

Submission Worksheet

Submission Data

Course: IT202-450-M2025

Assignment: IT202 PHP Problems

Student: Nathanael G. (ng569)

Status: Submitted | **Worksheet Progress:** 100+%

Potential Grade: 11.00/10.00 (110.00%)

Received Grade: 0.00/10.00 (0.00%)

Started: 6/10/2025 11:02:59 PM

Updated: 6/10/2025 11:49:49 PM

Grading Link: <https://learn.ethereallab.app/assignment/v3/IT202-450-M2025/it202-php-problems/grading/ng569>

View Link: <https://learn.ethereallab.app/assignment/v3/IT202-450-M2025/it202-php-problems/view/ng569>

Instructions

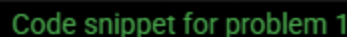
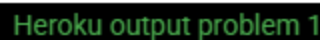
- Overview Link: <https://youtu.be/iwA6vyRPhoM>
- 1. Ensure you read all instructions and objectives before starting.
- 2. Create a new branch from dev called M2-Homework
 - 1. `git checkout dev` (ensure proper starting branch)
 - 2. `git pull origin dev` (ensure history is up to date)
 - 3. `git checkout -b M2-Homework` (create and switch to branch)
- 3. Copy the template code from here: [GitHub Repository - M2 Homework](#)
 - It includes Problems 1-4 and `base.php`. Put all into an M2 folder or similar inside your `public_html`
 - Immediately record to history
 - `git add public_html`
 - `git commit -m "adding M2 HW baseline files"`
 - `git push origin M2-Homework`
 - Create a Pull Request from M2-Homework to dev and keep it open
- 4. Fill out the below worksheet
 - Each Problem requires the following as you work
 - Ensure there's a comment with your UCID, date, and brief summary of how the problem was solved
 - Initial outline/plan of how you'll solve it via comments (add/commit after this stage)
 - Code solution (add/commit periodically as needed)
- 5. Once finished, click "Submit and Export"
- 6. Locally add the generated PDF to a folder of your choosing inside your repository folder and move it to Github
 - 1. `git add .`
 - 2. `git commit -m "adding PDF"`
 - 3. `git push origin M2-Homework`
 - 4. On Github merge the pull request from M2-Homework to dev
 - 5. On Github create a pull request from dev to prod and immediately merge. (This will trigger the prod deploy to make the heroku prod links work)
- 7. Upload the same PDF to Canvas
- 8. Sync Local

- Section #1: (2 pts.) Problem 1 - Odds

≡ Task #1 (2 pts.) - Edit the `printOdds` function to output odd values of the array

Part 1:

1. Snippet of relevant code showing solution (with ucid/date comment)
2. Full output of executing the program (visit the proper file on Heroku dev after a manual deploy)



🔗 Part 2:

- Direct link to the file in the homework related branch from Github (should end in `.php`)
- Direct link to the file on Heroku Prod (Just grab the base prod url and manually enter the path to the file)

URL #1

https://github.com/Nate-Gaw/ng569-IT202-450-M2-Homework/public_html/M2/problem1.php



URL

<https://github.com/Nate-Gaw/ng5>



URL #2

<https://ng569-it202-450-prod-3272507b1c51.herokuapp.com/M2/problem1.php>



URL

<https://ng569-it202-450-prod-3272507b1c51.herokuapp.com/M2/problem1.php>



Saved: 6/10/2025 11:13:02 PM

Part 3:

Progress: 100%

Details:

Briefly explain **how** the code solves the challenge (note: this isn't the same as **what** the code does)

Your Response:

I used a for loop that would iterate through each element in the array. Then I used the modulus symbol to return the remainder of the element modulus 2. This way if it was positive it would return 0 and if it was odd it would return 1. Then the odd numbers would be placed into another array full of the odd numbers in the original array, which would then be displayed by iterating over the entire array and outputting each element.



Saved: 6/10/2025 11:13:02 PM

Section #2: (2 pts.) Problem 2 - Sum

Progress: 100%

Task #1 (2 pts.) - Edit the `sumValues` function to sum the array values and present them in a format with exactly two decimal places

Progress: 100%

Part 1:

Progress: 100%

Details:

Two screenshots are expected

1. Snippet of relevant code showing solution (with ucid/date comment)
2. Full output of executing the program (visit the proper file on Heroku dev after a manual deploy)

```

// Challenge 1: Sum all the values in the array and assign to total
// Challenge 2: Sum all the values in the array and assign to total
// Challenge 3: Sum all the values in the array and assign to total
// Challenge 4: Sum all the values in the array and assign to total
// Challenge 5: Sum all the values in the array and assign to total
// Challenge 6: Sum all the values in the array and assign to total
// Challenge 7: Sum all the values in the array and assign to total
// Challenge 8: Sum all the values in the array and assign to total
// Challenge 9: Sum all the values in the array and assign to total
// Challenge 10: Sum all the values in the array and assign to total
// Challenge 11: Sum all the values in the array and assign to total
// Challenge 12: Sum all the values in the array and assign to total
// Challenge 13: Sum all the values in the array and assign to total
// Challenge 14: Sum all the values in the array and assign to total
// Challenge 15: Sum all the values in the array and assign to total
// Challenge 16: Sum all the values in the array and assign to total
// Challenge 17: Sum all the values in the array and assign to total
// Challenge 18: Sum all the values in the array and assign to total
// Challenge 19: Sum all the values in the array and assign to total
// Challenge 20: Sum all the values in the array and assign to total
// Challenge 21: Sum all the values in the array and assign to total
// Challenge 22: Sum all the values in the array and assign to total
// Challenge 23: Sum all the values in the array and assign to total
// Challenge 24: Sum all the values in the array and assign to total
// Challenge 25: Sum all the values in the array and assign to total
// Challenge 26: Sum all the values in the array and assign to total
// Challenge 27: Sum all the values in the array and assign to total
// Challenge 28: Sum all the values in the array and assign to total
// Challenge 29: Sum all the values in the array and assign to total
// Challenge 30: Sum all the values in the array and assign to total
// Challenge 31: Sum all the values in the array and assign to total
// Challenge 32: Sum all the values in the array and assign to total
// Challenge 33: Sum all the values in the array and assign to total
// Challenge 34: Sum all the values in the array and assign to total
// Challenge 35: Sum all the values in the array and assign to total
// Challenge 36: Sum all the values in the array and assign to total
// Challenge 37: Sum all the values in the array and assign to total
// Challenge 38: Sum all the values in the array and assign to total
// Challenge 39: Sum all the values in the array and assign to total
// Challenge 40: Sum all the values in the array and assign to total
// Challenge 41: Sum all the values in the array and assign to total
// Challenge 42: Sum all the values in the array and assign to total
// Challenge 43: Sum all the values in the array and assign to total
// Challenge 44: Sum all the values in the array and assign to total
// Challenge 45: Sum all the values in the array and assign to total
// Challenge 46: Sum all the values in the array and assign to total
// Challenge 47: Sum all the values in the array and assign to total
// Challenge 48: Sum all the values in the array and assign to total
// Challenge 49: Sum all the values in the array and assign to total
// Challenge 50: Sum all the values in the array and assign to total
// Challenge 51: Sum all the values in the array and assign to total
// Challenge 52: Sum all the values in the array and assign to total
// Challenge 53: Sum all the values in the array and assign to total
// Challenge 54: Sum all the values in the array and assign to total
// Challenge 55: Sum all the values in the array and assign to total
// Challenge 56: Sum all the values in the array and assign to total
// Challenge 57: Sum all the values in the array and assign to total
// Challenge 58: Sum all the values in the array and assign to total
// Challenge 59: Sum all the values in the array and assign to total
// Challenge 60: Sum all the values in the array and assign to total
// Challenge 61: Sum all the values in the array and assign to total
// Challenge 62: Sum all the values in the array and assign to total
// Challenge 63: Sum all the values in the array and assign to total
// Challenge 64: Sum all the values in the array and assign to total
// Challenge 65: Sum all the values in the array and assign to total
// Challenge 66: Sum all the values in the array and assign to total
// Challenge 67: Sum all the values in the array and assign to total
// Challenge 68: Sum all the values in the array and assign to total
// Challenge 69: Sum all the values in the array and assign to total
// Challenge 70: Sum all the values in the array and assign to total
// Challenge 71: Sum all the values in the array and assign to total
// Challenge 72: Sum all the values in the array and assign to total
// Challenge 73: Sum all the values in the array and assign to total
// Challenge 74: Sum all the values in the array and assign to total
// Challenge 75: Sum all the values in the array and assign to total
// Challenge 76: Sum all the values in the array and assign to total
// Challenge 77: Sum all the values in the array and assign to total
// Challenge 78: Sum all the values in the array and assign to total
// Challenge 79: Sum all the values in the array and assign to total
// Challenge 80: Sum all the values in the array and assign to total
// Challenge 81: Sum all the values in the array and assign to total
// Challenge 82: Sum all the values in the array and assign to total
// Challenge 83: Sum all the values in the array and assign to total
// Challenge 84: Sum all the values in the array and assign to total
// Challenge 85: Sum all the values in the array and assign to total
// Challenge 86: Sum all the values in the array and assign to total
// Challenge 87: Sum all the values in the array and assign to total
// Challenge 88: Sum all the values in the array and assign to total
// Challenge 89: Sum all the values in the array and assign to total
// Challenge 90: Sum all the values in the array and assign to total
// Challenge 91: Sum all the values in the array and assign to total
// Challenge 92: Sum all the values in the array and assign to total
// Challenge 93: Sum all the values in the array and assign to total
// Challenge 94: Sum all the values in the array and assign to total
// Challenge 95: Sum all the values in the array and assign to total
// Challenge 96: Sum all the values in the array and assign to total
// Challenge 97: Sum all the values in the array and assign to total
// Challenge 98: Sum all the values in the array and assign to total
// Challenge 99: Sum all the values in the array and assign to total
// Challenge 100: Sum all the values in the array and assign to total

```

Code snippet for problem 2



Heroku output problem 2

Saved: 6/10/2025 11:18:52 PM

Part 2:

Progress: 100%

Details:

- Direct link to the file in the homework related branch from Github (should end in `.php`)
- Direct link to the file on Heroku Prod (Just grab the base prod url and manually enter the path to the file)

URL #1

https://github.com/Nate-Gaw/ng569-IT202-450-M2-Homework/public_html/M2/problem2.php



<https://github.com/Nate-Gaw/ng5>



URL #2

<https://ng569-it202-450-prod-3272507b1c51.herokuapp.com/M2/problem2.php>



<https://ng569-it202-450-prod-3272507b1c51.herokuapp.com/M2/problem2.php>



Saved: 6/10/2025 11:18:52 PM

Part 3:


Progress: 100%

Details:

Briefly explain `how` the code solves the challenges (note: this isn't the same as `what` the code does)

Your Response:

I used a for loop to iterate over the array. Each element in the array would then be added to a variable called total. Then using number_format() function, which when given 2 arguments will

 Saved: 6/10/2025 11:18:52 PM

Progress: 100%

Progress: 100%

Progress: 100%

Two screenshots are expected

The screenshot shows a C++ program for calculating the area of a circle using Simpson's 1/3 rule. The program is as follows:

```

#include <iostream>
using namespace std;
const double pi = 3.14159265358979323846;

double area(double r) {
    return (pi/3) * (r*r * (1 + sin(pi/4) + sin(3*pi/4)));
}

int main() {
    double r = 5;
    double area = area(r);
    cout << "Area of the circle is: " << area << endl;
    return 0;
}


```

The output of the program is:

```

Area of the circle is: 78.5398

```

 Saved: 6/10/2025 11:25:48 PM

Progress: 100%

Details:

- Direct link to the file in the homework related branch from Github (should end in `.php`)
- Direct link to the file on Heroku Prod (Just grab the base prod url and manually enter the path to the file)

URL #1

https://github.com/Nate-Gaw/ng569-IT202-450-M2-Homework/public_html/M2/problem3.php



URL

<https://github.com/Nate-Gaw/ng5>**URL #2**

<https://ng569-it202-450-prod-3272507b1c51.herokuapp.com/M2/problem3.php>



URL

<https://ng569-it202-450-prod-3272507b1c51.herokuapp.com/M2/problem3.php>

Saved: 6/10/2025 11:25:48 PM

Part 3:

Progress: 100%

Details:

Briefly explain `how` the code solves the challenges (note: this isn't the same as `what` the code does)

Your Response:

I used a for loop to iterate over the array. Then I got the type and compared the first letter in the type in order to determine what type the element was. From there, I used the `abs()` function, which makes all negative values positive, to make all the elements positive after casting the elements as float. After that, depending on the original type, I recast the float into the appropriate type.



Saved: 6/10/2025 11:25:48 PM

Section #4: (2 pts.) Problem 4 - Strings

Progress: 100%

Task #1 (2 pts.) - Edit the `transformText` function to solve the challenges

Progress: 100%

Part 1:

Progress: 100%

Details:

Two screenshots are expected

1. Snippet of relevant code showing solution (with `ucid/date` comment)
2. Full output of executing the program (visit the proper file on Heroku dev after a manual `deploy`)

⇒ Part 3:

Progress: 100%

Details:

Briefly explain **how** the code solves the challenges (note: this isn't the same as **what** the code does)

Your Response:

Created a small and simple array of non-alphanumeric characters, so when I iterate over each char in the elements I was able to deduce the non-alphanumeric characters. Then I used `str_replace()` function, which replaces a chunk of a string based on the arguments, to remove the non-alphanumeric characters. Then I used `ucwords()` function in order to uppercase the first char in every word. Then I used `trim()` function, in order to remove any leading or preceeding whitespaces. Then I used a while loop to keep checking for 2 whitespaces within the element and keep replacing the 2 whitespace for 1, as to keep the separation between words but remove the excess.

 Saved: 6/10/2025 11:36:02 PM

≡ Task #2 (+ 1 pt.) - Edit the `transformText` function to solve the extra credit challenge (challenge 4)

Progress: 100%

📄 Part 1:

Progress: 100%

Details:

Two screenshots are expected

1. Snippet of relevant code showing solution (with uid/date comment)
2. Full output of executing the program (visit the proper file on Heroku dev after a manual deploy)

code snippet for ec



heroku output problem 4 part 1 ec

heroku output problem 4 part 2 ec



Saved: 6/10/2025 11:40:34 PM

⇒ Part 2:

Progress: 100%

Details:

Briefly explain **how** the code solves the extra credit challenge (note: this isn't the same as **what** the code does)

Your Response:

I first found the string length to determine the length. If the length was 3 then its already the middle 3 characters so nothing has to be done. If the length was less than 3 then there were not enough characters. If the length was 4, then we took the middle 2 characters. Then if the length was more than 4, then determine the middle of the string length and get the 3 characters in the middle.



Saved: 6/10/2025 11:40:34 PM

Section #5: (2 pts.) Misc

Progress: 100%

≡ Task #1 (0.67 pts.) - Github Details

Progress: 100%

Part 1:

Progress: 100%

Details:

From the Commits tab of the Pull Request screenshot the commit history Following minimum should be present


```
31 secs composer.lock
14 secs public_html/index.php
9 secs public_html/M2/Problem2
7 secs public_html/proposal.md
7 secs lib/config.php
5 secs Profile
3 secs README.md
2 secs public_html/test_db.php
1 sec public_html/README.md
0 secs myFile.txt
```

individual



Saved: 6/10/2025 11:44:31 PM

≡ Task #3 (0.67 pts.) - Reflection

Progress: 100%

⇒ Task #1 (0.33 pts.) - What did you learn?

Progress: 100%

Details:

Briefly answer the question (at least a few decent sentences)

Your Response:

I learned about php. I have never used php so it definitely was a struggle to understand sometimes. But it was very similar to pearl, which I've seen before. I also learned a lot about the different integrated functions that php has and how different variable manipulation is



Saved: 6/10/2025 11:47:20 PM

⇒ Task #2 (0.33 pts.) - What was the easiest part of the assignment?

Progress: 100%

Details:

Briefly answer the question (at least a few decent sentences)

Your Response:

The easiest part of the assignment has to be the odd number part. I've done this so many times in many other languages so it came very naturally. The only struggle was of course learning the language, but the math never changes.



Saved: 6/10/2025 11:48:20 PM

⇒ Task #3 (0.33 pts.) - What was the hardest part of the

assignment?

Progress: 100%

Details:

Briefly answer the question (at least a few decent sentences)

Your Response:

The hardest part of the assignment was probably removing the whitespace. I was trying not to use any loops because from what I saw in the video, loops were not needed. But after looking at the php index in w3schools, I still could not figure out how to do that without loops, so I had to resort to using one.



Saved: 6/10/2025 11:49:49 PM