# Delivery Drone Dispatch and Management System

Name: CH Vamsi Praneeth

Roll Number: ME24I1005

Date: 15th April

Batch: 4

Question No: 5

## Problem Being Solved

This assignment simulates the operations of a delivery drone dispatch system using fundamental data structures. The system manages delivery requests, logs completed missions, handles overloaded and serviced drones, and manages emergency rerouting operations.

## Key Objectives

- Efficiently process and prioritize delivery requests.  
- Maintain a fixed-size log of completed deliveries.  
- Track drones that are overloaded and require servicing.  
- Manage a list of drones available for emergency redeployment.

## Design Explanation

The data structures were selected based on their operational advantages:  
- Queue for FIFO request processing.  
- Stack for LIFO priority dispatching.  
- Array for fixed-size circular flight log.  
- Singly linked list for dynamically tracking overloaded drones.  
- Doubly linked list for easy forward and backward tracking of serviced drones.  
- Circular linked list for endless loop traversal in emergency rerouting.

## Efficiency of Data Structures

Using the above data structures provides both time and space efficiency. Queues and stacks are ideal for the ordering of tasks, linked lists help manage dynamic datasets like drone conditions, and circular structures allow continuous cycling through emergency resources without reinitialization.

## Logic of the Code

The program initializes multiple data structures, enqueues delivery requests, processes them in reverse using a stack, logs deliveries in a circular array, and maintains states of drones using linked lists. Each module represents a real-world counterpart in a drone dispatch and logistics scenario.

## Variables and Functions Used

- Structs: Queue, Stack, SLLNode, DLLNode, CLLNode  
- Arrays: requests, deliveries, flightlog  
- Functions: initQueue, enqueue, dequeue, initStack, push, pop, logDelivery, insertSLL, deleteSLL, insertDLL, traverseDLL, insertCLL, traverseCLL



