### Sir I have made the poster and also have dry run the code but on a same file due to graphical representation of linkedlist because it was also the part of dry run

# Doubly Linked List — ER Patient Queue (Program Execution Trace)

Name: Ahmad Faraz Khan

Roll No: 4

## Problem Statement

This program demonstrates a doubly linked list for managing an Emergency Room (ER) patient queue. The code performs insertions at the beginning and end, deletions from the beginning, and insertion at specific positions. Each step below shows how the linked list changes as the program executes.

## Proposed Solution

The program defines a Node class with data, prev, and next pointers. The Doublylist class contains functions to insert and delete patients and to display the list. The dry run below follows the exact sequence of operations written in the C++ main() function.

## Dry Run (Based on Program Operations)

1. 1. insertAtBeginning(5)
2. HEAD = tail =newnode

List: NULL <-> [5] <-> NULL  
Head: 5 Tail: 5

1. 2. insertAtBeginning(6)
2.  New node (6) created
3.  6 becomes new head
4.  6 → 5

List: NULL <-> [6] <-> [5] <-> NULL  
Head: 6 Tail: 5

1. 3. insertAtBeginning(7)
2. New node 7 created before 6
3. It will be new head

List: NULL <-> [7] <-> [6] <-> [5] <-> NULL  
Head: 7 Tail: 5

1. 4. insertAtBeginning(8)
2. New node created before 7
3. It will be new head

List: NULL <-> [8] <-> [7] <-> [6] <-> [5] <-> NULL  
Head: 8 Tail: 5

1. 5. deleteAtBeginning()

Deletes first node (8)

The new head will be 7  
List: NULL <-> [7] <-> [6] <-> [5] <-> NULL  
Head: 7 Tail: 5

1. 6. insertAtEnd(9)

Add at last position pointing to null

List: NULL <-> [7] <-> [6] <-> [5] <-> [9] <-> NULL  
Head: 7 Tail: 9

1. 7. insertAtEnd(10)

List: NULL <-> [7] <-> [6] <-> [5] <-> [9] <-> [10] <-> NULL

Add at last pos ppinting to null  
Head: 7 Tail: 10

1. 8. insertAtPosition(4, 3)

Add at specific pos for emergemcy patient

Inserts 4 at position 3  
List: NULL <-> [7] <-> [6] <-> [4] <-> [5] <-> [9] <-> [10] <-> NULL  
Head: 7 Tail: 10

1. 9. insertAtPosition(1, 10)
2. Invalid position (greater than length). Displayed message: 'invalid position:'  
   List remains unchanged:  
   NULL <-> [7] <-> [6] <-> [4] <-> [5] <-> [9] <-> [10] <-> NULL  
   Head: 7 Tail: 10
3. Final State of the List
4. Head Patient ID: 7
5. Tail Patient ID: 10
6. Forward Traversal (head → tail): 7 -> 6 -> 4 -> 5 -> 9 -> 10
7. Backward Traversal (tail → head): 10 -> 9 -> 5 -> 4 -> 6 -> 7
8. Notes