```java
package linkedlist;

// Java program for linked-list implementation of queue

// A linked list (LL) node to store a queue entry
class QNode {
    int key;
    QNode next;

    // constructor to create a new linked list node
    public QNode(int key)
    {
        this.key = key;
        this.next = null;
    }
}

// A class to represent a queue
// The queue, front stores the front node of LL and rear stores the
// last node of LL
class Queue {
    QNode front, rear;

    public Queue()
    {
        this.front = this.rear = null;
    }

    // Method to add an key to the queue.
    void enqueue(int key)
    {

        // Create a new LL node
        QNode temp = new QNode(key);

        // If queue is empty, then new node is front and rear both
        if (this.rear == null) {
            this.front = this.rear = temp;
            return;
        }

        // Add the new node at the end of queue and change rear
        this.rear.next = temp;
        this.rear = temp;
    }

    // Method to remove an key from queue.
    void dequeue()
    {
        // If queue is empty, return NULL.
        if (this.front == null)
            return;

        // Store previous front and move front one node ahead
        QNode temp = this.front;
        this.front = this.front.next;

        // If front becomes NULL, then change rear also as NULL
        if (this.front == null)
            this.rear = null;
    }
}

// Driver class
  class Test33 {
    public static void main(String[] args)
```

```
    {
        Queue q = new Queue();
        q.enqueue(10);
        q.enqueue(20);
        q.dequeue();
        q.dequeue();
        q.enqueue(30);
        q.enqueue(40);
        q.enqueue(50);
        q.dequeue();
        System.out.println("Queue Front : " + q.front.key);
        System.out.println("Queue Rear : " + q.rear.key);
    }
}
// This code is contributed by Gaurav Miglani
```