Problem 4. [Data Structures] In your prob4 package, you will find a class DoublyLinkedList. For this problem, you must implement the following methods

void addLast(String s)

This method inserts the input s into the linked list so that it is the last element in the list.

Example: Suppose your list has these values: ["Bob","Bill","Tom"] After executing addLast with input String "Carol", the list should contain these elements (in this order): ["Bob","Bill","Tom", "Carol"]

A toString method has been provided so you can test your code.

boolean remove(String s)

This method takes the String s and check if the linked list contains s then remove the element and return true. If the element is not in the list return false.

public boolean removeFirst()

This method removes the first node and check if the list. You have to check for careful deletion if the list is empty. Maintain the proper connection after removing the first node.

public void printReverse()

This method should print the list in Reversed order.

Example: Suppose your list has these values: [Bob Bill Tom] After executing **printReverse** on the console you have to print [Tom Bill Bob]

Requirements for Problem 4:

(1) Your code must run correctly if the list already contains one or more elements or if it contains no elements.

(2) No data may be placed in the header node.

(3) You may not introduce any new instance variables, and you may not modify the other methods in DoublyLinkedList.

(4) Any Node in your DoublyLinkedList must have correct values for the next and previous Nodes.

(5) You are allowed to add a constructor to the Node if you wish.

(6) There should be no compiler or runtime errors.