

Implementar

```
public static void copiaInvertida(Cola cola) {
```

El método tiene que invertir la secuencia y luego volver a poner la secuencia como estaba.

Que si tienes 2 6 9 luego tienes que tener 9 6 2 2 6 9 usando solo pilas y colas.

```
public static void copiaInvertida(Deque<Integer> cola) {  
    if (!Cola.isEmpty()) {
```

```
        Stack<Integer> pila1 = new Stack<>();
```

```
        Stack<Integer> pila2 = new Stack<>();
```

```
        int numElementos = cola.size();
```

```
        if (numElementos > 0) {
```

```
            for (int i = 0; i < numElementos; i++) {  
                int valor = cola.remove();  
                pila1.push(valor);  
            }
```

vaca la cola  
en la pila

9
6
2

```
            while (!pila1.isEmpty()) {  
                int valor = pila1.pop();  
                cola.add(valor);  
                pila2.push(valor);  
            }
```

vaca pila 1  
en la cola

9 6 2

y en la pila 2

2
6
9

```
            while (!pila2.isEmpty()) {  
                int valor = pila2.pop();  
                cola.add(valor);  
            }
```

vaca pila 2  
en la cola

9 6 2 2 6 9

Implementar

```
private boolean apareceMasVeces(Nodo nodo) {
```

Verificar si alguno tiene el mismo valor (dato) que el nodo recibido

Si tenemos 2<sub>0</sub> 3 4 2<sub>1</sub> apareceMasVeces(nodo 2<sub>0</sub>) retorna true  
pero apareceMasVeces(nodo 2<sub>1</sub>) retorna false

```
private boolean apareceMasVeces(Nodo nodo) {
```

```
    boolean res = false;
```

```
    int datoObjetivo = nodo.getDato();
```

```
    Nodo actual = nodo.getSiguiente();
```

```
    while (actual != null && !res) {
```

```
        if (actual.getDato() == datoObjetivo) {
```

```
            res = true;
```

```
        }
        actual = actual.getSiguiente();
```

```
    }
    return res;
```

```
}
```

```
public void siParRepetidoAñadir()
```

Cuando se encuentra un par repetido se inserta después de él dato+1

```
public void siParRepetidoAñadir() {
```

```
    Nodo actual = inicio;
```

```
    while (actual != null) {
```

```
        if (actual.getDato() % 2 == 0 && apareceMasVeces(actual)) {
```

```
            Nodo nuevo = new Nodo();
```

```
            nuevo.setDato(actual.getDato() + 1);
```

```
            nuevo.setSiguiente(actual.getSiguiente());
```

} Nodo con dato + 1

```
            actual.setSiguiente(nuevo);
```

← Inserta nuevo detrás de actual

```
            actual = nuevo.getSiguiente();
```

← Avanza al siguiente

```
        } else {
```

```
            actual = actual.getSiguiente();
```

← Avanza normal

```
    }
}
```