

# Submission Worksheet

CLICK TO GRADE

<https://learn.ethereallab.app/assignment/IT202-008-S2024/it202-m2-php-problems/grade/lm457>

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

## Submissions:

Submission Selection

1 Submission [active] 2/5/2024 3:16:55 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

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

## Task Screenshots:

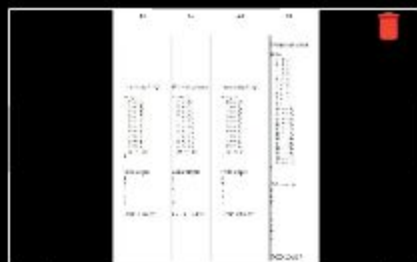
☐ Large Gallery


## Checklist Items (3)

#1 Edits were done only in the processArray() function and original template code/comments remain untouched

#2 Only \$arr is used (no direct usage of \$a1, \$a2, \$a3, \$a4)

#4 Includes code



## Checklist Items (2)

#3 Only odd values output (not odd indexes/keys)

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

Problem 1 code and heroku dev output.

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

Problem 1 code solved in IDE. Changes staged and  
committed.



^ 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:

In order to print out only the odd values in a given array, I first make use of a for loop to iterate through the length of the array using the count() function. Then I use an if statement to check the primary condition of whether or not each element in the array is divisible by 2, and output only those that aren't divisible by 2. This prints out all values in an array that are odd.



^ 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://it202-lm457-prod-cf97d0c34365.herokuapp.com/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
#1	1	Edits were done only in the getTotal() function and original template code/comments remain untouched (unless noted)
#2	1	Only \$arr is used (no direct usage of \$a1, \$a2, \$a3, \$a4)
#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)
#4	1	Includes code comments with student's ucid and date
#5	1	Output of code includes heroku dev URL with student's ucid visible

## Task Screenshots:

☐ Large Gallery



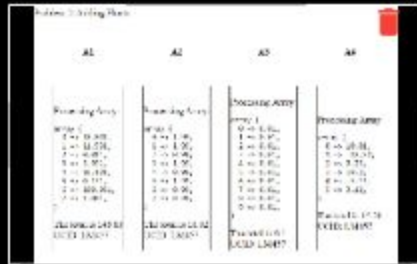
## Checklist Items (3)

#1 Edits were done only in the getTotal() function and original template code/comments remain untouched (unless noted)

#2 Only \$arr is used (no direct usage of \$a1, \$a2, \$a3, \$a4)

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

completed code of problem 2 in IDE.



## Checklist Items (2)

#3 Passed in array's values get summed AND rounded to two decimal places like currency (i.e., 0.00, 0.10, 1.10)

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

Heroku dev output of problem 2.

Task #2 - Points: 1

Text: Explain your solution

## Checklist

\*The checkboxes are for your own tracking

#	Points	Details
---	--------	---------

#	Points	Details
#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:

In order to sum the values in the array, I created a for-each loop that would iterate over every element/value in a given array. Within the loop, I added each value of the array to the counter "total" that would output the final summation value. Once I had added up all of the array's components, I used the round() function with "total" and "2" as parameters to get the program to round the total summation value to two decimal places. Lastly, using echo, I outputted the rounded total value of an array.

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://it202-lm457-prod-cf97d0c34365.herokuapp.com/problem2.php>

#### Problem 3 (3 pts.)

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
#1	1	Edits were done only in the bePositive() function and original template code/comments remain untouched

#2	1	Only \$arr is used (no direct usage of \$a1, \$a2, \$a3, \$a4)
#3	5	Passed in array's values will get converted to a positive version AND converted back to the original data type
#4	1	Includes code comments with student's ucid and date
#5	1	Output of code includes heroku dev URL with student's ucid visible

#### Task Screenshots:

☐ Large Gallery



#### Checklist Items (3)

#1 Edits were done only in the bePositive() function and original template code/comments remain untouched

#2 Only \$arr is used (no direct usage of \$a1, \$a2, \$a3, \$a4)

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

completed code of problem 3 in IDE



#### Checklist Items (2)

#3 Passed in array's values will get converted to a positive version AND converted back to the original data type

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

heroku dev output for problem 2 with positive values and data type.

☐ COLLAPSE ^

#### Task #2 - Points: 1

Text: Explain your solution

#### Checklist

\*The checkboxes are for your own tracking

#	Points	Details
#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 first create a for-each loop to iterate through every element/value in the array that is passed. Then, a new variable called "positiveValue" is assigned the absolute value of each individual element in the array. I then use var\_dump and pass "positiveValue" as an attribute to print out both the final data type as well as the absolute value of each element in the array on a new line each.

☐ COLLAPSE ^

#### Task #3 - Points: 1

Text: Link to Problem 3 file from Production



**i 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://it202-lm457-prod-cf97d0c34365.herokuapp.com/problem3.php>

Reflection (1 pt.)

^ COLLAPSE ^

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:

I had quite some difficulty with problem 2. Even though I was using the inbuilt round() function to round the total values to 2 decimal places, my final output didn't result in two decimal places for two of the 4 existing arrays. I tried several different ways of solving the problem until I found a mostly working approach. I realized that using a regular for loop wasn't giving the desired output for two of the 4 arrays in the program. One of them was rounded to 3 decimal places and the other one was like 0.09999999. After trial and error, eventually, I tried using a for-each loop instead, which gave me more desirable results, in that 3 of the four were properly formatted, and the 4th was semi-partially. The array that initially had a total value of 0.099999, now became 0.1, which while better, wasn't the 0.10 I was expecting. Either way, that was the best result I could obtain so I stuck with it. Upon fixing that issue, the rest of the process was relatively smooth and seamless. I had to remember how to stage and commit changes and push them to Git Hub after each edit. Also, deploying in dev and then moving it to prod in Heroku was also something I needed a reminder about given how confusing a lot of the commands are. But once I looked through past lessons, everything fell in line fairly easily.

Task #2 - Points: 1

Text: Include the pull request link for this branch

 Details:

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