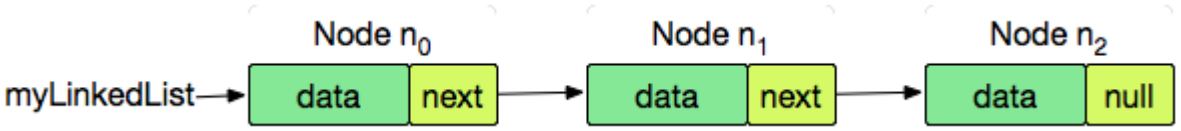
Day 15: Linked List | HackerRank

Terms you'll find helpful in completing today's challenge are outlined below, along with sample Java code (where appropriate).

A singly linked list is a data structure having a list of elements where each element has a reference pointing to the next element in the list. Its elements are generally referred to as *nodes*; each node has a *data* field containing a data value and a *next* field pointing to the next element in the list (or `null` if it is the last element in the list). The diagram below depicts a linked list of length **3**:



The sample code below demonstrates how to create a *LinkedList* of Strings, and some of the operations that can be performed on it.

```
LinkedList<String> myLinkedList = new LinkedList<String>();

// Add a node with data="First" to back of the (empty) list
myLinkedList.add("First");

// Add a node with data="Second" to the back of the list
myLinkedList.add("Second");

// Insert a node with data="Third" at front of the list
myLinkedList.addFirst("Third");

// Insert a node with data="Fourth" at back of the list
myLinkedList.addLast("Fourth");

// Insert a node with data="Fifth" at index 2
myLinkedList.add(2, "Fifth");

// Print the list: [Third, First, Fifth, Second, Fourth]
System.out.println(myLinkedList);

// Print the value at list index 2:
System.out.println(myLinkedList.get(2));

// Empty the list
myLinkedList.clear();

// Print the newly emptied list: []
System.out.println(myLinkedList);

// Adds a node with data="Sixth" to back of the (empty) list
myLinkedList.add("Sixth");
System.out.println(myLinkedList); // print the list: [Sixth]
```

The above code produces the following output:

```
[Third, First, Fifth, Second, Fourth]
Fifth
[]
[Sixth]
```

Solve Problem