'''

Rules: We will use 30 seconds for each lane ( 4 lanes so trafiic lights will be 4 different sets)

The order will be 1 -> 2 -> 3 -> 4 -> 1 -> 2 and so on.

Soppose when the emergency vehicles comes, we will stop the current time and store that for whcih lane it was

continuing and the lane from where the mergency vehicle comes we will strat that lane without time till it passes

and then we will stop that lane and resume the one that we stopped earlier snd so on the cycle will contiue. When

the signal is red for a a partcular lane, we can allow the green for the pdestrians to cross for 10 seconds

when a partcuar signal turns green first 10 second pedestrains cross and then the vehicles are allowed.

Assumptions: 30 seconds for the emergecy vehicle to cross.

             10 seconds for each pedestraisn lane to cross

             30 seconds for the vehicles to cross

'''

import time

import random

class traffic:

    # constrcutor

    def \_\_init\_\_(self, lane\_no):

        self.lane\_no = lane\_no

        # initialze lane to 1

        self.current = 1

        self.timing = {i: 30 for i in range(1, lane\_no + 1)}

        self.pedestrian = {i: False for i in range(1, lane\_no + 1)}

    def simu(self):

        while True:

            self.traffic\_lights()

            self.emergency()

            self.people()

    def traffic\_lights(self):

        print("--------------------------------------- ")

        for lane in range(1, self.lane\_no + 1):

            if lane == self.current:

                print(f"Lane {lane}: Green")

            else:

                print(f"Lane {lane}: Red")

            if not self.pedestrian[lane] and lane != self.current:

                print(f"Lane {lane} Pedestrian Signal: Red")

            else:

                print(f"Lane {lane} Green for 10 seconds (Pedestrians Crossing)")

                #  waiting for allowing pedestrians to cross

                self.waiting()

        time.sleep(30)

        self.current = (self.current % self.lane\_no) + 1

    def emergency(self):

        if random.randint(1, 10) == 1:

            self.handle()

    def handle(self):

        e\_lane = random.randint(1, self.lane\_no)

        print(f"Emergency vehicle approaching from Lane {e\_lane}!")

        #  allow the emergency vehicle to pass

        self.timing[self.current] = 0

        while self.timing[self.current] < 30:

            print(f"Lane {self.current}: Time left {30 - self.timing[self.current]} seconds (Emergency Vehicle Passing)")

            time.sleep(1)

            self.timing[self.current] += 1

        # Reset timings

        self.timing[self.current] = 30

    def people(self):

        if random.randint(1, 15) == 1 and not self.pedestrian[self.current]:

            self.ACT\_PED()

    def ACT\_PED(self):

        #   pedestrian signal for the current lane

        self.pedestrian[self.current] = True

        print(f"Lane {self.current} Pedestrian Signal: Green for 10 seconds (Pedestrians Crossing)")

        time.sleep(10)

        self.pedestrian[self.current] = False

    def waiting(self):

        #   traffic to clear

        while self.timing[self.current] < 30:

            print(f"Lane {self.current}: Time left {30 - self.timing[self.current]} seconds (Traffic to Clear)")

            time.sleep(1)

            self.timing[self.current] += 1

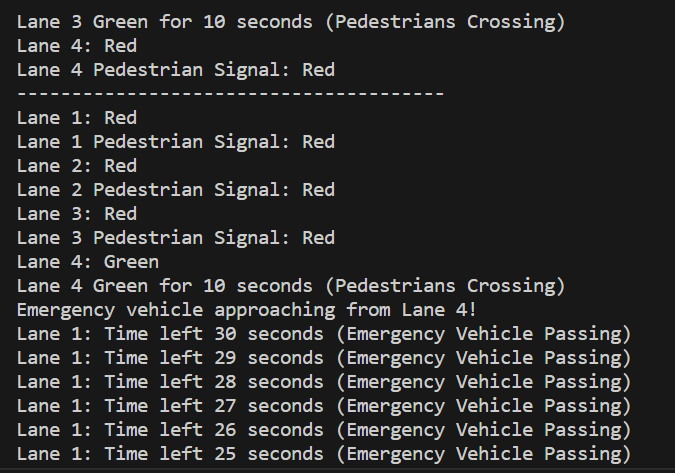
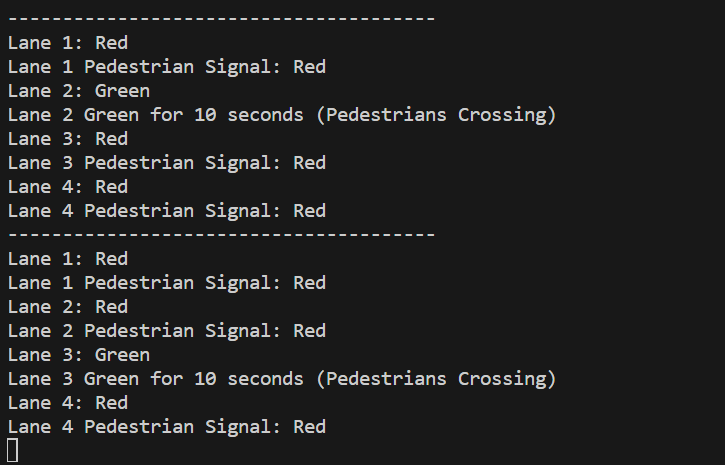
if \_\_name\_\_ == "\_\_main\_\_":

    lane\_no = 4

    simulator = traffic(lane\_no)

    simulator.simu()

**OUTPUT**

****