

```

;package com.gradescope.labTwo
;import com.gradescope.labTwo.tests.DoublyLinkedList

    } public class LinkedString

;private final DoublyLinkedList<Character> list

    } ()public LinkedString
;()<>this.list = new DoublyLinkedList
    {

        } public LinkedString(String str)
;()<>this.list = new DoublyLinkedList
        } if (str != null)
} for (int i = 0; i < str.length(); i++)
    ;list.addLast(str.charAt(i))
        {
            {
                {

                    } ()public int size
;()return list.size
                {

                    } public void append(String str)
                        } if (str == null)
                            ;return
                                {
} for (int i = 0; i < str.length(); i++)
    ;list.addLast(str.charAt(i))
        {
            {

                } public void append(LinkedString other)
                    } if (other == null)
                        ;return
                            {

;()DoublyLinkedList.Node<Character> curr = other.getFirstCharacterNode

        } while (curr != null && curr.getElement() != null)
            ;list.addLast(curr.getElement())
                ;()curr = curr.getNext
                    {
                        {

} ()public DoublyLinkedList.Node<Character> getFirstCharacterNode
;()return list.getHead
        {

            } ()public void reverse
;()int n = list.size
            } if (n <= 1)
                ;return

```

```

{

;()DoublyLinkedList.Node<Character> left = list.getHead
;()DoublyLinkedList.Node<Character> right = list.getTail

// نقوم بمبادلة values للعقد من الطرفين حتى منتصف القائمة
    } for (int i = 0; i < n / 2; i++)
        ;()Character tmp = left.getElement
        ;left.setElement(right.getElement())
        ;right.setElement(tmp)

        ;()left = left.getNext
        ;()right = right.getPrev
    {
        {

    } public void deleteCharAt(int index)
    { if (index < 0 || index >= list.size())
        ;return
        {

;()DoublyLinkedList.Node<Character> curr = list.getHead
    } for (int i = 0; i < index; i++)
        ;()curr = curr.getNext
        {

        ;list.remove(curr)
        {

Override@
        } ()public String toString
        ;()StringBuilder sb = new StringBuilder
;()DoublyLinkedList.Node<Character> curr = list.getHead
    } while (curr != null && curr.getElement() != null)
        ;sb.append(curr.getElement())
        ;()curr = curr.getNext
        {
        ;()return sb.toString
        {
        {

```