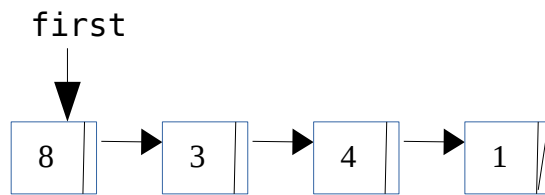


2. Obtener lista simétrica (1 punto)

Se tiene una lista enlazada simple, que contiene valores positivos distintos de cero. Por ejemplo:



Se quiere implementar el método *obtenerListaSimetrica* que, dada la lista anterior, obtenga una lista doblemente enlazada circular que contenga el doble de elementos de la lista original, de tal manera que, si la lista inicial contiene $(x_1, x_2, x_3, \dots, x_n)$, la lista resultado contendrá:
 $(-x_n, \dots, -x_3, -x_2, -x_1, x_1, x_2, x_3, \dots, x_n)$.

```
public class Node<T> {
    T data;
    Node<T> next;
}

public class DoubleNode<T> {
    T data;
    DoubleNode<T> prev;
    DoubleNode<T> next;
}

public class SimpleLinkedList { // Lista enlazada simple no circular
    Node<Integer> first;
}

public class MiDoubleLinkedList { // Lista circular doblemente enlazada
    DoubleNode<Integer> first;

    public void obtenerListaSimetrica(SimpleLinkedList l){
        //Pre: la lista "l" contiene valores positivos distintos de cero (x, y, ... z)
        //Post: el resultado es una lista del tipo (-z, ... -y, -x, x ... y, z)
        // donde cada elemento x de la lista original aparece dos veces (-x, x) de manera
        // simétrica
    }
}
```

Tomando la lista anterior como dato de entrada, el resultado de llamar a *obtenerListaSimetrica(l)* sería:

