Python Assignment: Mystery Delivery System

Scenario: You are designing a logistics simulator for a fictional delivery company called **FastBox**. There are multiple warehouses, delivery agents, and packages, all interacting under random conditions. Your goal: simulate one day of operations and produce a report showing which packages were delivered, total distance traveled by each agent, and the most efficient agent.

Input Data (data.json):

```
{
  "warehouses": {
    "W1": [0, 0],
    "W2": [50, 75],
    "W3": [100, 25]
  },
  "agents": {
    "A1": [5, 5],
    "A2": [60, 60],
    "A3": [95, 30]
  },
  "packages": [
    {"id": "P1", "warehouse": "W1", "destination": [30, 40]},
    {"id": "P2", "warehouse": "W2", "destination": [70, 90]},
    {"id": "P3", "warehouse": "W3", "destination": [105, 20]},
    {"id": "P4", "warehouse": "W1", "destination": [10, 10]},
    {"id": "P5", "warehouse": "W2", "destination": [40, 80]}
  ]
}
```

Tasks:

1. Read and parse the JSON file manually.
2. Assign each package to the nearest agent based on Euclidean distance from agent to warehouse.
3. Simulate delivery: agent picks up packages from warehouse and delivers to destination. Compute total distance traveled.
4. Generate a report:

```
{
  "A1": {"packages_delivered": 2, "total_distance": 85.32, "efficiency": 42.66},
  "A2": {"packages_delivered": 2, "total_distance": 120.12, "efficiency": 60.06},
  "A3": {"packages_delivered": 1, "total_distance": 50.00, "efficiency": 50.00},
  "best_agent": "A1"
}
```

5. Save the report to `report.json`.

Bonus (Optional):

- Random delivery delays
- Visualize routes in ASCII
- Handle new agent joining mid-day
- Export top performer to CSV

Evaluation Criteria:

Criteria	Weight	Description
JSON parsing	10%	Reads data correctly
Distance calculation	20%	Uses correct Euclidean logic
Agent-package assignment	25%	Correct nearest-agent mapping
Simulation & report	25%	Outputs valid report
Code clarity & comments	10%	Understandable logic
Bonus creativity	10%	Adds realistic extensions

Notes:

- Test your code with different JSON inputs.
- Make sure total packages delivered matches total packages.
- Comment your code for clarity.