Assignment Question 5: Cargo Drone Traffic Controller

Name: Santhana Sreenivasa S
Roll Number: ME24B1017
Branch: B.Tech Mechanical
Subject: Elementary Data Structures and Logical Thinking (EDSLT)
Language Used: C Programming
Submission Date: 28-04-25

Objective:

The goal of this assignment was to practically apply essential data structures like queues, stacks, arrays, and various types of linked lists to simulate a real-world challenge: managing a fleet of cargo drones in a smart logistics airspace.

Through this simulation, I explored how mechanical engineering problems, such as delivery management, emergency rerouting, and drone maintenance, can be efficiently solved using computer science techniques.

Scenario Summary:

We are tasked with designing a Cargo Drone Traffic Controller system that handles various operations including:

- Accepting delivery requests
- Prioritizing urgent deliveries
- Logging completed missions
- Tracking overloaded or emergency-condition drones

Each type of operation is mapped to a different data structure based on what fits best for that particular task.

Approach and Logic Explained:

1. Delivery Requests and Urgent Dispatch — Using Queue and Stack

Logic:
Delivery requests naturally come in one after another, so a Queue (First-In-First-Out) is perfect for this. However, emergencies arise (like fuel shortages), needing immediate attention. To model this, I used a Stack (Last-In-First-Out) to prioritize the most recent urgent requests.

Implementation Steps:

- Add normal deliveries ("Food", "Medicine", "Fuel", etc.) into a queue.
- Urgent deliveries are dequeued from the queue and pushed into a stack.

- Pop from the stack for urgent dispatches, ensuring latest emergencies are handled first.

Why it Works:
This system reflects how real control centers work, where sudden emergencies override routine tasks.

---

2. Flight Log Unit — Fixed-Size Array with Archiving

Logic:
Every completed delivery must be logged. To manage memory efficiently, I used a fixed-size array that holds the six most recent deliveries. If it fills up, the oldest entry is archived (overwritten) to make space for new ones.

Implementation Steps:

- Insert each completed delivery into the array.
- When full, shift all entries left and insert the new delivery at the end.

Why it Works:
In aviation and logistics, it's common to keep only the latest records easily accessible for quick reference and audits.

---

3. Overloaded Drone Tracker — Singly and Doubly Linked Lists

Logic:
Drones that face issues like overloading are first tracked using a singly linked list. After maintenance or recalibration, they move to a doubly linked list to allow forward and backward inspection.

Implementation Steps:

- Add overloaded drones (e.g., "Drone3", "Drone6") to a singly linked list.
- Upon servicing, remove from the singly linked list and add into a doubly linked list.
- Traverse forward to see latest repairs and backward to audit older services.

Why it Works:
This separation makes it easier for maintenance teams to track drone status transitions over time.

---

4. Emergency Rerouting — Circular Linked List

Logic:
In critical conditions like sudden storms or airspace congestion, rerouting needs continuous checking. A circular linked list models this well, as drones can loop through reroute options until safe paths are found.

Implementation Steps:

- Insert drones needing emergency reroute ("Drone1", "Drone4") into a circular linked list.
- Traverse the list repeatedly to simulate ongoing rerouting attempts.

Why it Works:
Circular structure allows indefinite cycling through emergency response strategies until resolution.

Interactive Implementation:

In this project, the user can:

- Add normal delivery requests
- Dispatch urgent deliveries
- Maintain a flight delivery log
- Track overloaded and repaired drones
- Handle rerouted emergency drones

The modular structure makes the controller robust and relatable to real-world drone logistics systems.

Overall Learning and Conclusion:

Working on this assignment helped me think from the perspective of a systems engineer handling real-time logistics problems. It showed me how fundamental data structures, beyond academic exercises, directly model real-world mechanical and logistical scenarios.

I gained hands-on experience in C programming while also understanding how data handling, maintenance tracking, and dynamic rerouting could be effectively built into drone management systems. It was an excellent blend of theoretical learning and practical engineering thinking.