Q1: Recursion.

Remove duplicates from a linked list recursively. He used the linked list class which is already implemented in java. I think you need to use remove and contains methods.

Q2: Data Structure.

He provides a class called MyDoublyLinkedList and another one called Node.

```
Public class Node {
        String value;
        Node next;
        Node previous;
        ….
}
Public class MyDoubleyLinkedList {
        Node header;
        Public void addLast(){….}
        ……..
}
```

Assuming that we do the following in the main method

```
MyDoubleyLinkedList list = new MyDoubleyLinkedList();
List.addLast(“A”);
List.addLast(“B”);
```

You need to write a toString method in the MyDoubleyLinkedList which should return the string as the following: [“A”, “B”], if the list is empty, it should return [],

Note: you should not return extra commas or spaces, so [“A”, “B”,] is not right answer.

Q3: Polymorphism.

He will provide all classes. You just need to know how to use Polymorphism. You just need to add one method in the super class, then iterate polymorphically on each object in the list.