Assignment Question 5: Cargo Drone Traffic Controller

Name: Kovuru Girish Chandra
Roll Number: ME24B1048
Branch: B.Tech Mechanical
Subject: Elementary Data Structures and Logical thinking(EDSLT)
Language Used: C Programming
Submission Date: 23-04-25

Objective:

The objective of this assignment is to apply key data structures such as queues, stacks, arrays, and linked lists (singly, doubly, circular) to simulate a real-world scenario — managing and controlling a fleet of cargo drones in a smart logistics airspace.

This simulation demonstrates how mechanical engineering problems like delivery management, emergency rerouting, and maintenance tracking can be efficiently solved using computer science principles.

Scenario Summary:

We are designing a Cargo Drone Traffic Controller that handles different types of drone-related operations like:

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

Each task was mapped to a suitable data structure that best fits the nature of that operation.

Approach and Logic Explained:

1. Delivery Requests and Urgent Dispatch – Queue and Stack

Logic:
In real life, delivery requests arrive sequentially, so we use a Queue (FIFO) to store them. However, some deliveries (like fuel or medicine) are more urgent and need to be dispatched immediately. For this, we simulate a Stack (LIFO) to reverse the order and prioritize the most recent urgent tasks.

Implementation Steps:

- Enqueue delivery requests like "Food", "Medicine", "Fuel", etc.
- Dequeue them and push onto a stack.
- Pop from the stack to simulate urgent dispatch in LIFO order.

Why it works:
This mimics a control room prioritizing emergency deliveries that appeared last but must go first due to urgency.

2. Flight Log Unit – Fixed Size Array with Archiving

Logic:
Drones log each completed delivery. We use a fixed-size array (6 slots) to keep track of the most recent 6 deliveries. Once the array is full, the oldest delivery is archived, and the new one takes its place.

Implementation Steps:

- Log each delivery one by one.
- If the log is full, shift entries to remove the oldest and insert the new one at the end.

Why it works:
In aviation and logistics, only recent data is actively monitored while older logs are moved to storage for record-keeping or compliance.

3. Overloaded Drone Tracker – Singly and Doubly Linked Lists

Logic:
Overloaded drones (due to excess weight or system failure) are tracked using a singly linked list. Once they are repaired or recalibrated, they move to a doubly linked list to allow bidirectional inspection.

Implementation Steps:

- Add overloaded drones (e.g., Drone3, Drone6) to a singly linked list.
- Once fixed, remove them from that list and add to the doubly linked list.
- Traverse forward to view latest repairs, backward to audit older services.

Why it works:
This structure helps maintenance engineers maintain a simple log of overloaded drones and review serviced ones both forward and backward.

4. Emergency Rerouting – Circular Linked List

Logic:
In case of extreme weather or air traffic conflicts, emergency drones need to loop through rerouting logic continuously. This is efficiently simulated with a circular linked list.

Implementation Steps:

- Add emergency drones (e.g., Drone1, Drone4) to the circular list.
- Traverse the list in a loop to simulate repeated route-checks.

Why it works:
The circular nature allows emergency drones to cycle through possible paths until the route is cleared or redirected.

Interactive Implementation Using Switch Case

The entire system is made interactive using a menu-based switch-case in C. This makes it more practical and user-driven. The user can:

- Add delivery requests
- Dispatch urgent items
- Log completed deliveries
- Track overloaded drones
- Update maintenance
- Manage emergency situations

This modular approach makes the system scalable and relatable to real drone logistics platforms.

Overall Learning and Conclusion:

This assignment gave me an opportunity to think like a system designer for real-world drone operations. I was able to connect how fundamental data structures — which we often learn theoretically — can be directly applied to mechanical systems and logistics management.

This exercise improved my understanding of not just programming in C, but also how data handling, routing, and system maintenance could be modeled efficiently. It was a good combination of engineering logic and structured coding.