Circular Linked Lists and Splitting Circular Linked List

```
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }
}

class CircularLinkedList {
  constructor(data) {
    this.head = null;
  }

  addLast(data) {
    const newNode = new Node(data);
    if (!this.head) {
      this.head = newNode;
      newNode.next = newNode;
      return;
    }
    let current = this.head;
    while (current.next !== this.head) {
      current = current.next;
    }
    current.next = newNode;
    newNode.next = this.head;
  }
  size() {
    let count = 0;
    if (!this.head) {
      return count;
    }
    let current = this.head;

    do {
      count++;
      current = current.next;
    } while (current !== this.head);
    return count;
  }
  print() {
    if (!this.head) {
      return;
    }
    let current = this.head;
    do {
      console.log(current.data);
      current = current.next;
    } while (current !== this.head);
  }
  checkCircular() {
    if (!this.head) {
      return;
    }
    let count = this.size();
    let current = this.head;

    do {
      count--;
      current = current.next;
    } while (current !== this.head && count > 0);

    if (current !== this.head) {
      return false;
    }
    return true;
  }
  splitTwoHalf() {
    if (!this.head) {
      return null;
    }

    let slow = this.head;
    let fast = this.head;

    while (fast.next !== this.head && fast.next.next !== this.head) {
      slow = slow.next;
      fast = fast.next.next;
    }

    let secondHalfHead = slow.next;
    slow.next = this.head;

    if (fast.next.next === this.head) {
      fast = fast.next;
    }

    fast.next = secondHalfHead;

    return { firstHalfHead: this.head, secondHalfHead: secondHalfHead };
  }
}
function printCircularLinkedList(head) {
  if (!head) {
    console.log("List is empty");
    return;
  }
  let current = head;
  do {
    console.log(current.data);
    current = current.next;
  } while (current !== head);
}
const linkedlist = new CircularLinkedList();

linkedlist.addLast(3);
linkedlist.addLast(13);
linkedlist.addLast(8);
linkedlist.addLast(5);
linkedlist.addLast(10);

console.log("Size =====>", linkedlist.size());
linkedlist.print();
console.log(linkedlist.checkCircular());
const halves = linkedlist.splitTwoHalf();
console.log("First Half:");
printCircularLinkedList(halves.firstHalfHead);
console.log("Second Half:");
printCircularLinkedList(halves.secondHalfHead);
```