CARGO DRONE TRAFFIC CONTROLLER SIMULATION USING DATA STRUCTURES

ELEMENTRY DATA STRUCTURE AND LOGICAL THINKINIG

NAME: HADIL MALIK BASHA F
ROLL NO:ME24B1073
DATE: April 19, 2025

PROBLEM STATEMENT

In modern logistics, managing cargo drone operations autonomously is critical for:

- Efficient delivery scheduling
- Handling urgent shipments
- Tracking maintenance and emergencies
- Maintaining flight logs

This simulation models a drone traffic controller using data structures to:

1. Process delivery requests (FIFO)
2. Dispatch urgent deliveries (LIFO)
3. Maintain limited flight logs
4. Track overloaded drones
5. Monitor emergency drones in real-time

DESIGN EXPLANATION

1. **Queue (Array-based)**
    1. Manages delivery requests in FIFO order.
    2. First-in requests are processed first.
2. **Stack (Array-based)**
    1. Handles urgent deliveries in LIFO order.
    2. Ensures priority dispatch.
3. **Fixed-size Array**
    1. Stores the 6 most recent flight logs.
    2. Overwrites oldest logs when full.
4. **Singly Linked List**
    1. Tracks overloaded drones needing repair.
    2. Efficient for insertions/deletions.
5. **Doubly Linked List**
    1. Maintains serviced drones with bidirectional traversal.
    2. Allows forward/backward inspection.
6. **Circular Linked List**
    1. Continuously monitors emergency drones.
    2. Simulates round-robin priority checks.

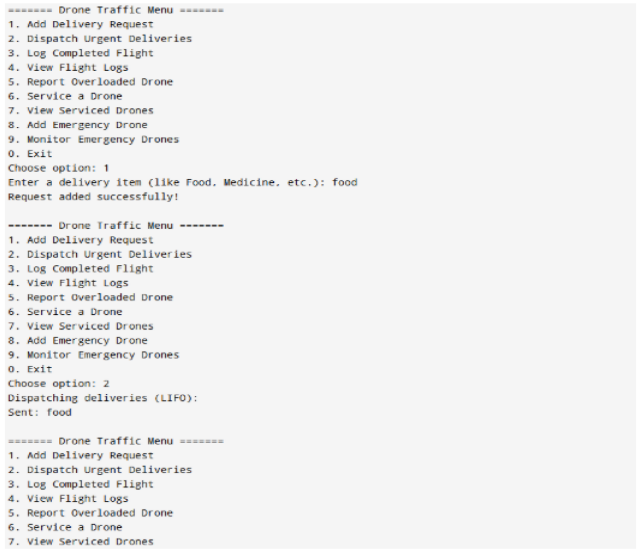
LOGIC OF THE CODE

1. **Delivery Management**:
   - Add requests to queue (addDeliveryRequest)
   - Dispatch urgent deliveries via stack (dispatchUrgentDeliveries)
2. **Logging**:
   - Record completed flights (logFlightCompletion)
   - View logs (showFlightLogs)
3. **Maintenance**:
   - Report overloaded drones (reportOverloadedDrone)
   - Service drones (serviceNextDrone)
   - View serviced drones (showServicedDrones)
4. **Emergency Handling**:
   - Add drones to emergency loop (addEmergencyDrone)
   - Monitor emergencies (monitorEmergencyDrones)

SUMMARY OF DATA STRUCTURES AND USAGE

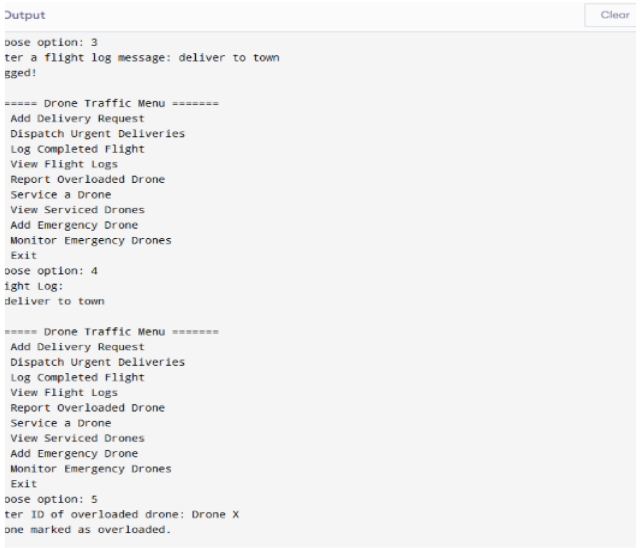
Data Structure	Purpose	C Implementation
Queue (Array)	Delivery requests (FIFO)	deliveryQueue[], front/rear
Stack (Array)	Urgent dispatches (LIFO)	urgentStack[], stackTop
Fixed Array	Flight logs (6 max)	flightLogs[], logCount
Singly Linked List	Overloaded drones	Drone struct with next
Doubly Linked List	Serviced drones	Serviced struct with prev/next
Circular Linked List	Emergency monitoring	Emergency struct with circular next

OUTPUT

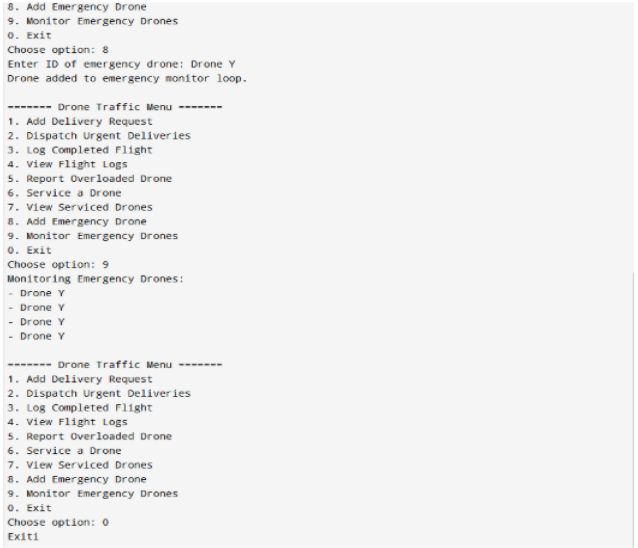


```
======= Drone Traffic Menu =======
1. Add Delivery Request
2. Dispatch Urgent Deliveries
3. Log Completed Flight
4. View Flight Logs
5. Report Overloaded Drone
6. Service a Drone
7. View Serviced Drones
8. Add Emergency Drone
9. Monitor Emergency Drones
0. Exit
Choose option: 1
Enter a delivery item (like Food, Medicine, etc.): food
Request added successfully!

------- Drone Traffic Menu -------
1. Add Delivery Request
2. Dispatch Urgent Deliveries
3. Log Completed Flight
4. View Flight Logs
5. Report Overloaded Drone
6. Service a Drone
7. View Serviced Drones
8. Add Emergency Drone
9. Monitor Emergency Drones
0. Exit
Choose option: 2
Dispatching deliveries (LIFO):
Sent: food

======= Drone Traffic Menu =======
1. Add Delivery Request
2. Dispatch Urgent Deliveries
3. Log Completed Flight
4. View Flight Logs
5. Report Overloaded Drone
6. Service a Drone
7. View Serviced Drones
```

```
Output                                                    Clear

. Monitor Emergency Drones
. Exit
hoose option: 6
o overloaded drones right now.

------ Drone Traffic Menu -------
. Add Delivery Request
. Dispatch Urgent Deliveries
. Log Completed Flight
. View Flight Logs
. Report Overloaded Drone
. Service a Drone
. View Serviced Drones
. Add Emergency Drone
. Monitor Emergency Drones
. Exit
hoose option: 7
erviced Drones (Forward):
erviced Drones (Backward):

------ Drone Traffic Menu -------
. Add Delivery Request
. Dispatch Urgent Deliveries
. Log Completed Flight
. View Flight Logs
. Report Overloaded Drone
. Service a Drone
. View Serviced Drones
. Add Emergency Drone
. Monitor Emergency Drones
. Exit
hoose option: 8
nter ID of emergency drone: Drone Y
rone added to emergency monitor loop.
```



```
Output                                                    Clear

oose option: 3
ter a flight log message: deliver to town
gged!

===== Drone Traffic Menu =======
 Add Delivery Request
 Dispatch Urgent Deliveries
 Log Completed Flight
 View Flight Logs
 Report Overloaded Drone
 Service a Drone
 View Serviced Drones
 Add Emergency Drone
 Monitor Emergency Drones
 Exit
oose option: 4
ight Log:
deliver to town

===== Drone Traffic Menu =======
 Add Delivery Request
 Dispatch Urgent Deliveries
 Log Completed Flight
 View Flight Logs
 Report Overloaded Drone
 Service a Drone
 View Serviced Drones
 Add Emergency Drone
 Monitor Emergency Drones
 Exit
oose option: 5
ter ID of overloaded drone: Drone X
one marked as overloaded.
```



```
8. Add Emergency Drone
9. Monitor Emergency Drones
0. Exit
Choose option: 8
Enter ID of emergency drone: Drone Y
Drone added to emergency monitor loop.

------- Drone Traffic Menu -------
1. Add Delivery Request
2. Dispatch Urgent Deliveries
3. Log Completed Flight
4. View Flight Logs
5. Report Overloaded Drone
6. Service a Drone
7. View Serviced Drones
8. Add Emergency Drone
9. Monitor Emergency Drones
0. Exit
Choose option: 9
Monitoring Emergency Drones:
- Drone Y
- Drone Y
- Drone Y
- Drone Y

------- Drone Traffic Menu -------
1. Add Delivery Request
2. Dispatch Urgent Deliveries
3. Log Completed Flight
4. View Flight Logs
5. Report Overloaded Drone
6. Service a Drone
7. View Serviced Drones
8. Add Emergency Drone
9. Monitor Emergency Drones
0. Exit
Choose option: 0
Exiti
```