## Code 1: Fizz Buzz Game

This Python program is a simple Fizz Buzz Game that allows a user to play multiple rounds against the computer. At the beginning, the program greets the player and asks for their name. Then it asks how many rounds the player wants to play. If the player enters an invalid input, the program automatically sets the number of rounds to 5.

In each round, both the player and the computer take turns. The player plays on odd-numbered turns and guesses the correct word based on the Fizz Buzz rules:  
- If the total number is divisible by both 3 and 5, the correct answer is 'Fizz Buzz'.  
- If divisible by only 3, it’s 'Fizz'.  
- If divisible by only 5, it’s 'Buzz'.  
- Otherwise, the answer is simply the number itself.

If the player guesses correctly, their score increases, and after every three correct answers, a message of appreciation is shown. If the guess is wrong, the game ends immediately. On even-numbered turns, the computer plays automatically and follows the same Fizz Buzz logic.

Sometimes, the computer makes a mistake when a specific condition is met — when (total + computer\_turns) % 4 == 1 and total > 10. This adds randomness and fun to the game. If this mistake happens, the game ends instantly. At the end of all rounds, the program congratulates the player and gives feedback based on their performance.

## Code 2: Movie Budget Analysis

This program manages and analyzes a list of movies with their budgets. It starts with a predefined list of movies, each represented as a tuple containing the movie name and its budget amount.

The program asks the user how many new movies they want to add, and for each movie, it collects the name and budget. All new movies are added to the existing list.

After adding movies, the program calculates the total and average movie budget. Then it identifies which movies have budgets higher than the average and prints how much more they spent compared to the average.

Finally, the program finds and displays the movie with the highest budget and the one with the lowest budget. This code demonstrates how to use loops, lists, tuples, and conditional statements in Python to handle and analyze data efficiently.

Overall, both codes are great examples of using loops, conditions, and user interaction in Python programs.