****

**Name : Talha Usman**

**Class: SU92-BSAIM-F24-064**

**Subject: AI-Lab**

Fizz-Buzz Game

# Introduction:

This project is a Python implementation of a modified version of the classic FizzBuzz game. Instead of counting sequentially, it generates random numbers each round and uses the sum of the current and previous number to determine what to display. This makes the game more challenging and unpredictable, requiring players to think quickly and remember the previous number.

## ****Rules of the Game:****

1. The game will run for a fixed number of rounds (default 30).
2. In the first round, the number shown is used as it is.
3. From the second round onwards, the current number is added to the previous number to form a total.
4. If the total is divisible by 3 → write **“Fizz”**.
5. If the total is divisible by 5 → write **“Buzz”**.
6. If the total is divisible by both 3 and 5 → write **“Fizz Buzz”**.
7. If the total is not divisible by either, write the total number itself.
8. If you give a wrong answer, the game ends and your final score is displayed.

# Function and Initialization:

The program defines a function named fizzbuzz\_twist which accepts a parameter 'rounds' that controls how many times the game will run. A variable 'score' is initialized to keep track of the number of correct answers. A variable 'prev' is initialized as None to store the previous number.

# Game Logic:

For each round, a random number between 1 and 30 is generated. If it's the first round, that number is taken as the total, otherwise it is added to the previous number to form the new total. The program then checks if this total is divisible by 3, 5, or 15 and assigns 'Fizz', 'Buzz', or 'Fizz Buzz' as the correct answer respectively. If not divisible by any, the correct answer is the number itself.

# User Interaction:

The program asks the user to input their answer. If it matches the correct answer (case-insensitive), the score is increased, and the game moves to the next round. If the answer is wrong, the correct answer is displayed, and the game ends showing the final score.

# Output of the code :



# Flow of code:

