CS61B Week 2: Pointers

Draw box and pointer diagrams to represent the IntLists after each statement

```
IntList L = IntList.list(1, 2, 3, 4);
IntList M = L.tail.tail;
N = IntList.list(5, 6, 7);
N.tail.tail.tail = N;
L.tail.tail = N.tail.tail.tail.tail;
M.tail.tail = L;
```

Complete the following Java functions so that they perform as indicated in their comments

(a)
```
/** Given an integer n, return the given IntList L with every nth element removed.
 *  Don't use 'new'. You may modify the original IntList. */
public static IntList skipList(IntList L, int n) {
```

(b)
```
/** Return a new IntList that is the reverse of the given IntList L.
 *  Don't modify the original IntList. */
public static IntList reverseList(IntList L) {
```

(c)
```
/** Return an IntList that is the reverse of the given IntList L.
 *  Don't use 'new'. You may modify the original IntList. */
public static IntList destructiveReverseList(IntList L) {
```

(d)
```
/** Return an IntList that is composed of the odd elements of the given
 *  IntList L followed by the even elements of that IntList, maintaining order.
 *  Ex. if L is {3, 4, 6, 5, 7, 2}, return {3, 5, 7, 4, 6, 2}
 *  Don't use 'new'. You may modify the original IntList. */
public static IntList oddEvenList(IntList L) {
```

Sample Interview Question of the Week:
How would you figure out if an IntList contains a loop? Try to make your algorithm as efficient as possible.