

PROBLEM

A small hospital is managing patients in the Emergency Room (ER). Because patient priority can change quickly (new critical patients arrive, discharged patients leave, etc.), the hospital wants a flexible system to keep track of the current patients in the ER.

Each node in the list represents one patient, and stores:

- patientID (integer)
- pointers to previous and next patients

The following operations on this ER queue:

1. Add a new patient at the beginning of the list (Insert from beginning)
2. Add a new patient at the end of the list (insert from ending)
3. Add a new patient at a specific position in the list (insert at specific position)
4. Remove a patient from the beginning of the list (Delete from beginning)



MUHAMMAD HADI
ROLL NO : 21



PROPOSED SOLUTION

The solution uses a Doubly Linked List implemented in C++. The Node struct holds patientID and pointers to prev and next. The DoublyLinkedList class manages head, tail, and size, with methods for insertion and deletion. Edge cases are handled: empty list, single node, position 1, and positions beyond length (append to end). The dry run simulates operations and prints the list graphically.

GRAPHICAL REPRESENTATION OF OPERATIONS

Initial Empty List

NILL → NILL

After insertAtEnd(101)

101 → NILL

After insertAtBeginning(182)

182 → 182 → 101 → NILL

After insertAtPosition(203, 2)

203 → 121 → 182 → NILL

After deleteFromBeginning

199 ← 121 → 183 → NILL

After insertAtEnd(198)

198 ← 101 ← 183 → 300 → NILL