tags: lists, list manipulation, data structures

Problem 3 (8 points)

Write a SchemeList method that, given a nonnegative int k, rotates the nodes in this list by k positions. That is, it should unlink the first k nodes in the list and move them to the end of the list, as shown below. Assume that, if N is the number of elements in this list, then k < N. Don't create any new ConsNodes or use any other methods other than those you define yourself.

Examples:

k	*Represented list before rotation*	*Represented list after rotation*
0	(A B C D)	(A B C D)
2	(A B C D)	(C D A B)
3	(A B C D)	(D A B C)

```
public class SchemeList {

        private ConsNode myHead; // pointer to the first node in a nonempty list
        private ConsNode myTail; // pointer to the last node in a nonempty list

        private class ConsNode {

                public Object myCar;
                public ConsNode myCdr;
        }
        public void rotate (int k) {



        }
}
```