

# CS 284: Quiz 4 – Fall 2020

October 13, 2020

Student Names:

Honor Pledge:

## Exercise 1

Write a **recursive** method `private static void incList(Node<Integer> l)` that adds one to every integer in the list that starts at node `l`. For example, the result of applying this operation to `[2,9,5,4]` should be `[3,10,6,5]`. Here is the stub:

```
public class SLL<E> {

    public static class Node<F> { ... }

    private static void incList(Node<Integer> l) {
        // TODO
    }

    public static void incList(SLL<Integer> l) {
        incList(l.head);
    }

    public static void main(String[] args) {

        SLL<Integer> l0 = new SLL<>();

        l0.add(2);
        l0.add(9);
        l0.add(5);
        l0.add(4);
        incList(l0);
        System.out.println(l0);
    }
}
```

## Exercise 2

Implement the following operation to be included in the class `SLL<E>`

```
public void mergeInto(SLL<E> l2)
```

that modifies the recipient list by merging in the one supplied as argument. For example,

- the result of merging `[2,9,5,4]` and `[21,91,51,41]` should be `[2,21,9,91,5,51,4,41]`.
- the result of merging `[2,9]` and `[21,91,51,41]` should be `[2,21,9,91]`.
- the result of merging `[2,9,5,4]` and `[21,91]` should be `[2,21,9,91,5,4]`.

```
public class SLL<E> {

    public static class Node<F> { ... }

    private Node<E> mergeInto(Node<E> l1, Node<E> l2) {
        // TODO
    }

    public void mergeInto(SLL<E> l2) {
        head = mergeInto(head, l2.head);
        size = size + Math.min(size, l2.size);
    }

    public static void main(String[] args) {

        SLL<Integer> l1 = new SLL<>();

        l1.add(2);
        l1.add(9);
        l1.add(5);
        l1.add(4);
        System.out.println(l1);

        SLL<Integer> l2 = new SLL<>();

        l2.add(21);
        l2.add(91);
        l2.add(51);
        l2.add(41);
        System.out.println(l2);

        l1.mergeLists(l2);

        System.out.println(l1);
    }
}
```