

**Name:**  Muhammad Awais Asghar

**Roll Number:** SU92-BSAIM-F24-065

**SUBJECT:** Artifical Intelligence (LAB)

**Task No -2**

**Question 1:**

**Why this code was made:**

The FizzBuzz game is a well-known coding that can be played with this program.  
Print "Fizz" if a number is divisible by three.  
Print "Buzz" if a number is divisible by 5.  
Print "Fizz Buzz" if a number is divisible by both 3 and 5.  
The purpose of this program is to create a small interactive **FizzBuzz game**.  
It challenges the user to guess whether a hidden number is fizz, buzz, fizzbuzz, or just a plain number.  
The program also keeps track of the score, allows the user to restart after making a mistake, and ends the game when the player chooses to quit.

**How this code works:**

**FizzBuzz function:**

* A number is checked:
  + If divisible by both 3 and 5 → return "fizzbuzz".
  + If divisible only by 3 → return "fizz".
  + If divisible only by 5 → return "buzz".
  + Otherwise → return "number".

**Game initialization:**

* score is set to 0 at the start.
* previous number is set to 0 (this is used to calculate the hidden number).

**Gameplay loop (while True):**

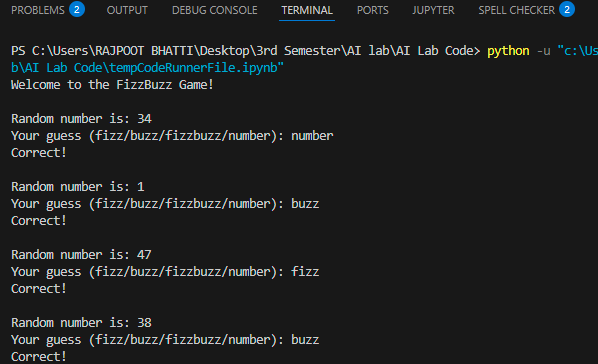
* A random number between 1 and 50 is generated.
* A **hidden number** is calculated by adding the random number to the previous number.
* The random number is shown to the user, and they must guess if the hidden number is fizz, buzz, fizzbuzz, or number.

**Answer checking:**

* The hidden number is passed into the fizzbuzz() function.
* If the user’s guess matches the correct answer → print "Correct!", increase score by 1, and update previous with the current number.
* If the guess is wrong → the correct answer and hidden number are displayed, along with the current score.

**Continue or exit:**

* After a wrong guess, the user is asked if they want to continue.
* If "yes" → the game restarts with score = 0 and previous = 0.
* If "no" → the game prints the final score and ends.



**Question 2:**

**Why this code was made:**

The purpose of this program is to analyze film budgets.

It determines a collection of films' average budget.

It indicates which films have more money than the norm.

Before the computations are performed, users can dynamically add more movies.

**How this code works:**

Movie names and budgets are first stored in a list of tuples. Tuples are used to give each film a structured name and budget. The user is repeatedly prompted by the while True loop to add more movies. If so, the user's name and budget are taken.  
The application attempts to translate the budget into a numerical value. The budget is reset to 0 if the user enters an invalid value. The movie list is then updated with the new film. To determine the average, the program adds up all of the film budgets and divides that total by the total number of films. The average movie budget in the dataset. Every movie is repeated by the program. If a movie’s budget is higher than the average, it prints: The movie’s name, Its budget, How much it exceeds the average. The quantity of above-average films is also counted. Lastly, the number of films that spent more than average is printed.

