DAY 18

// Implementing Queue using Linkedlist

class Node {

int data;

Node next;

Node(int x) {

data = x;

next = null;

}

}

class QueueLinkedList {

Node front, rear;

Node createNode(int x) {

Node np = new Node(x);

return np;

}

void enqueue(int x) {

Node nptr = createNode(x);

if(rear == null) {

front = rear = nptr;

}

else {

rear.next = nptr;

rear = nptr;

}

}

void dequeue() {

if(front == null) {

System.out.println("No node to delete");

return;

}

System.out.println("Deleted item: " + front.data);

if(front == rear)

{

front = rear = null;

return;

}

front = front.next;

}

void display() {

if(front == null) {

System.out.println("Queue Empty");

return;

}

System.out.println("Queue elements are: ");

Node temp = front;

while(temp != null) {

System.out.print(temp.data + " ");

temp = temp.next;

}

System.out.println();

}

public static void main(String[] args) {

QueueLinkedList li = new QueueLinkedList();

li.front = li.rear = null;

li.enqueue(10);

li.display();

li.enqueue(20);

li.display();

li.enqueue(30);

li.display();

li.dequeue();

li.display();

}

}