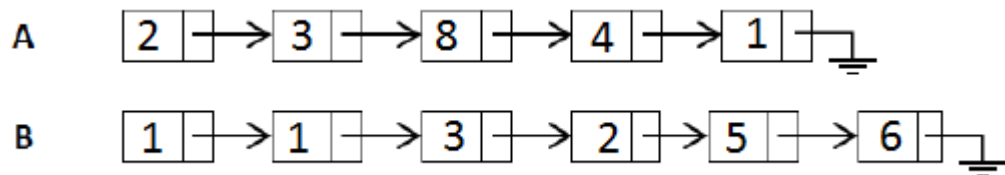


Implemente el método `sumaMenores(LinkedList <Integer> A, LinkedList <Integer> B)` que recibe dos listas A y B, y retorna una lista con la misma cantidad de elementos que A, en donde cada nodo tiene la suma de los primeros nodos de B cuyo valor es menor al valor del nodo correspondiente en A.

Por ejemplo: Sean las listas A y B



Valores de lista de retorno:

nodo 1: primeros valores menores a 2:  $1 + 1 = 2$

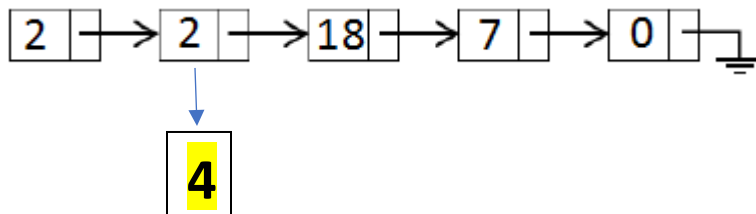
nodo 2: primeros valores menores a 3:  $1 + 1 + 2 = 4$

nodo 3: primeros valores menores a 8:  $1 + 1 + 3 + 2 + 5 + 6 = 18$

nodo 4: primeros valores menores a 4:  $1 + 1 + 3 + 2 = 7$

nodo 5: primeros valores menores a 1:  $= 0$

Lista resultado:

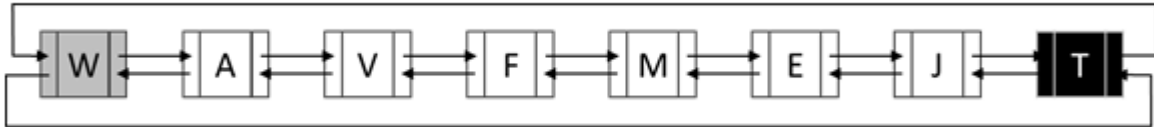


## Solución:

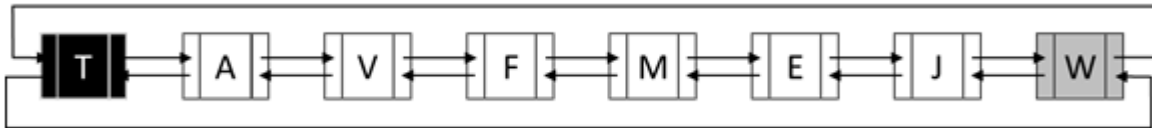
```
public static LinkedList<Integer> sumaMenores(LinkedList <Integer> A, LinkedList <Integer> B){
    LinkedList <Integer> sumaLista = new LinkedList<>();
    Iterator<Integer> iteratorA = A.iterator();
    while (iteratorA.hasNext()) {
        int acumulador=0;
        int valorA=iteratorA.next();
        Iterator<Integer> iteratorB = B.iterator();
        while (iteratorB.hasNext()) {
            int valorB=iteratorB.next();
            if(valorA>valorB){
                acumulador+=valorB;
            }
        }
        sumaLista.addLast(acumulador);
    }
    return sumaLista;
}
```

Implemente método `switchFirstAndLast` que intercambia al primer y último nodo de una lista circular doblemente enlazada. El siguiente diagrama ilustra lo solicitado:

Lista original:



Resultado después de llamar a `switchFirstAndLast`:



Al implementar su método, asuma que éste es siempre invocado por listas de tamaño mayor o igual a 3 elementos. Considere que, como se ilustra arriba, el método requiere manipular **los nodos** de la lista, no simplemente setear sus contenidos.

## Solución:

```
public void switchFirstAndLast() {  
  
    Node<E> primero= tail.getNext();  
    primero.getNext().setPrevious(tail);  
    tail.getPrevious().setNext(primeros);  
    tail.setNext(primeros.getNext());  
    primero.setPrevious(tail.getPrevious());  
    tail.setPrevious(primeros);  
    primero.setNext(tail);  
    tail=primeros;  
  
}
```