TRAFFIC MANAGEMENT PYTHON SCRIPT

Program No:1

```
import time

# Define the traffic light states
RED = "Red"
GREEN = "Green"

# Define the traffic light timings in seconds
GREEN_DURATION = 10
RED_DURATION = 5

# Define a function to control the traffic lights
def control_traffic_lights():
    while True:
        # Set the traffic light to green
        print("Traffic Light: " + GREEN)
        time.sleep(GREEN_DURATION)
        # Set the traffic light to red
        print("Traffic Light: " + RED)
        time.sleep(RED_DURATION)

if __name__ == "__main__":
    try:
        control_traffic_lights()
    except KeyboardInterrupt:
        print("Traffic Management System Stopped")
```

Program No:2

```
import random
import time

class TrafficLight:
    def __init__(self):
        self.state = "RED"

    def change_light(self):
        if self.state == "RED":
            self.state = "GREEN"
        else:
            self.state = "RED"

class Car:
    def __init__(self, id):
        self.id = id

    def move(self):
        print(f"Car {self.id} is moving.")

# Simulate traffic management
def main():
    traffic_light = TrafficLight()

    for i in range(10):
        # Change traffic light state every 5 seconds
        traffic_light.change_light()
        print(f"Traffic light is {traffic_light.state}")

        # Simulate cars moving when the light is green
        if traffic_light.state == "GREEN":
            num_cars = random.randint(1, 5)
            cars = [Car(j) for j in range(num_cars)]
            for car in cars:
                car.move()

        time.sleep(5)  # Sleep for 5 seconds

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

Program No:3

```
import pygame
import sys
import random

# Initialize Pygame
pygame.init()

# Constants
SCREEN_WIDTH = 800
SCREEN_HEIGHT = 600
BACKGROUND_COLOR = (255, 255, 255)
```

```
CAR_COLOR = (0, 0, 0)
INTERSECTION_COLOR = (150, 150, 150)
CAR_WIDTH = 20
CAR_HEIGHT = 40
CAR_SPEED = 3
INTERSECTION_WIDTH = 200
INTERSECTION_HEIGHT = 200

# Initialize the screen
screen = pygame.display.set_mode((SCREEN_WIDTH, SCREEN_HEIGHT))
pygame.display.set_caption("Traffic Management")

# Car class
class Car:
    def __init__(self, x, y, direction):
        self.x = x
        self.y = y
        self.direction = direction

    def move(self):
        if self.direction == "up":
            self.y -= CAR_SPEED
        elif self.direction == "down":
            self.y += CAR_SPEED
        elif self.direction == "left":
            self.x -= CAR_SPEED
        elif self.direction == "right":
            self.x += CAR_SPEED

    def draw(self):
        pygame.draw.rect(screen, CAR_COLOR, (self.x, self.y, CAR_WIDTH,
CAR_HEIGHT))

# Intersection class
class Intersection:
    def __init__(self, x, y):
        self.x = x
        self.y = y

    def draw(self):
        pygame.draw.rect(screen, INTERSECTION_COLOR, (self.x, self.y,
INTERSECTION_WIDTH, INTERSECTION_HEIGHT))

# Create cars
```

```
cars = [Car(SCREEN_WIDTH // 2 - CAR_WIDTH // 2, SCREEN_HEIGHT - CAR_HEIGHT, "up")]

# Create the intersection
intersection = Intersection(SCREEN_WIDTH // 2 - INTERSECTION_WIDTH // 2, SCREEN_HEIGHT // 2 - INTERSECTION_HEIGHT // 2)

clock = pygame.time.Clock()

while True:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()

    # Generate random cars
    if random.random() < 0.02:
        direction = random.choice(["up", "down", "left", "right"])
        if direction == "up":
            cars.append(Car(SCREEN_WIDTH // 2 - CAR_WIDTH // 2, 0, direction))
        elif direction == "down":
            cars.append(Car(SCREEN_WIDTH // 2 - CAR_WIDTH // 2, SCREEN_HEIGHT, direction))
        elif direction == "left":
            cars.append(Car(SCREEN_WIDTH, SCREEN_HEIGHT // 2 - CAR_HEIGHT // 2, direction))
        elif direction == "right":
            cars.append(Car(0, SCREEN_HEIGHT // 2 - CAR_HEIGHT // 2, direction))

    screen.fill(BACKGROUND_COLOR)
    intersection.draw()

    # Move and draw cars
    for car in cars:
        car.move()
        car.draw()

    # Remove cars that are out of bounds
    cars = [car for car in cars if 0 <= car.x <= SCREEN_WIDTH and 0 <= car.y <= SCREEN_HEIGHT]

    pygame.display.flip()
    clock.tick(60)
```