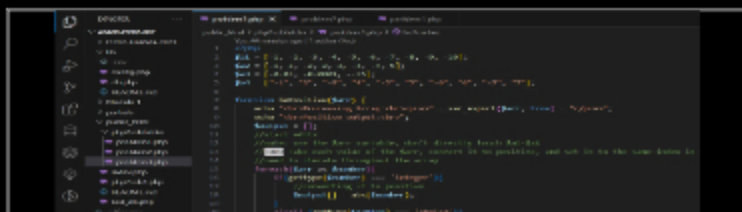
100%

Group: Problem 3
Task #1: Problem 3 Evidence
Sub Task #2: Show the code solution (ucid/date as comment must be present)

🖼 Task Screenshots

Gallery Style: 2 Columns

4   2   1

code evidence for prob 3

≡, Task Response Prompt

Explain in concise steps how this logically works

Response:

I start of with a loop once again to iterate through the array. It checks for the type of each element in the array. I have three cases set up, checking for integer, strings, and doubles. If the element registers as an integer, it is converted into a postive number using the abs function and assigned to an array variable (which I named output). This logic is followed for the other cases as well, except I make sure to cast the type into the output as well. For example, for string. I cast the output as a string ($output[] = (string)abs($number)), since the problem wanted the original data type to be displayed.

End of Task 1

End of Group: Problem 3
Task Status: 1/1

Group

100%

Group: Reflection
Tasks: 3
Points: 1

∧ COLLAPSE ∧

Task

100%

Group: Reflection
Task #1: Reflect on your experience
Weight: ~33%
Points: ~0.33

∧ COLLAPSE ∧

ⓘ Details:

Talk about any issues you had, how you resolved them, and anything you learned during this process.

Provide concrete details/examples. At least a few sentences.

≡, Task Response Prompt

Response:

I initially had trouble with problem 3, regarding displaying the original data types for decimal and string data types. It

turns out, after tinkering around, that I can type cast and convert the value into a positive value all in one line i.e. ```php
$output[] = (string)abs($number). I also had forgotten that decimal values in PHP are called doubles and that resolved that issue. For example, I had been writing elseif (gettype($number)==='decimal') instead of (gettype($number)==='double').

End of Task 1

100%

Group: Reflection
Task #2: Include the pull request link for this branch
Weight: ~33%
Points: ~0.33

^ COLLAPSE ^

ⓘ Details:

The correct link will end with /pull/ and a number.

🔗Task URLs

URL #1

https://github.com/AlCSS442/asa89-IT202-007/pull/4

URL
https://github.com/AlCSS442/asa89-IT202-007/p

End of Task 2

Task

100%

Group: Reflection
Task #3: Add Screenshot of Wakatime
Weight: ~33%
Points: ~0.33

^ COLLAPSE ^

ⓘ Details:

Note: The duration of time isn't directly related to the grade, the goal is to just make sure time is being tracked
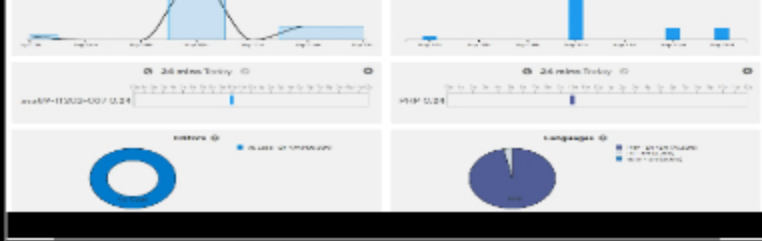
🖼 Task Screenshots

Gallery Style: 2 Columns

4   2   1

2 hrs 47 mins over the Last 7 Days.

screenshot of wakatime

End of Task 3

End of Group: Reflection
Task Status: 3/3

End of Assignment