Question 7 (15 marks total):

Consider the generic `LinkedQueue` class and its `Node` class that we studied in the course:

```
public class LinkedQueue<E> implements Queue<E> {

    private int size;
    private Node<E> front;
    private Node<E> back;

    static class Node<E> {
        E elem;
        Node<E> next;

        Node(E elem, Node<E> node) {
            this.elem = elem;
            this.next = node;
        }
    }

    // rest of class not shown
}
```

Part A (2 marks)

Provide an implementation of the method `public void swapFrontBack()` that swaps the elements of the front and last nodes of the queue.

Part B (5 marks)

Provide an implementation of the method `LinkedQueue<E> retainFront()` that transforms this queue so that it ends up having only the front element of the queue, and the remaining elements of the queue are returned in a new `LinkedQueue`. For example, if this queue has the elements `[‘a’, ‘b’, ‘c’, ‘d’]` then after calling the method this queue has only the element `[‘a’]` and the returned queue has the elements `[‘b’, ‘c’, ‘d’]`. You may assume that this queue is not empty.

Part C (8 marks)

Provide an implementation of a copy constructor that initializes a linked queue by copying the elements from another `LinkedQueue` object.