Question 3 version 1 documentation

Aim: The aim of this code is to calculate and display the overview of miles run by a group of runners for seven days. It allows the user to input the number of runners, their names, and the miles run by each runner on different days of the week. The code then calculates the total miles and average miles for each runner and displays the data in a tabular format.

Process:

1. The code prompts the user to enter the number of runners and their names.
2. It accepts the input for the miles run by each runner on each day of the week.
3. The code calculates the total miles and average miles for each runner and stores the values.
4. It determines the maximum length of names and values for proper tabular formatting.
5. Based on the length of values, the code prints the tabular format overview of the miles run by the runners.
6. The code provides an option to search for a specific runner and displays their data if found.
7. Finally, the code displays a thank you message and terminates.

The refactored code introduces the following new features:

Modularized Functions: The code is divided into separate functions to handle different tasks such as accepting input, calculating total and average miles, printing data, and handling user choice. This modular approach improves code readability, maintainability, and reusability.

Parameterized Functions: The functions accept parameters to receive and process data specific to each task. For example, the accept function takes an array of names, a 2D array of miles, and an array of predefined names as parameters to store the runner's name and miles run.

Looping Functionality: The choice function allows the user to repeatedly choose between displaying data for all runners or a specific runner. The loop continues until the user chooses to exit by entering '0'.

Advantage

- ✓ Increase code readability
- ✓ Can be debugged easily
- ✓ Efficient management of spacing between characters.