```
import socket
import json
import random
import time
import threading

SERVER_IP = "127.0.0.1"
PORT = 5000

# --- Server code ---
def run_server():
    server_sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    server_sock.bind((SERVER_IP, PORT))
    server_sock.listen(1)
    print(f"Server listening on {SERVER_IP}:{PORT}")

    conn, addr = server_sock.accept()
    print("Connected by", addr)

    data = conn.recv(1024).decode()
    if data:
        print("Received:", json.loads(data))

    conn.close()
    server_sock.close()

# --- Client code ---
def run_client():
```

```
    time.sleep(1)  # wait for server to start
    sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    print("Connecting...")
    sock.connect((SERVER_IP, PORT))
    print("Connected to receiver")

    count = random.randint(0, 30)

    if count >= 20:
        level = "HIGH"
    elif count > 10:
        level = "MEDIUM"
    else:
        level = "LOW"

    data = {
        "signal_id": "TS01",
        "vehicle_count": count,
        "congestion": level
    }

    sock.send(json.dumps(data).encode())
    print("Sent:", data)

    time.sleep(2)
    sock.close()

# --- Run both server and client ---
```

```
server_thread = threading.Thread(target=run_server)
client_thread = threading.Thread(target=run_client)

server_thread.start()
client_thread.start()

server_thread.join()
client_thread.join()
# ========================================
# Ambulance Routing Effectiveness System
# First-Year Friendly Python Project
# ========================================

# ------------------------------
# Hospital Data (Predefined)
# ------------------------------
hospitals = [
  {
    "name": "City Government Hospital",
    "distance_km": 6,
    "icu_available": True
  },
  {
    "name": "Private Heart Care Hospital",
    "distance_km": 8,
    "icu_available": True
  },
  {
```

```
    "name": "Local Clinic",
    "distance_km": 3,
    "icu_available": False
  }
]

# ------------------------------
# Road & Traffic Data (Predefined)
# ------------------------------
roads = [
  {
    "road_name": "Main Road",
    "traffic_level": 3,        # 1=Low, 2=Medium, 3=High
    "road_condition": 2,       # 1=Good, 2=Average, 3=Bad
    "infrastructure": 1,       # 1=Good, 2=Limited
    "unpredictable_events": 2   # 0=None, 1=Sometimes, 2=Frequent
  },
  {
    "road_name": "Bypass Road",
    "traffic_level": 1,
    "road_condition": 1,
    "infrastructure": 2,
    "unpredictable_events": 1
  },
  {
    "road_name": "Market Area Road",
    "traffic_level": 3,
    "road_condition": 3,
```

```
        "infrastructure": 2,
        "unpredictable_events": 2
    }
]

# ------------------------------
# Function to calculate route score
# Lower score = better route
# ------------------------------
def calculate_route_score(hospital, road):
    score = 0

    # Distance impact
    score += hospital["distance_km"] * 2

    # Traffic impact
    score += road["traffic_level"] * 5

    # Road condition impact
    score += road["road_condition"] * 4

    # Infrastructure impact
    score += road["infrastructure"] * 3

    # Unpredictable road behavior
    score += road["unpredictable_events"] * 6

    # ICU priority
```

```
    if not hospital["icu_available"]:
        score += 20

    return score

# ------------------------------
# Find Best Route
# ------------------------------
best_option = None
lowest_score = float("inf")

for hospital in hospitals:
    for road in roads:
        score = calculate_route_score(hospital, road)

        if score < lowest_score:
            lowest_score = score
            best_option = {
                "hospital": hospital["name"],
                "road": road["road_name"],
                "score": score
            }

# ------------------------------
# Output Result
# ------------------------------
print("🚑 AMBULANCE ROUTING DECISION 🚑\n")
print("Best Hospital :", best_option["hospital"])
```

```
print("Best Road     :", best_option["road"])
print("Route Score   :", best_option["score"])

print("\n(Note: Lower score means faster and safer route)")
```