

1. The GitHub URL to the repository containing your code:

<https://github.com/ArunRao1997/Homework-04a---Develop-with-the-Perspective-of-the-Tester-in-mind>

2. Thoughts I had when designing the code and the approach used:

When designing the code for this assignment, I first thought about how to break down the problem into smaller, manageable parts. I knew that I needed to first retrieve the list of repositories for a given user, and then for each repository, retrieve the number of commits. So, I started by writing functions to handle each of these tasks.

To make testing the program easier, I made sure to separate the functions into smaller, modular pieces. This way, I could test each function individually and ensure that it was working correctly before moving on to the next one. Additionally, I made sure to handle any errors that might arise, such as invalid user IDs or repositories with no commits. I have addressed these challenges by writing unit tests for different scenarios to ensure that the program works as expected.

Furthermore, I made sure to use input parameters that are relevant to the function's task and return values that indicate the status of the operation. This made it easier to write unit tests for each function. I also made sure to use clear and descriptive function names to make it easy to understand what each function does.

One challenge that I faced when testing this assignment was that GitHub has API rate limits, which can cause errors if too many requests are made within a certain period of time. To overcome this, I made sure to include a delay between each API request to ensure that I did not exceed the rate limit. Initially, I used **Travis-ci** but I had to switch back to **circleci** due to subscription issues.

I faced the below error while setting up Travis-ci:

(Error Banner on Travis-ci: We are unable to start your build at this time. You exceeded the number of users allowed for your plan. Please review your plan details and follow the steps to resolution.)

Overall, I think that the key to making this program easy to test was to break it down into smaller, modular functions and to use descriptive function names. Additionally, I made sure to be mindful of GitHub's API rate limits when testing the program.