NAME – Yogesh Kumar Verma

ROLL NO - 202401100300289

BRANCH - CSE(AI)

SECTION - D

COLLEGE - KIET GROUP OF INSTITUTION

FACULTY - ABHISHEK SUKLA

PROBLEM STATEMENT -Traffic Light Control System

INTRODUCTION

A Traffic Light Control System is a crucial component in managing road traffic efficiently. It operates by cycling through a sequence of lights—Red, Yellow, and Green—to regulate the movement of vehicles and pedestrians at intersections. This system helps prevent congestion, reduce accidents, and ensure smooth traffic flow.

In this project, we implement a Traffic Light Control System using Python. The system simulates real-world traffic signals by randomly determining the number of cycles it will run before stopping. Each cycle consists of a Red light (stop), a Yellow light (get ready), and a Green light (go), with predefined durations for each phase. At the end of the simulation, the program displays the final light that remains glowing.

The implementation makes use of Python's random module to generate a random number of cycles and the time module to introduce realistic delays between light transitions. This simulation provides a foundational understanding of traffic control mechanisms, which can be further expanded to incorporate real-time traffic data and adaptive control strategies.

METHODOLOGY

The methodology for implementing the Traffic Light Control System involves a structured approach, ensuring both functionality and realism. The process can be broken down into the following steps:

1. **Defining the Traffic Light States:** The system follows a sequential transition of lights:
   - Red Light: Indicates that vehicles must stop.
   - Yellow Light: Serves as a warning that the signal is about to change.
   - Green Light: Allows vehicles to move forward.
2. **Generating Random Traffic Cycles:**
   - To introduce variability, a random number is generated using Python's random.randint() function. This number determines how many times the traffic light sequence will run before stopping.
3. **Implementing Time Delays:**
   - The time.sleep() function is used to create realistic time delays for each light phase:

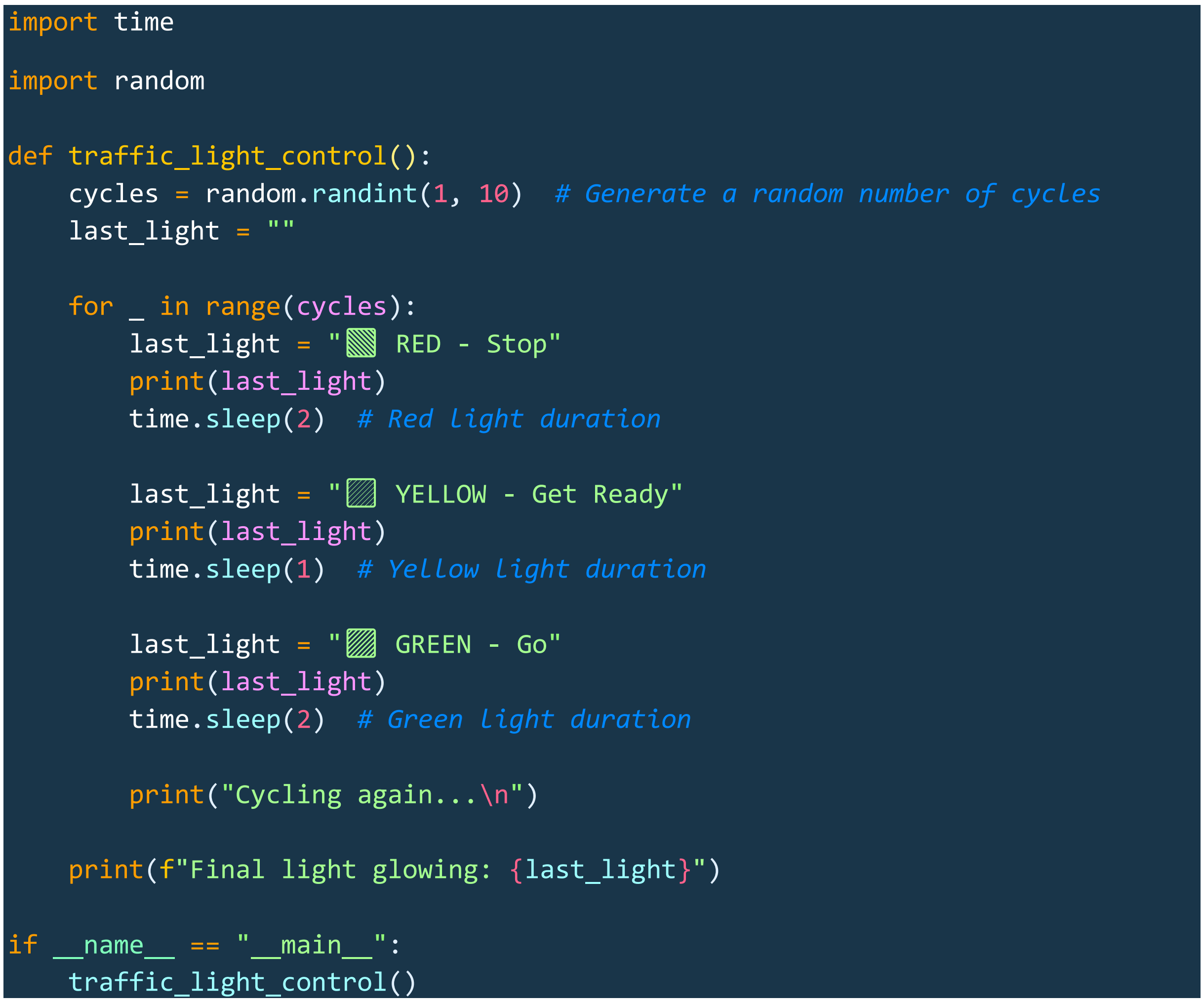
        - Red Light: 2 seconds
        - Yellow Light: 1 second
        - Green Light: 2 seconds

4. **Looping Through Traffic Cycles:**
    - A loop iterates through the predefined number of cycles, changing the lights in the specified order.
    - Each light change is printed to the console for visualization.
5. **Final Light Status Display:**
    - After completing all cycles, the last glowing light is displayed, indicating the final state of the traffic system.
6. **Scalability and Future Enhancements:**
    - The current implementation serves as a basic model that can be enhanced further by incorporating:
        - Traffic density-based adaptive controls.
        - Integration with sensor inputs for real-world traffic management.

- Graphical User Interface (GUI) for improved visualization.

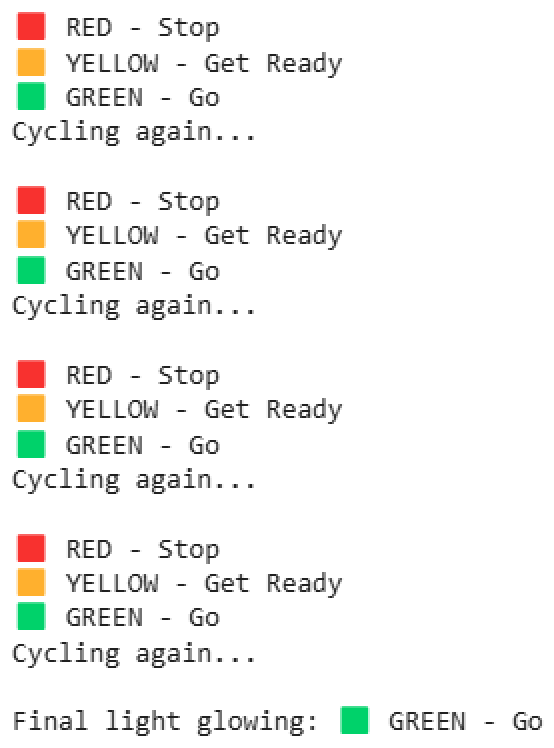
By following this methodology, the Traffic Light Control System effectively simulates a real-world traffic management process while remaining flexible for future enhancements.

CODE



```python
import time

import random

def traffic_light_control():
    cycles = random.randint(1, 10)  # Generate a random number of cycles
    last_light = ""

    for _ in range(cycles):
        last_light = "▨ RED - Stop"
        print(last_light)
        time.sleep(2)  # Red light duration

        last_light = "▨ YELLOW - Get Ready"
        print(last_light)
        time.sleep(1)  # Yellow light duration

        last_light = "▨ GREEN - Go"
        print(last_light)
        time.sleep(2)  # Green light duration

        print("Cycling again...\n")

    print(f"Final light glowing: {last_light}")

if __name__ == "__main__":
    traffic_light_control()
```

SCREEN SHOT OF OUTPUT



```
🟥 RED - Stop
🟨 YELLOW - Get Ready
🟩 GREEN - Go
Cycling again...

🟥 RED - Stop
🟨 YELLOW - Get Ready
🟩 GREEN - Go
Cycling again...

🟥 RED - Stop
🟨 YELLOW - Get Ready
🟩 GREEN - Go
Cycling again...

🟥 RED - Stop
🟨 YELLOW - Get Ready
🟩 GREEN - Go
Cycling again...

Final light glowing: 🟩 GREEN - Go
```

REFERENCE

The codes and methodologies used in this report were generated with the assistance of AI-based tools such as ChatGPT and other AI-powered platforms. These tools helped in data exploration, cleaning, and visualization techniques. Additional information and best practices were sourced from publicly available documentation and standard data science practices.