LinkedList Exercise

The following Java class definitions will be used for the exercises below.

```java
public class MyLinkedList {
    private class Node {
        private Object data;
        private Node next;

        public Node(Object data) {
            this.data = data;
            next = null;
        }
    }
}
```

**IMPORTANT: You may not use the Java API LinkedList class.**

1. Implement a `MyLinkedList` constructor that defines an empty linked list.
2. Implement a method named `isEmpty` that determines whether a list is empty.
3. Implement a method named `size` that returns the number of elements in the list.
4. Implement a `MyLinkedList` constructor that has the following signature:

   ```java
   MyLinkedList(ArrayList<Object> dataArray)
   ```

   The constructor will initialize the linked list using the data elements from the array.

5. Implement a method named `duplicate` that creates a duplicate (shallow copy) of the list.
6. Would it be possible to implement a method that determines whether a list is full?
7. Implement a `delete(Object targetElement)` method that removes all instances of `targetElement` from the list.
8. Define a method named `filter` that returns an `ArrayList` with those elements of the list in a specified range. You can assume the data elements of the list implement the `Comparable` interface.
9. Add a `prev` field to the `Node` structure and implement a method named "createDoublyLinkedList" that initializes the `prev` field of a `MyLinkedList` in order to turn the list into a doubly-linked list.