



# Hospital Emergency Queue System using Doubly Linked List

Ibad Ur Rahman | BS AI-B

## Problem Statement

A small hospital wants to manage Emergency Room (ER) patients using a flexible queue system. Because patient priorities change (critical arrivals, discharges, etc.), a Doubly Linked List is used for dynamic insertion and deletion. Each node represents one patient, storing patientID, prev and next pointers.

The system supports:

1. Inserting at beginning (critical patients)
2. Inserting at end (normal patients)
3. Inserting at a specific position
4. Deleting from beginning (treated patients)

## Proposed Solution

Implemented a C++ doubly linked list with functions:

- insertAtBeginning(int id)
- insertAtEnd(int id)
- insertAtPosition(int id, int pos)
- deleteFromBeginning()

Each operation updates head, tail, and pointer links correctly.

If inserted at a position greater than the list length, the patient is added to the end of the queue.

## Graphical Representation

Visual diagrams showing list evolution after each operation:

Inserting 101 at end:

NULL ← 101 → NULL

Inserting 102 at end:

NULL ← 101 → ← 102 → NULL

Inserting 200 at end:

NULL ← 200 → ← 101 → ← 102 → NULL

Inserting 150 at position 2:

NULL ← 200 → ← 150 → ← 101 → ← 102 → NULL

Deleting node from beginning:

NULL ← 150 → ← 101 → ← 102 → NULL

Inserting 300 at end:

NULL ← 150 → ← 101 → ← 102 → ← 300 → NULL

## Final Output

Head: 150

Tail: 300

Forward Traversal: 150 → 101 → 102 → 300

Backward Traversal: 300 → 102 → 101 → 150