Our Solution(s) Run Code

Solution 1

```
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

public class Program {
  public class DoublyLinkedList {
    public Node Head;
    public Node Tail;

    // O(1) time | O(1) space
    public void SetHead(Node node) {
      if (Head == null) {
        Head = node;
        Tail = node;
        return;
      }
      InsertBefore(Head, node);
    }

    // O(1) time | O(1) space
    public void SetTail(Node node) {
      if (Tail == null) {
        SetHead(node);
        return;
      }
      InsertAfter(Tail, node);
    }

    // O(1) time | O(1) space
    public void InsertBefore(Node node, Node nodeToInsert) {
      if (nodeToInsert == Head && nodeToInsert == Tail) return;
      Remove(nodeToInsert);
      nodeToInsert.Prev = node.Prev;
      nodeToInsert.Next = node;
      if (node.Prev == null) {
        Head = nodeToInsert;
      } else {
        node.Prev.Next = nodeToInsert;
      }
      node.Prev = nodeToInsert;
    }

    // O(1) time | O(1) space
    public void InsertAfter(Node node, Node nodeToInsert) {
      if (nodeToInsert == Head && nodeToInsert == Tail) return;
      Remove(nodeToInsert);
      nodeToInsert.Prev = node;
      nodeToInsert.Next = node.Next;
      if (node.Next == null) {
        Tail = nodeToInsert;
      } else {
        node.Next.Prev = nodeToInsert;
      }
      node.Next = nodeToInsert;
    }

    // O(p) time | O(1) space
    public void InsertAtPosition(int position, Node nodeToInsert) {
      if (position == 1) {
        SetHead(nodeToInsert);
        return;
```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```
// Feel free to add new properties and methods to the class.
public class Program {
  public class DoublyLinkedList {
    public Node Head;
    public Node Tail;

    public void SetHead(Node node) {
      // Write your code here.
    }

    public void SetTail(Node node) {
      // Write your code here.
    }

    public void InsertBefore(Node node, Node nodeToInsert) {
      // Write your code here.
    }

    public void InsertAfter(Node node, Node nodeToInsert) {
      // Write your code here.
    }

    public void InsertAtPosition(int position, Node nodeToInsert) {
      // Write your code here.
    }

    public void RemoveNodesWithValue(int value) {
      // Write your code here.
    }

    public void Remove(Node node) {
      // Write your code here.
    }
```

Our Tests

Custom Output Submit Code

```
}

private void expectHeadToCliProgram(DoubleLinkedList linkedList, Pro
  Program.Node head) {
  while (expectToCliLinkedList.head == head) {
  while (expectToCliLinkedList.tail == tail) {
}

private void expectStringAtNode(Program.DoubleLinkedList linkedList, I
  while (expectToCliLinkedList.head == node) {
  while (expectToCliLinkedList.tail == node) {
}

private List<int> getNodeValuesHeadToCliProgram.DoubleLinkedList
  List<int> values = new List<int>();
  Program.Node node = linkedList.head;
```

Run or submit code when you're ready.