Submission Worksheet

CLICK TO GRADE

https://learn.ethereallab.app/assignment/IT114-451-M2024/it114-module-2-java-problems/grade/sa2796

IT114-451-M2024 - [IT114] Module 2 Java Problems

Submissions:

Submission Selection

1 Submission [active] 5/27/2024 5:38:32 PM

Instructions

^ COLLAPSE ^

Overview Video: https://youtu.be/4M8Di5jrcZQ

Guide:

- Make sure you're in the main branch locally and git pull origin main any pending changes.
- 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.
 - They should end with the file extension in lowercase .java.
- 5. Move the unedited template files to GitHub.
 - 1. git add .
 - git commit -m "adding template files"
 - 3. git push origin branch name (see below).
 - Create and open a pull request from the homework branch to main (leave it open until later steps).
- Note: As you work, it's recommended to add/commit at least after each solution is done (i.e., 3+ times in this case).
 - Make sure the files are saved before doing this.
- Fill in the items in the worksheet below (save as often as necessary).
- 8. Once finished, export the worksheet.
- Add the output file to any location of your choice in your repository folder (i.e., a Module2 folder).
- Check that git sees it via git status.
- 11. If everything is good, continue to submit.

- 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).
- 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.

Branch name: M2-Java-Problems

Tasks: 6 Points: 10.00

Problem 1 (3 pts.)



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)

#1) Screenshot the output of the solved problem



Caption (required) ~

Describe/highlight what's being shown

I'm showing the output of the code of problem 1.

#2) Screenshot the code solution (ucid/date must be included as a comment)



Caption (required) <

Describe/highlight what's being shown

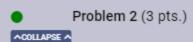
I am displaying the code of Problem 1 which includes UCID, Date, and Explanation.

Explanation (required) ~

Explain in concise steps how this logically works

PREVIEW RESPONSE

A for loop is used to iterate each element by using "num" in arrays and checks to determine if a number is odd by using the if statement which in this case was if(num%2 !0). What this if statement does is sees if the number isn't divisible by 2 and will print out a odd number.





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 summed AND the final result converted to two decimal places (i.e., 0.10, 1.00, 1.01).

Requires at least 2 screenshots (code + output from terminal)

#1) Screenshot the output of the solved problem



```
State Problem

State
```

Caption (required) <

Describe/highlight what's being shown

I'm showing the output of the code of problem 2.

#2) Screenshot the code solution (ucid/date must be included as a comment)



```
| Production to "A Production to "A production to the state of the sta
```

Caption (required) 🗸

Describe/highlight what's being shown

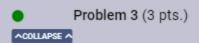
I am displaying the code of Problem 2 which includes UCID, Date, and Explanation.

Explanation (required) 🗸

Explain in concise steps how this logically works



Firstly, I used a for loop that adds up all the elements in array "arr" and the result is stored in the total variable. I then used "String.format" which was the method used to translate the total variable to 2 decimal places.





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)

#1) Screenshot the output of the solved problem



Caption (required) 🗸

Describe/highlight what's being shown

I'm showing the output of the code of problem 3.

#2) Screenshot the code solution (ucid/date must be included as a comment)



```
### Comment of the Co
```



Caption (required) 🗸

Describe/highlight what's being shown

I am displaying the code of Problem 3 which includes UCID, Date, and Explanation.

Explanation (required) <

Explain in concise steps how this logically works



The conversion to positive run by checking to see the type of data of each element in the array. The utilization of Math.abs() was used for Double and Integer values to acquire the positive values. The String values had the code parse as a Double to then convert it to a positive and convert it back to a String while still utilizing Math.abs(). Casting was used for Integer and Double alongside conditional checks to make sure data types were bringing values that are uniform with the original types of data





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. At least a few sentences.

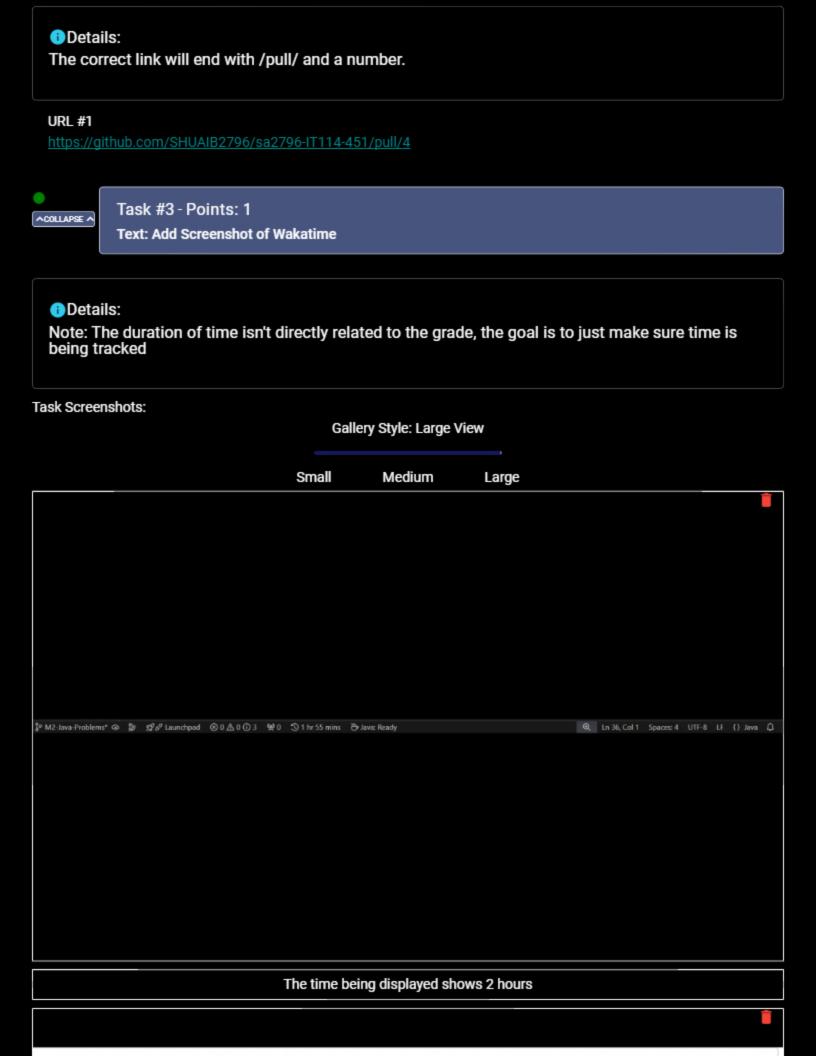
Response:

I learned how to calculate sums of elements in an array and many other Java concepts that I hadn't explored yet which I found intriguing. I learned how to properly use String.format and fixed some parts of Math.abs{} that was giving me some errors. Professor Toegel had helped me in the introductory parts in making sure my Git Bash was running well and that everything was synced to ensure no data was lost.



Task #2 - Points: 1

Text: Include the pull request link for this branch

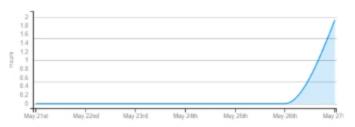


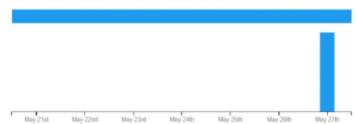
Projects • sa2796-IT114-451

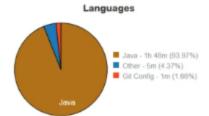
total 0 secs

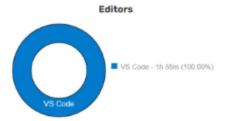
0 -

1 hr 55 mins over the Last 7 Days in sa2796-IT114-451 under all branches. 🖎

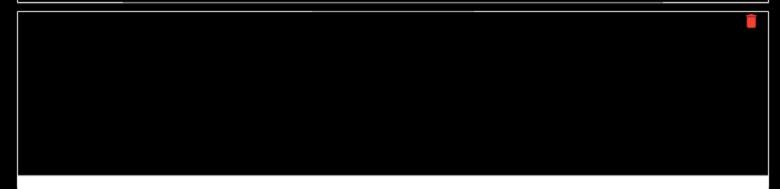








Showing total repo time



Files

45 mins Problem1.java 45 mins Problem2.java

18 mins Problem3.java

5 mins __eadings_IT114-451-M2024.pdf

1 min .gitignore

Branches

1 hr 55 mins M2-Java-Problems

Showing specific times for each problem

End of Assignment