Isaac Ray Shoebottom
CS 1083
Assignment 9
3429069

Source Code:

Sparce.java:

```
import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;

public class Sparse {
    private Node head;

    Sparse(String file) {
        try {
            readList(file);
        } catch (FileNotFoundException exception) {
            System.err.println("File not found");
        }
    }

    private void readList(String filename) throws FileNotFoundException {
        File file = new File(filename);
        Scanner sc = new Scanner(file);
        while (sc.hasNextLine()){
            long index = sc.nextLong();
            long value = sc.nextLong();
            insertValue(index, value);
            if (sc.hasNextLine())
                sc.nextLine();
        }
    }

    public void mergeLists(Sparse other) {
        Node headNode = other.head;
        while (headNode != null) {
            insertValue(headNode.index, headNode.value);
            headNode = headNode.next;
        }
    }

    private void insertValue(long index, long value) {
        if (head == null) {
            head = new Node(index, value);
        }
        else {
            Node nextNode = head;
            Node lastNode = head;
            if (value == 0)
                return;
            while (nextNode != null && nextNode.index < index) {
                lastNode = nextNode;
                nextNode = nextNode.next;
            }
            if (nextNode != null && nextNode.index == index) {
                nextNode.value += value;
                if (nextNode.value == 0) {
                    lastNode.next = nextNode.next;
                }
            }
            else {
                Node insertedNode = new Node(index, value);
                insertedNode.next = lastNode.next;
                lastNode.next = insertedNode;
            }

        }
    }

    private void printRecR(Node node) {
        if (node != null) {
```

```java
            System.out.println(node.index + " : " + node.value);
            printRecR(node.next);
        }
    }

    public void printRec() {
        printRecR(head);
    }

    private void printRecBackwardsR(Node node) {
        if (node != null) {
            printRecBackwardsR(node.next);
            System.out.println(node.index + " : " + node.value);
        }
    }

    public void printRecBackwards() {
        printRecBackwardsR(head);
    }

    private static class Node {
        long index;
        long value;
        Node next;
        Node(long index, long value) {
            this.index = index;
            this.value = value;
            this.next = null;
        }
    }
}
```

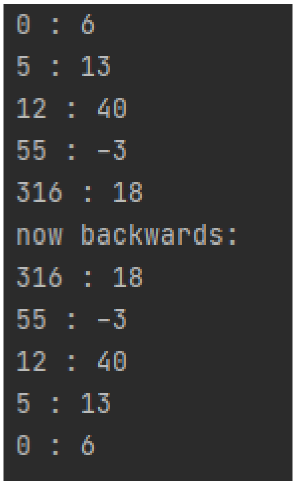
TestSparce.java:

```java
public class TestSparse {
    public static void main(String[] args) {
        Sparse sparse1 = new Sparse(args[0]);
        Sparse sparse2 = new Sparse(args[1]);

        sparse1.mergeLists(sparse2);
        sparse1.printRec();
        System.out.println("now backwards:");
        sparse1.printRecBackwards();
    }
}
```

Test 1:

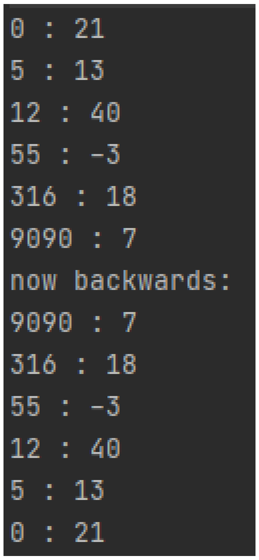
Output:



```
0 : 6
5 : 13
12 : 40
55 : -3
316 : 18
now backwards:
316 : 18
55 : -3
12 : 40
5 : 13
0 : 6
```

Test 2:

Output:



```
0 : 21
5 : 13
12 : 40
55 : -3
316 : 18
9090 : 7
now backwards:
9090 : 7
316 : 18
55 : -3
12 : 40
5 : 13
0 : 21
```