Dry Run of Deque using Doubly Linked List

Step 1: Create Empty Deque

- Constructor sets front=null, rear=null
- Queue is empty

Diagram:
front=null, rear=null

Step 2: insertAtRear(10)

- New node(10) created
- Queue was empty → front=rear=node(10)

Diagram:
front → [10] ← rear

Step 3: insertAtFront(5)

- New node(5) created
- node(5).next = front (10)
- front.prev = node(5)
- front = node(5)

Diagram:
front → [5] ↔ [10] ← rear

Step 4: insertAtRear(20)

- New node(20) created
- rear.next = node(20)
- node(20).prev = rear (10)
- rear = node(20)

Diagram:
front → [5] ↔ [10] ↔ [20] ← rear

Step 5: getFront() & getRear()

- front.item = 5
- rear.item = 20
- Output: Front=5, Rear=20

Step 6: deleteAtFront()

- Delete node(5)
- front = front.next (10)
- front.prev = null

Diagram:

front → [10] ↔ [20] ← rear

Step 7: deleteAtRear()

- Delete node(20)
- rear = rear.prev (10)
- rear.next = null

Diagram:
front → [10] ← rear

Final State

- Queue has one element: [10]
- If we delete again → front=null, rear=null (empty deque).