Lab 4: Recursion Linked List

```
Location: CourseWeb -> Labs/Recitations -> Lab 4: Recursion Linked List
Download the following files:
  1. RecursionLinkedList.java
  2. RecursionLLTester.java
```

Introduction

Recall the implementation of an ADT using a singly linked list to store a collection of data discussed in class. Most methods of the ADT that require traversing the link chain usually use a `while` loop. For this lab, you are going to implement three methods that require traversing the linked chain but you **must** use recursions instead of `while` loops.

What to do

The starter class (`RecursionLinkedList.java`) is a very simple ADT for storing integers (int) with only four public methods as follows:

- `public void add(int anItem)`
- `public boolean contains(int anItem)`
- `public int getFrequencyOf(int anItem)`
- `public String toString()`
- `public int getIndexOf(int anItem)`

The method `add()` is given. What you need to do is to implement methods `contains()`, `getFrequencyOf()`, `toString()`, `getIndexOf()` using recursions. You are allowed to create helper methods if you need to. For example, the method `contains()` may call another recursive method, etc.

The meaning of methods `contains()` and `getFrequencyOf()` are pretty much straightforward. For the method `toString()`, it should return a String in the following form:

- Begins with the open square bracket (`[`)
- Ends with the closed square bracket (`]`)
- Use comma to separate between two items (`,`)
- No spaces are allowed.

For example, suppose the class `RecursionLinkedList` contains 1,2,3,4,5 from the first node to the last node, your method `toString()` should return the string `"[1,2,3,4,5]"`. For the method `getIndexOf()`, assume that the first entry is at index 0 and return the index of the first occurrence of `anItem`.

Test Class

A test class (`RecursionLLTester.java`) is provided. Simply compile and run this test class. This program will test your `RecursionLinkedList.java` and show your total points (out of 10). If you do not get the full 10 points, you should keep trying to fix your code until you get 10 points. Note that this test class in not perfect. It cannot tell you why your program is incorrect. You may have to look at the source code of `RecursionLLTester.java` and see why it says `FAIL` and trace your code.

Due Date and Submission

For the due date, please check the lab in the CourseWeb. Submit your `RecursionLinkedList.java` to the CourseWeb under this lab by the due date. **No late submission will be accepted**.