**Name: Usman Ul Haq**

**Roll# BSAIM-035**

**AI LAB TASKs**

**Documentation**

**Fizz Buzz Game Documentation**

This program is a multiplayer Fizz Buzz game that takes three inputs from the user: a starting number, an ending number, and the number of players. The game follows the classic Fizz Buzz rules, where numbers divisible by 3 are replaced with "Fizz", those divisible by 5 with "Buzz", and numbers divisible by both 3 and 5 with "Fizz Buzz". Any other number is printed as is.

The game begins with the first player and cycles through the players in order. The program calculates whose turn it is by using the sequence of numbers, ensuring that turns rotate among all players. It then checks each number against the Fizz Buzz conditions and prints the appropriate output.

Once the program reaches the ending number, it displays a "Game Over!" message, signaling the end of the game. This implementation effectively divides turns among players while maintaining the traditional Fizz Buzz logic, making it an interactive and engaging counting game.

**MiniProject 2** In particular we're going to finding the average budget of the films in our data set, and we're going to identify high budget films that exceed the average budget we calculate.

This program manages a list of movies along with their budgets and allows the user to add additional movies. It then calculates the **average movie budget** and identifies films that exceed this average.

**Functioning of the Program**

1. **Initial Movie List:**
   * The program starts with a predefined list of four movies, each paired with its production budget.
2. **User Input for Additional Movies:**
   * The user is prompted to enter how many new movies they would like to add.
   * For each movie, the user provides a name and a budget, which are then added to the existing list.
3. **Calculating the Average Budget:**
   * Once all movies are recorded, the program calculates the **total budget** by summing up the budgets of all movies.
   * The **average budget** is then determined by dividing the total budget by the number of movies.
4. **Identifying High-Budget Movies:**
   * The program compares each movie’s budget with the average budget.
   * Movies that exceed the average budget are identified and stored separately.
5. **Displaying Results:**
   * The program prints the **average movie budget** in a properly formatted currency style.
   * For each movie that exceeds the average budget, it displays the movie name and how much **higher** its budget is compared to the average.
   * Finally, it shows the **total number of movies** with above-average budgets.

This program provides an insightful analysis of movie budgets by allowing user interaction, performing calculations, and presenting a clear comparison between different films.