

Given the following classes:

```
class LL {  
    private:  
        LLN *head;  
};
```

```
class LLN {  
    private:  
        string data;  
        LLN *next;  
};
```

write the method

```
int LL::CountOutOfOrder ();
```

This method returns the number of instances in the list in which a node comes alphabetically after its following node. (For example, if the list were MY-->DOG-->HAS-->FLEAS, the number would be 2, since MY comes after DOG and HAS comes after FLEAS.) You may write additional helper methods in LL or LLN if you would like. You may assume that standard constructors, destructors, accessors, and mutators already exist. Do not use loops; use recursion only.

Northern Michigan University (Marquette Co, MI)
CS222-61-21W Computer Science I (Andrew A. Poe)
Practice Quiz 3
Thursday 11 February 2021 10:00 A.M. EST
Time: 10 minutes

```
int LL::CountOutOfOrder () {  
  
    if (!head) return 0;  
    return head->CountOutOfOrder ();  
}  
  
int LLN::CountOutOfOrder () {  
  
    if (!next) return 0;  
    int ct = next->CountOutOfOrder ();  
    if (data > next->getdata()) return ct+1;  
    return ct;  
}
```