Exercise 6. (More operation functions for dynamically linked list)

In exercise 5 you implemented some operation functions for dynamically allocated linked list. The type definition of LinkedList was

```
     public class LinkedList<T extends Comparable<T>> {
          private Node first; // no last variable
                              // in this implementation
          …
     }
```

Continue to develop further the program of exercise 5. Write the following operation functions for a character list:

```
     void delete_first();
     void delete_last();
     int  find_pos(T item);
```

The find_pos function returns the order number (starting from 0, the first item) if item is found and -1 otherwise.

Use the following main function to test your new list operations:

```
public static void main(String[] args) {
        LinkedList<Character> list = new LinkedList<Character>();
        Scanner s = new Scanner(System.in);
        int  order_no;
        char to_be_searched;

        try {
            list.delete_last();

            list.insert_to_end('?');
            list.delete_last();

            list.insert_to_end('x');
            list.insert_to_end('a');
            list.insert_to_end('b');
            list.insert_to_end('c');
            list.insert_to_end('d');
            list.insert_to_end('y');
            System.out.println("List: " + list);

            System.out.print("Enter first character to be searced ? ");
            to_be_searched = s.next().charAt(0);
            if ((order_no = list.find_pos_in_list(to_be_searched)) >= 0)
                System.out.println("The order no is " + order_no);
            else
                System.out.println("Not found");

            System.out.print("Enter second character to be searced ? ");
            to_be_searched = s.next().charAt(0);
            if ((order_no = list.find_pos_in_list(to_be_searched)) >= 0)
```

```
                System.out.println("The order no is " + order_no);
            else
                System.out.println("Not found");

            list.delete_first();
            System.out.println("List: " + list);

            list.delete_last();
            System.out.println("List: " + list);
        } catch (Exception e) {
            System.out.println("Exception " + e);
        }
    }
```