

Solution to CSE143X Section #19 Problems

1. One possible solution appears below.

```
public int evenSum() {
    int sum = 0;
    boolean even = true;
    ListNode current = front;
    while (current != null) {
        if (even) {
            sum += current.data;
        }
        even = !even;
        current = current.next;
    }
    return sum;
}
```

2. One possible solution appears below.

```
public void removeDuplicates() {
    ListNode current = front;
    while (current != null) {
        ListNode current2 = current;
        while (current2.next != null) {
            if (current2.next.data == current.data) {
                current2.next = current2.next.next;
            } else {
                current2 = current2.next;
            }
        }
        current = current.next;
    }
}
```

3. Two possible solutions appear below. The second is shorter because it is written recursively.

```
public void switchPairs() {
    if (front != null && front.next != null) {
        ListNode current = front.next;
        front.next = current.next;
        current.next = front;
        front = current;
        current = current.next;
        while (current.next != null && current.next.next != null) {
            ListNode temp = current.next.next;
            current.next.next = temp.next;
            temp.next = current.next;
            current.next = temp;
            current = temp.next;
        }
    }
}
```

```

public void switchPairs() {
    front = switchPairs(front);
}

private ListNode switchPairs(ListNode list) {
    if (list != null && list.next != null) {
        ListNode temp = list.next;
        list.next = temp.next;
        temp.next = list;
        list = temp;
        list.next.next = switchPairs(list.next.next);
    }
    return list;
}

```

4. One possible solution appears below.

```

public void takeSmallerFrom(LinkedList other) {
    if (front != null && other.front != null) {
        if (front.data > other.front.data) {
            ListNode temp = front;
            front = other.front;
            other.front = temp;
            temp = front.next;
            front.next = other.front.next;
            other.front.next = temp;
        }
        ListNode current1 = front;
        ListNode current2 = other.front;
        while (current1.next != null && current2.next != null) {
            if (current1.next.data > current2.next.data) {
                ListNode temp = current1.next;
                current1.next = current2.next;
                current2.next = temp;
                temp = current1.next.next;
                current1.next.next = current2.next.next;
                current2.next.next = temp;
            }
            current1 = current1.next;
            current2 = current2.next;
        }
    }
}

```