

Given the following classes:

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

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

write the method

```
void LL::RemoveFirstLast ();
```

This method removes the first and last nodes from the linked list. If the list were THIS-->QUIZ-->REALLY-->SUCKS, then after this list were run, the list would be QUIZ-->REALLY. Make sure you handle all special cases.

Do not use loops; use recursion only. You may write additional methods in LL and LLN if you wish. You may assume that standard constructors, destructors, accessors, and mutators have already been written. Make sure your code contains no memory leaks.

```
void LL::RemoveFirstLast () {

    if (!head) return;
    LLN *t = head;
    head = head->getnext();
    t->setnext (nullptr);
    delete t;
    if (!head) return;
    if (head->getnext()) head->RemoveFirstLast ();
    else {
        delete head;
        head = nullptr;
    }
}

void LLN::RemoveFirstLast () {

    if (next->getnext()) next->RemoveFirstLast ();
    else {
        delete next;
        next = nullptr;
    }
}
```