

```
public class ListChain<T> implements ListInterface<T> {

    private MyNode firstNode;
    private int numberOfEntries;

    private MyNode getNodeAt(int position) {
        if (position >= 1 && position <= numberOfEntries) {
            MyNode currentNode = firstNode;
            for (int i = 1; i < position; i++)
                currentNode = currentNode.getNext();
            return currentNode;
        } else return null;
    }

    public ListChain() {
        firstNode = null;
        numberOfEntries = 0;
    }

    public void add(T newEntry) {
        MyNode newNode = new MyNode<T>(newEntry);
        if (numberOfEntries == 0) firstNode = newNode;
        else {
            MyNode lastNode = getNodeAt(numberOfEntries);
            lastNode.setNext(newNode);
        }
        numberOfEntries++;
    }

    public T remove(int position) {
        T result = null;
        if (position >= 1 && position <= numberOfEntries) {
            if (position == 1) {
                result = (T) firstNode.getData();
                firstNode = firstNode.getNext();
            } else {
                MyNode nodeBefore = getNodeAt(position - 1);
                MyNode nodeToRemove = nodeBefore.getNext();
                MyNode nodeAfter = nodeToRemove.getNext();
                result = (T) nodeToRemove.getData();
                nodeBefore.setNext(nodeAfter);
            }
            numberOfEntries--;
            return result;
        } else throw new IndexOutOfBoundsException("Remove position is out of bounds");
    }
}
```

```
public T replace(int position, T newEntry) {
    if (position >= 1 && position <= numberOfEntries) {
        MyNode currentNode = getNodeAt(position);
        T dataValue = (T)currentNode.getData();
        currentNode.setData(newEntry);
        return dataValue;
    } else throw new IndexOutOfBoundsException("Replace position is out of bounds");
}

public T getEntry(int position) {
    if (position >= 1 && position <= numberOfEntries) {
        MyNode currentNode = getNodeAt(position);
        return (T)currentNode.getData();
    } else throw new IndexOutOfBoundsException("Get entry position is out of bounds");
}

public int getLength() {
    return numberOfEntries;
}

public boolean isEmpty() {
    return (numberOfEntries == 0);
}
}
```