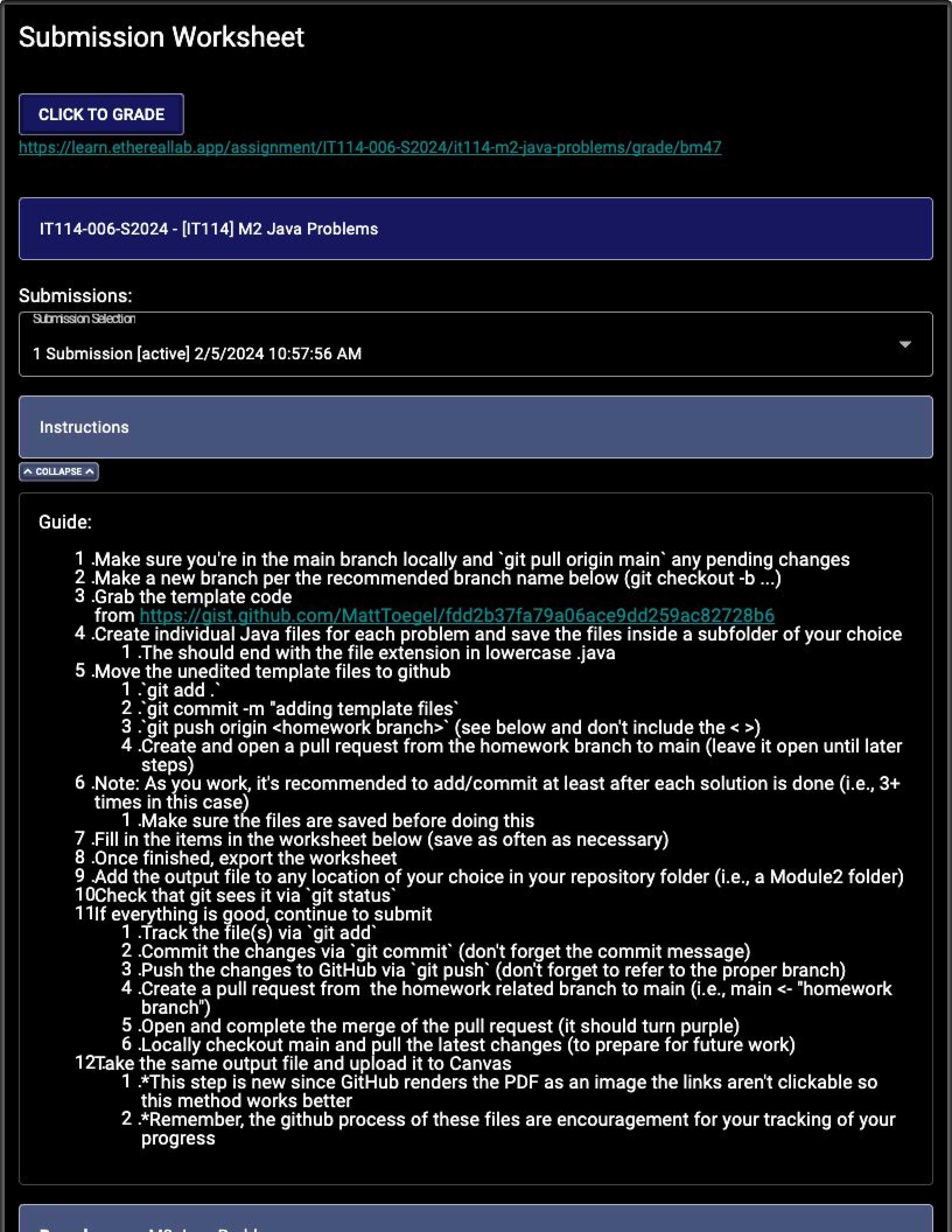


Submission Worksheet

https://learn.ethereallab.app/assignment/IT114-006-S2024/it114-m2-java-problems/grade/bm47

IT114-006-S2024 - [IT114] M2 Java Problems

Submissions:

Submission Selection

1 Submission [active] 2/5/2024 10:57:56 AM

Instructions

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/fdd2b37fa79a06ace9dd259ac82728b6
4 .Create individual Java files for each problem and save the files inside a subfolder of your choice
  1 .The should end with the file extension in lowercase .java
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)
10Check that git sees it via `git status`
11If 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)
12Take 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-Java-Problems

Tasks: 8 Points: 10.00

●

^ COLLAPSE ^

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 terminal)

Checklist

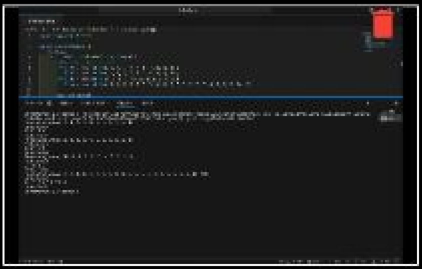
*The checkboxes are for your own tracking

#	Points	Details
□ #1	1	Edits were done only in the processArray() method 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	Terminal output is fully visible

Task Screenshots:

Large Gallery

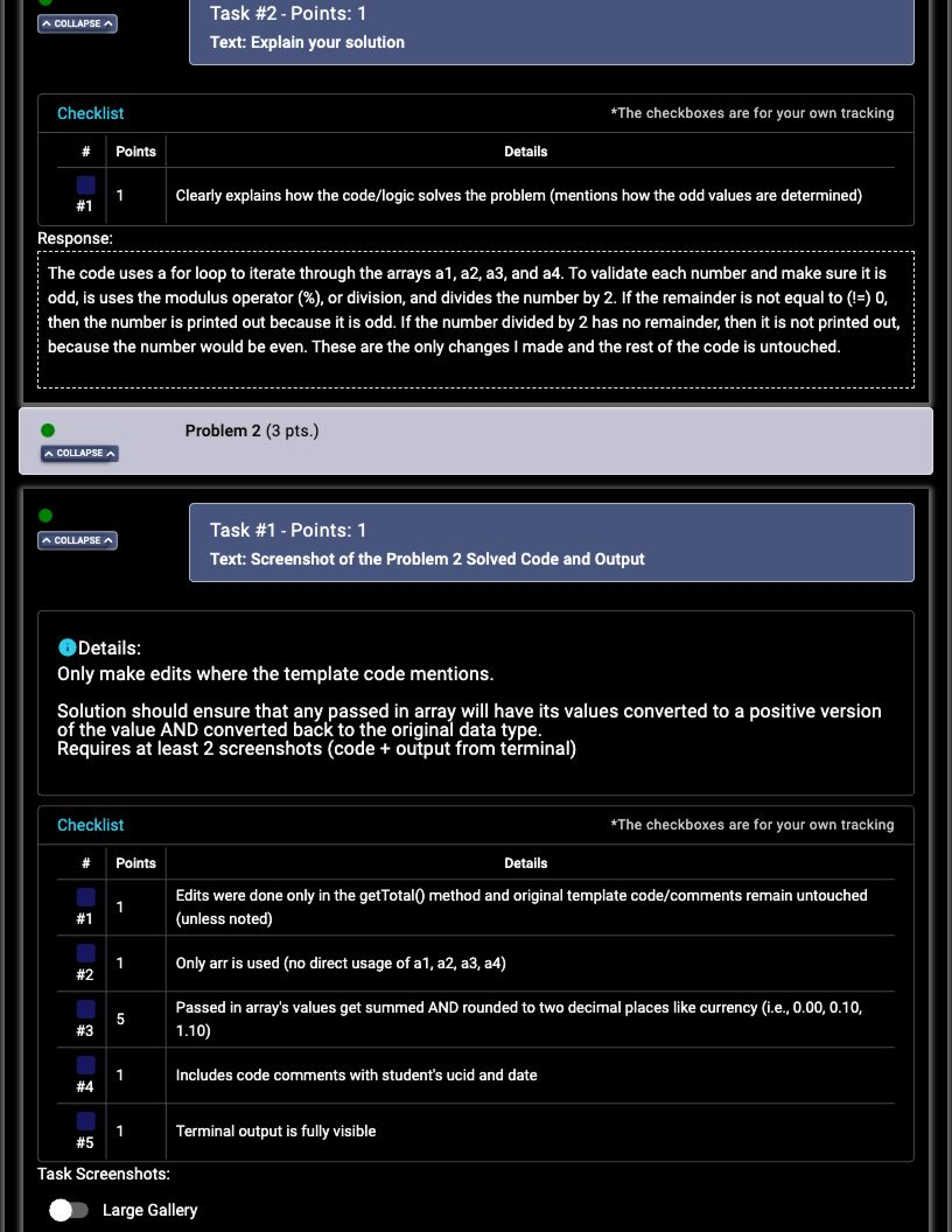
Checklist Items (0)



Checklist Items (0)

the code used for problem 1. arr is used, and there is no direct usage of the arrays by name. also, there are only odd values outputted in the terminal.

the terminal output is here for the code. only odd values are outputted.



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 how the odd values are determined)

Response:

The code uses a for loop to iterate through the arrays a1, a2, a3, and a4. To validate each number and make sure it is odd, is uses the modulus operator (%), or division, and divides the number by 2. If the remainder is not equal to (!=) 0, then the number is printed out because it is odd. If the number divided by 2 has no remainder, then it is not printed out, because the number would be even. These are the only changes I made and the rest of the code is untouched.

● 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 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 terminal)

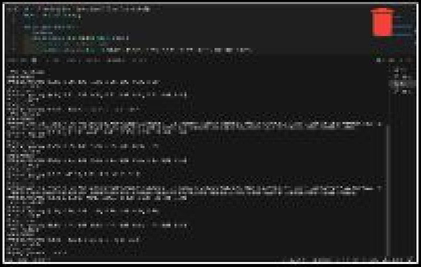
Checklist *The checkboxes are for your own tracking

#	Points	Details
□ #1	1	Edits were done only in the getTotal() method 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	Terminal output is fully visible

Task Screenshots:

Large Gallery

Checklist Items (0)



Checklist Items (0)

shows the code and edited parts, along with my UCID	terminal output (ran a few times)

●

^ 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 how the values get summed and how the rounding is solved correctly)

Response:

Line 29 rounds the integer to the nearest decimal place. I found that the way that made the most sense to me was to multiply the number by 100, getting rid of the decimal places, and this would round the number to the nearest whole number. Once I divide this number by 100, it will be rounded to two decimal places like intended. Lines 17-26 contain the method used to add all of the numbers in the array. It uses the for loop to loop through each array and adds each number to the current sum, and then when it reaches the last number and adds it, it goes to the next array.

●

^ COLLAPSE ^

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 terminal)

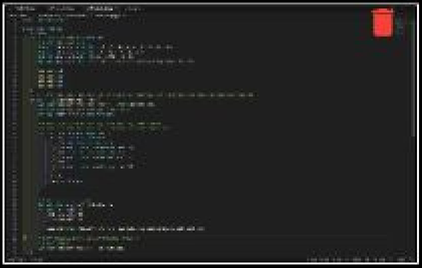
Checklist *The checkboxes are for your own tracking

#	Points	Details
□ #1	1	Edits were done only in the bePositive() method 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	Terminal output is fully visible

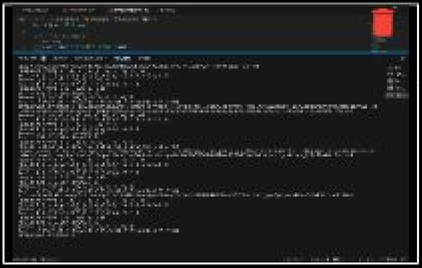
Task Screenshots:

Large Gallery



Checklist Items (0)

Code that does what the instructions asked.



Checklist Items (0)

Terminal output, ran code a few times in a row.

● 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:

Lines 17-44 both order the numbers correctly, ensure that they are positive numbers when outputted, and make sure that they are the correct specified datatype.

● Reflection (1 pt.)

COLLAPSE

● COLLAPSE

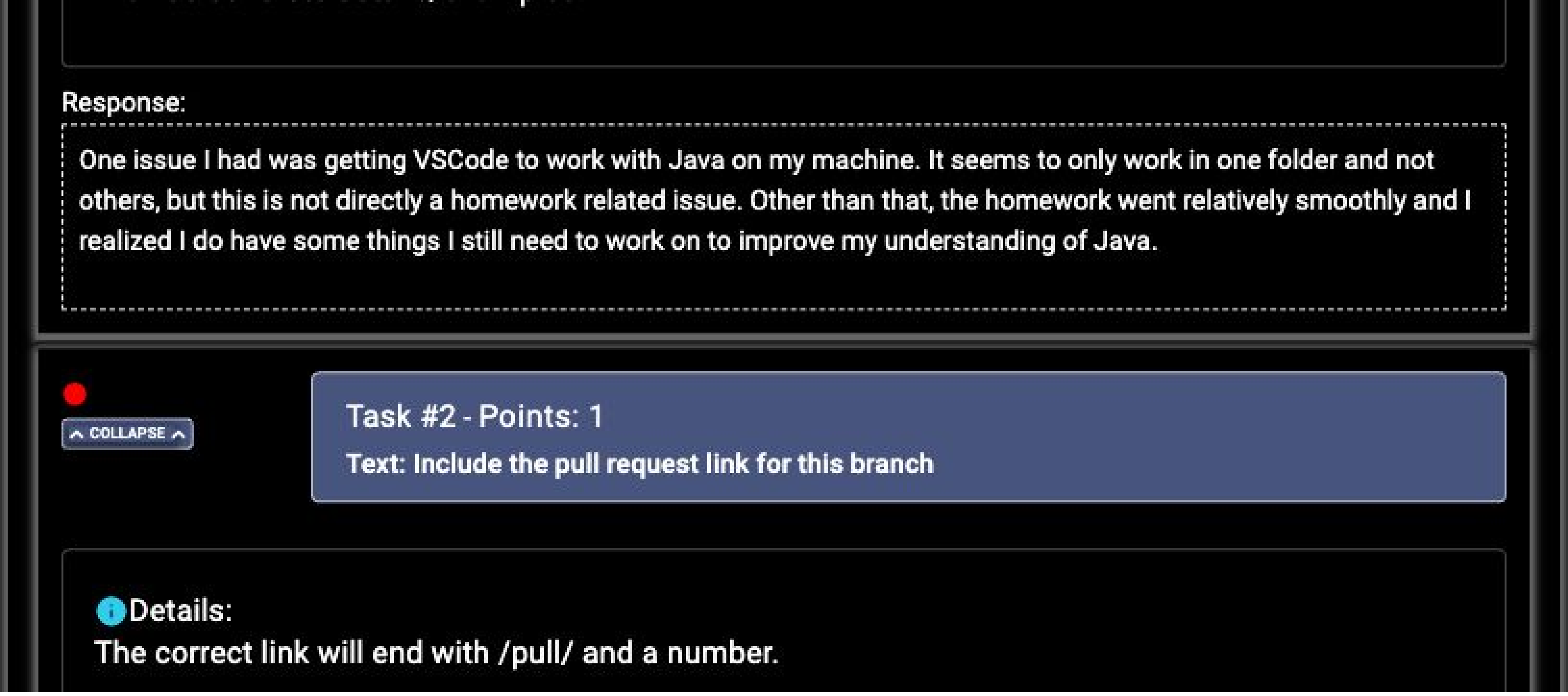
Task #1 - Points: 1

Text: Reflect on your experience

Details:

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

Provide concrete details/examples.



Response:

One issue I had was getting VSCode to work with Java on my machine. It seems to only work in one folder and not others, but this is not directly a homework related issue. Other than that, the homework went relatively smoothly and I realized I do have some things I still need to work on to improve my understanding of Java.

●

Task #2 - Points: 1

Text: Include the pull request link for this branch

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