SCENARIO:
Implement a LinkedArrayList class which implements List interface and extends AbstractList class.
Your class should keep a linked list where each node contains an array of elements with constant capacity. These arrays should be partially filled arrays. Arrays in the nodes of the linked list, may contain different number of elements.

When you remove an element in the list, you should find the array containing the elements, first. During remove operation, you should shift the elements (coming after the removed element) in that array only. The other arrays should not be affected. Of course, if the number of elements in an array becomes zero, the node containing this array should also be removed from the linked list.

When you add a new element, you need to add the elements into the corresponding array. If the capacity of the array is exhausted, you should create a new node (containing an empty array) in the linked list instead of incrementing the size of the array (you are not allowed to reallocate). Of course, you may want to move some of the elements in the exhausted array into the new array.

You are not allowed to use any class in Collection hierarchy as a member. So, you need to define your own Node class to build a linked list and use basic array structure in Java. Note that you need to implement all required methods and ListIterator class.
Write a Main class to test each method in your class.