Exercise 1:

Code:

```java
public boolean contains(T el){
    if(isEmpty())
        return false;

    return contains(el, head);
}

private boolean contains(T el, SLLNode<T> p){
    if(el == p.info) //The current node is the required
        return true;
    else if (p.next == null) //Reached the end of the list
        return false;
    else
        return contains(el, p.next);
}
```

```java
public String toString(){
    if(isEmpty())
        return "[ ]";

    return toString(head, "[");
}

private String toString(SLLNode<T> p, String str){
    str = str + "  " + p.info; // append the el to the string
    if(p.next == null) //Reached the end
        return str + "  ]";
    else
        return toString(p.next, str);
}
```

Output:

```
The list is: [  Taif  Dammam  Abha  Riyadh  Jubail  ]
It is true that the list contains Dammam.
It is false that the list contains Jeddah.
```

Exercise 2:

Code:

```java
public T dequeue(){
    if (isEmpty())
        throw new UnsupportedOperationException("Queue is empty!");

    T el = queue[front]; //Store the first el
    if (rear == 0) {
        rear = front = -1;
    }
    else
        dequeue(0);

    return el;
}

private void dequeue(int idx){
    queue[idx] = queue[idx+1];//Move the next el

    if(idx+1 == rear) { //Reached the end
        rear = idx;
        return;
    }
    else
        dequeue(idx+1);
}
```

Output:

```
The queue is:  4   4   3   11   6
First dequeue element is: 4
Second dequeue element is: 4
The queue after dequeuing two elements is:  3   11   6
The element at the front is: 3
```