**Data Structures: Problem solving with Linked Lists**

**Instructions**

In this challenge you will work on several functions that manipulates data in a linked list.

**Existing Files**

| **File** | **Description** |
| --- | --- |
| \_\_tests\_\_/cycle.test.js | Contains tests for the cycle task. You are welcome to look at this file but do not modify it. |
| \_\_tests\_\_/editor.test.js | Contains tests for the editor task. You are welcome to look at this file but do not modify it. |
| src/cycle.js | Write your solution to the cycle problem in this file. |
| src/editor.js | Write your solutions to the editor task in this file. |
| src/lib/linkedList.js | An implementation of a linked list. Do not edit this file. |
| src/lib/node.js | An implementation of a node. Do not edit this file. |

**Tasks**

**Cycle in a list**

Write an algorithm to find whether a linked list has a cycle (i.e., whether a node in the list has its next value pointing to an earlier node in the list). Return true if a cycle is found, false otherwise.

1:1

Diagram

Description automatically generated

**Text Editor**

You are working on creating a text editor. The editor allows the user to type text, delete text and move the cursor around using the arrow keys.

The characters of the text in the editor is stored in a linked list. A pointer to the nodes of the list tracks the position of the cursor. If the cursor is null it means that the cursor is before the first character in the editor.

An *Editor* class has been started to store and manipulate the data in the editor. It contains two keys: text is a linked list storing the characters that are in the editor and cursor is a pointer to the position of the cursor in the editor.

In this challenge you will complete the *Editor* class by implementing the following methods. Each method should return a reference to the editor so that they become chainable.

**insert(char)**

Insert the given character at the cursor position. For example, given the following editor:

Diagram

Description automatically generated

If you run insert("K") the result will be:

A picture containing diagram

Description automatically generated

Remember to consider what happens if the editor is empty or if the cursor is null.

**delete()**

Remove the character currently indicated by the cursor. The cursor should end up pointing to the previous character to the one that was deleted. Remember to consider what happens if the editor is empty or if the cursor is positioned at the head of the list.

**arrowLeft()**

Move the cursor one place to the left. That is, the cursor points to the previous character. Remember to consider what happens if the cursor is already at the head of the list.

**arrowRight()**

Move cursor one place to the right. That is, the cursor points to the next character in the list. Remember to consider what happens when the cursor is already at the last character in the list.