```
// Name: Isha Tanwar
// Student number: v00958689

public class A3LinkedList implements A3List {
	private A3Node head;
	private A3Node tail;
	private int length;

	public A3LinkedList() {
		head = null;
		tail = null;
		length = 0;
	}

	public void addFront(String s) {
		A3Node n = new A3Node(s);
		if(head!=null){
			n.next = head;
			head.prev = n;
		} else {
			tail = n;
		}
		head = n;
		length++;
	}

	public void addBack(String s) {
		A3Node n = new A3Node(s);
		if(head==null){
			head = n;
		} else {
			tail.next = n;
			n.prev = tail;
		}
		tail = n;
		length++;
	}

public int size() {
		return length;
}

public boolean isEmpty() {
		return length==0;
}

public void removeFront() {
		A3Node n = head;
		if(head!=null){
			head = head.next;
			head.prev = null;
			n = null;
		}
		length--;
}

public void removeBack() {
		A3Node n = tail;
		if(tail!=null){
			tail = tail.prev;
			tail.next = null;
			n.prev = null;
		}
		length--;
}

public void addFront(A3Node n) {
		if(head!=null){
			n.next = head;
			head.prev = n;
		} else {
			tail = n;
		}
		head = n;
		length++;
}

public void rotate(int n) {


		for(int i=1; i<=n; i++){
		    A3Node cur1 = head;
		    A3Node cur2 = tail;
			cur2.next = cur1;
			cur1.prev = cur2;
			A3Node cur3 = cur2.prev;
			cur3.next = null;
			cur2.prev = null;
			tail = cur3;
			head = cur2;
		}
}

public void interleave(A3LinkedList other) {
A3Node cur1 = head;
A3Node cur2 = other.head;
A3Node cur3 = tail;
A3Node cur4 = other.tail;

A3Node store1;
A3Node store2;

while(cur1!=null){
     store1 = cur1.next;
     cur1.next = cur2.next;
     cur2.next = store1;
     cur1 = cur1.next;
     cur2 = cur2.next;
}

while(cur3!=null){
     store2 = cur3.prev;
     cur3.prev = cur4.prev;
     cur4.prev = store2;
     cur3 = cur3.prev;
     cur4 = cur4.prev;
}
}

/* Purpose: return a string representation of the list
 *          when traversed from front to back
 * Parameters: none
 * Returns: nothing
 */
public String frontToBack() {
		String result = "{";
		A3Node cur = head;
		while (cur != null) {
			result += cur.getData();
			cur = cur.next;
		}
		result += "}";
		return result;
}

/* Purpose: return a string representation of the list
 *          when traversed from back to front
 * Parameters: none
 * Returns: nothing
 */
public String backToFront() {
		String result = "{";
		A3Node cur = tail;
		while (cur != null) {
			result += cur.getData();
			cur = cur.prev;
		}
		result += "}";
		return result;
}
}
```