```
------------------------------Link_List--------------------------------

class Node{
  constructor(data, next=null){
    this.data = data;
    this.next = next;
  }
}
-------------------
class LinkedList {
  constructor(){
    this.head = null;
  }

  insertFisrt(data){
    this.head = new Node(data, this.head)
  }

  size(){
    let node = this.head;
    let counter=0;
    while(node){
      counter++;
      node = node.next;
    }
    return counter;
  }

  getFirst(){
    return this.head;
  }

  getLast(){
        let node = this.head;
      while(node.next){
      node = node.next;
    }
    return node;
  }

  clear(){
    this.head = null;
  }

  removeFirst(){
    if(!this.head){
      return;
    }
    let removed = this.head;
    this.head = this.head.next;
    return `removed data - ${removed.data}`;
  }
```

```
removeLast(){
  if(!this.head.next){
    this.head = null;
    return;
  }
  let node = this.head;
  while(node.next.next){
    node=node.next;
  }
  node.next = node.next.next;
}

insertLast(data){
  if(!this.head){
    this.head = new Node(data);
  }
  let node = this.head;
  while(node.next){
    node = node.next;
  }
  node.next = new Node(data);
}

getAt(index){
  let node = this.head;
  let counter=0;
  while(node){

    if(counter === index){
      return node;
    }
    counter++;
    node = node.next;
  }
  return null;
}

removeAt(index){
  if(!this.head){
    return
  }
  if(index === 0){
    this.head = this.head.next;
    return
  }
  const previous = this.getAt(index - 1);
  if(!previous || !previous.next){
    return
  }
  previous.next = previous.next.next;
}
```

```
  insertAt(data,index){
    if(!this.head){
      this.head = new Node(data);
      return;
    }
    if(index === 0){
      this.head = new Node(data, this.head);
      return;
    }
    let previous = this.getAt(index-1);
    previous.next = new Node(data, previous.next);
  }

  forEach(fn){
    let counter=0;
    let node = this.head;
    while(node){
      fn(node,counter);
      counter++;
      node=node.next;
    }
  }
}

const list1 = new LinkedList();
list1.insertFisrt(4);
list1.insertFisrt(5);
list1.insertFisrt(6);
list1.insertFisrt(7);
list1.insertLast(3);

// console.log(list1)
console.log(list1.size())
// console.log(list1.getFirst())
// console.log(list1.getLast())
// console.log(list1.removeFirst())
// console.log(list1.removeLast())
// console.log(list1.insertLast(3))
// console.log(list1.getAt(2))
// console.log(list1.removeAt(2))
// console.log(list1.insertAt(101,2));
list1.forEach((node,index) => {
  node.data+=10;
});
console.log(list1.getLast())
console.log(list1.getFirst())
```