

Submission Worksheet

CLICK TO GRADE

<https://learn.ethereallab.app/assignment/IT202-008-S2024/it202-m2-php-problems/grade/yc73>

IT202-008-S2024 - [IT202] M2 PHP Problems

Submissions:

Submission Selection

1 Submission [active] 2/4/2024 8:11:22 PM

Instructions

^ COLLAPSE ^

Guide:

1. Make sure you're in the main branch locally and ``git pull origin main`` 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 <homework branch>`` (see below and don't include the `< >`)
 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 main (i.e., `main <- "homework branch"`)
 5. Open and complete the merge of the pull request (it should turn purple)
 6. Locally checkout main and pull the latest changes (to prepare for future work)
12. Take the same output file and upload it to Canvas
 1. *This step is new since GitHub renders the PDF as an image the links aren't clickable so this method works better
 2. *Remember, the github process of these files are encouragement for your tracking of your progress

Branch name: M2-PHP-Problems

Tasks: 11 Points: 10.00

Problem 1 (3 pts.)

^ COLLAPSE ^

Task #1 - Points: 1

Text: Screenshot of the Problem 1 Solved Code and Output

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)

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input type="checkbox"/> #1	1	Edits were done only in the processArray() function and original template code/comments remain untouched
<input type="checkbox"/> #2	1	Only \$arr is used (no direct usage of \$a1, \$a2, \$a3, \$a4)
<input type="checkbox"/> #3	5	Only odd values output (not odd indexes/keys)
<input type="checkbox"/> #4	1	Includes code comments with student's ucid and date
<input type="checkbox"/> #5	1	Output of code includes heroku dev URL with student's ucid visible

Task Screenshots:

☐ Large Gallery



Checklist Items (1)

#4 Includes code comments with student's ucid and date



Checklist Items (1)

#5 Output of code includes heroku dev URL with student's ucid visible

Showing code for problem 1

Showing output from heroku dev (problem 1)

Task #2 - Points: 1

^ COLLAPSE ^

Task #2 - Points: 1

Text: Explain your solution

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	Clearly explains how the code/logic solves the problem (mentions how the odd values are determined)

Response:

I used a for each loop to go through each number in the array. For each iteration, the current number (\$num) is divided by 2 and the if-statement checks if the remainder is not equal to 0 (to verify if it's odd). This is because even numbers are exactly divisible by 2 which means their remainder is 0. While odd numbers are not evenly divided by 2, so their remainder cannot be 0. So, if it's not 0, then it's true the number is odd, and it executes the echo to output the (odd) value. If it's false, the number is even, and it executes 'continue' to skip the number and move to the next one.



^ COLLAPSE ^

Task #3 - Points: 1

Text: Link to Problem 1 file from Production

Details:

Recommended: Use the notes tool on the right to store the base URL for your heroku production URL

Paste the production URL below, append the path to the file from your dev URL.

Note: This link won't be active until you merge the pull request, so it's expected to not exist at the time of filling this task

URL #1

<https://yc73-it202-008-prod-35e9bd30f553.herokuapp.com/Module2/problem1.php>



Problem 2 (3 pts.)

^ COLLAPSE ^



^ COLLAPSE ^

Task #1 - Points: 1

Text: Screenshot of the Problem 2 Solved Code and Output

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)

Requires at least 2 screenshots (code + output from heroku dev)

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input type="checkbox"/> #1	1	Edits were done only in the getTotal() function and original template code/comments remain untouched (unless noted)
<input type="checkbox"/> #2	1	Only \$arr is used (no direct usage of \$a1, \$a2, \$a3, \$a4)
<input type="checkbox"/> #3	5	Passed in array's values get summed AND rounded to two decimal places like currency (i.e., 0.00, 0.10, 1.10)
<input type="checkbox"/> #4	1	Includes code comments with student's ucid and date
<input type="checkbox"/> #5	1	Output of code includes heroku dev URL with student's ucid visible

Task Screenshots:

☐ Large Gallery



Checklist Items (1)

#4 Includes code comments with student's ucid and date



Checklist Items (1)

#5 Output of code includes heroku dev URL with student's ucid visible

Showing code for problem 2

Showing output from heroku dev (problem 2)



^ COLLAPSE ^

Task #2 - Points: 1

Text: Explain your solution

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input type="checkbox"/> #1	1	Clearly explains how the code/logic solves the problem (mentions both how the values get summed and how the rounding is solved correctly)

Response:

To get the sum of the values, I used a for each loop that iterates through the array to grab each number and add it to the \$total. After it goes through each number, you would have the complete sum of the current array. Then, to round the total I used "number_format" which allows me to specify how many decimals to set my number with, which in this case is 2 (to round to 2 decimals).



^ COLLAPSE ^

Task #3 - Points: 1

Text: Link to Problem 2 file from Production

Details:

Recommended: Use the notes tool on the right to store the base URL for your heroku production URL

Paste the production URL below, append the path to the file from your dev URL.

Note: This link won't be active until you merge the pull request, so it's expected to not exist at the time of filling this task

URL #1

<https://yc73-it202-008-prod-35e9bd30f553.herokuapp.com/Module2/problem2.php>

Problem 3 (3 pts.)

^ COLLAPSE ^

Task #1 - Points: 1

Text: Screenshot of the Problem 2 Solved Code and Output

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.

Requires at least 2 screenshots (code + output from heroku dev)

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input type="checkbox"/> #1	1	Edits were done only in the bePositive() function and original template code/comments remain untouched
<input type="checkbox"/> #2	1	Only \$arr is used (no direct usage of \$a1, \$a2, \$a3, \$a4)
<input type="checkbox"/> #3	5	Passed in array's values will get converted to a positive version AND converted back to the original data type
<input type="checkbox"/> #4	1	Includes code comments with student's ucid and date
<input type="checkbox"/> #5	1	Output of code includes heroku dev URL with student's ucid visible

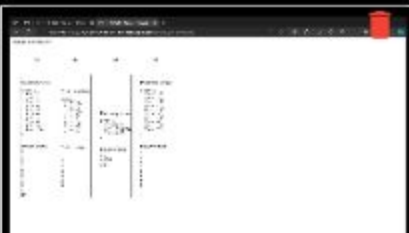
Task Screenshots:

☐ Large Gallery



Checklist Items (1)

#4 Includes code comments with student's ucid and date



Checklist Items (1)

#5 Output of code includes heroku dev URL with student's ucid visible

Showing code for problem 3

Showing output from heroku dev (problem 3)

Task #2 - Points: 1

Text: Explain your solution

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	Clearly explains how the code/logic solves the problem (mentions both the conversion to positive and conversion to original data type)

Response:

I used a foreach loop again to go through each number in the array. First, I handled the data types with if/elseif-statements that check if the current number is an int, float, or string. Next, (overall) for each data type, I did a positive conversion by taking the absolute value of the number with "abs()". I did this because absolute numbers are always positive, they can't be negative. For int and float I used it directly on \$num, but for the string data type, I first converted \$num to an integer by casting it with (int), then I took the absolute value. Then, to convert the numbers back to their original data type, I used casting - such as (int) to cast it back to an integer, (float) to cast it to a float, and (string) to cast it to a string.

Task #3 - Points: 1

Text: Link to Problem 3 file from Production

Details:

Recommended: Use the notes tool on the right to store the base URL for your heroku production URL

Paste the production URL below, append the path to the file from your dev URL.

Note: This link won't be active until you merge the pull request, so it's expected to note exist at the time of filling this task

URL #1

<https://yc73-it202-008-prod-35e9bd30f553.herokuapp.com/Module2/problem3.php>

Reflection (1 pt.)

Task #1 - Points: 1

Text: Reflect on your experience

i Details:

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

Provide concrete details/examples.

Response:

Although these are not major issues, I did require help trying to round/format numbers and convert data types. Even though I know how to do those things in Java, it's not the same in every language, so I had to use resources like W3schools to learn how those are written in PHP. When solving each problem, I learned how to rewrite functions and statements I've used before in other languages but in PHP - like checking data types "is_int", casting "(int)", and formatting "number_format."

Here's the pull request link in case the PDF cuts it out: <https://github.com/yaneliii/yc73-it202-008/pull/5>



^ COLLAPSE ^

Task #2 - Points: 1

Text: Include the pull request link for this branch

i Details:

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