

```
package structure;
```

```
class ListElem<T> {
```

```
    public ListElem<T> prev;  
    public ListElem<T> next;  
    public T data;
```

```
    ListElem() {  
        prev = next = null;  
        data = null;  
    }
```

```
    ListElem(T data, ListElem<T> prev, ListElem<T> next) {  
        this.data = data;  
        this.prev = prev;  
        this.next = next;  
    }
```

```
}
```

```
// last element linked with first
```

```
public class TwoLinkedList<T> {
```

```
    public TwoLinkedList() {
```

```
    }
```

```
    public void pushFront(T data)  
    {
```

```
        ListElem<T> elem = null;  
        if (head != null) {  
            elem = new ListElem<T>(data, head.prev, head);  
            head.prev.next = elem;  
            head.prev = elem;  
        }  
        else {  
            elem = new ListElem<T>(data, null, null);  
            elem.prev = elem;  
            elem.next = elem;  
            head = elem;  
        }  
    }
```

```
}
```

```
    public T popEnd() {  
        ListElem<T> ret = null;
```

```

        if (head != null) {
            if (head.prev != head) {
                ret = head.prev;
                ret.prev.next = head;
                head.prev = ret.prev;
            }
            else {
                ret = head;
                head = null;
            }
        }
        if (ret != null) return ret.data;
        else return null;
    }

    public boolean empty() {
        return head == null;
    }

    @Override
    public String toString() {
        String str = new String("Queue:{ ");
        ListElem<T> elem = head;
        if (elem != null) {

            str += new String "{" + elem.data.toString() + "}";
            if (elem.next != head) str += new String(", ");
            elem = elem.next;

            while(elem != head) {
                str += new String "{" + elem.data.toString() + "

            }

            if (elem.next != head) str += new String(", ");
            elem = elem.next;
        }

        str += new String(" }");
        return str;
    }

    // TODO :
    //     public void pushEnd(T data)      {}
    //     public T popFront()              {return null;}
    //     public void clear()              {}
    //     public boolean insert(T a, T b)  {return false;} // insert a

```

```
after b
    //    public boolean erase(T data)    {return false;}
    //    public int size()              {return 0;} // recursive

    protected ListElem<T> head;

}
```