

Northern Michigan University (Marquette Co, MI)

CS422-01-25W: Algorithms (Andrew A. Poe)
Quiz 1

Name: _____
Monday 27 January 2025 9:00 A.M. EST

Time: 15 minutes

```
public class LinkedListNode {  
    private int value;  
    private LinkedListNode next;  
}
```

Write this method as part of the `LinkedListNode` class: `public LinkedListNode split ()`
{...}

`split()`, on the portion of the list beginning with `this`, will remove half of the nodes from the list and put them into another list, and return a pointer to the first element of the new list. So, half of the nodes will be reachable from `this`, and the other half will be reachable from a new pointer that will be returned. I do not care what order the nodes are in when all the work is done. If there are an odd number of nodes, I do not care which list gets the extra one. Do not create nodes or change the data fields within existing nodes. Do this by pointer manipulation only. Do not use loops; use recursion only.

For example, this-->1-->2-->3-->4-->5:

`split()` might render this to be 1-->2-->3, and the new pointer to 4-->5; this would be one valid way to do this.

There is no need for the `LinkedList` class, so I didn't define one. There is no need to use a head variable. You may assume that `LinkedListNode` has reasonable sets and gets. You do not need a constructor for this problem.

```
public int length() {  
    if (next==null) return 1;  
    return 1+next.length();  
}  
  
public LinkedListNode beforebreak(int s) {  
    if (s==0) return this;  
    return next->beforebreak (s-1);  
}  
  
public LinkedListNode split () {  
    int l = length();  
    LinkedListNode llN = beforebreak ((l-1)/2);  
    LinkedListNode nh = llN.getNext();  
    nh.setnext(null);  
    return nh;  
}
```

Northern Michigan University (Marquette Co, MI)

CS422-01-25W: Algorithms (Andrew A. Poe)
Quiz 1

Name: _____
Monday 27 January 2025 9:00 A.M. EST

OR

```
public void split () {  
    if (next==null) return null;  
    LinkedListNode n = next;  
    next = n.split();  
    return n;  
}
```