

## CSE143X Section #9 Problems

For these problems, assume that we are using the standard `ListNode` class:

```
public class ListNode {
    public int data;          // data stored in this node
    public ListNode next;    // link to next node in the list

    public ListNode() {
        this(0, null);
    }

    public ListNode(int data) {
        this(data, null);
    }

    public ListNode(int data, ListNode next) {
        this.data = data;
        this.next = next;
    }
}
```

Define a class called `LinkedList` that has the basic functionality as the simple `ArrayIntList` class that we wrote initially. In doing so, assume that you have a single data field of type `ListNode` called `front`:

```
public class LinkedList {
    private ListNode front;

    <methods>
}
```

We wrote the appending `add` method in lecture:

```
public void add(int value) {
    if (front == null) {
        front = new ListNode(value);
    } else {
        ListNode current = front;
        while (current.next != null) {
            current = current.next;
        }
        current.next = new ListNode(value);
    }
}
```

Write each of the following methods. Assume all index values are legal.

- 1) a zero-argument constructor
- 2) `size()` that returns the current number of elements in the list
- 3) `get(index)` that returns the value at the given index
- 4) `toString()` that returns a comma-separated list in square brackets of the values in the list
- 5) `indexOf(value)` that returns the index of the first occurrence of the given value or `-1` if not found
- 6) `add(index, value)` that inserts the given value at the given index
- 7) `remove(index)` that removes the value at the given index