# Earthquake Rescue Robot Coordinator

Name: N. Vamsi

Roll Number: ME24I1032

Date: 15th April

## Problem Being Solved

This assignment simulates an Earthquake Rescue Coordination System using a combination of data structures: queue, stack, singly linked list, doubly linked list, and circular linked list. The goal is to efficiently manage the sequence of rescue missions, track urgent tasks, handle the repair workflow of damaged robots, and log operations for future reference.

## Key Objectives

- Implement mission queuing and execution.  
- Handle urgent rescue priorities using stack.  
- Track damaged and repaired robots using linked lists.  
- Manage redeployment cycles through circular linked lists.  
- Maintain a circular log of operations.

## Design Explanation

Each data structure was selected based on the operational requirements:  
- Queue for FIFO mission management.  
- Stack for LIFO urgent mission handling.  
- Singly linked list to record damaged robots (simple, sequential insertions).  
- Doubly linked list for repaired robots to allow forward and backward traversal.  
- Circular linked list for infinite redeployment cycles.  
- Circular array for efficient mission log rotation.

## Efficiency of Data Structures

The chosen data structures are optimal:  
- Queue allows fair handling of rescue missions.  
- Stack helps in immediate response to urgent calls.  
- Linked lists simplify robot tracking without memory overhead.  
- Circular structures avoid the need to reset or reinitialize after each cycle.

## Logic of the Code

The program begins with menu-driven options for managing missions and robot statuses. Functions are used to insert and remove data in appropriate structures, simulating a real-time rescue coordination system.

## Variables and Functions Used

- Structs: Queue, Stack, Singlynode, Doublynode, Circularnode  
- Global Pointers: front, rear, top, damagehead, repairhead, repairtail, circularhead, circulartail  
- logarray: Circular log storage array  
- Functions: enqueue, dequeue, push, pop, logMission, insertDamagedRobot, deleteDamagedRobot, insertRepairedRobot, redeployDamagedRobot, and multiple display/traversal functions for each data structure



