Complete the code foe implementing Circular Queue using Linked List

```
class Node {
int data;
Node next;

public Node(int data) {
this.data = data;
this.next = null;
}
}

class CircularQueue {
private Node front;
private Node rear;
private final int maxSize;

public CircularQueue() {
this.front = null;
this.rear = null;

}

public boolean isEmpty() {
// Return true if the circular queue is empty, false otherwise
}

public boolean isFull() {
```

```
// Return true if the circular queue is full, false otherwise
}

public int getSize() {
// Return the current size of the circular queue
}

public void enqueue(int data) {
// Add a new element with the given data to the end of the circular queue
// If the circular queue is full, print an error message
// Update the front and rear pointers accordingly
}

public int dequeue() {
// Remove and return the element at the front of the circular queue
// If the circular queue is empty, print an error message and return -1
// Update the front and rear pointers accordingly
}

public void display() {
// Display the elements in the circular queue
// If the circular queue is empty, print an appropriate message
}
}

public class Demo {
public static void main(String[] args) {
CircularQueue queue = new CircularQueue();
//Complete Main Class
```

```
}
}
```

Note:

Size of circular queue should be fix (5 nodes only) if enqueuing more than 5 values program shows error message that queue is full