

Given the following classes:

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

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

write the method

```
void LL::DeleteLastProfessor ();
```

This method removes the node containing the string “PROFESSOR” from the linked list. If there is more than one such node, this method removes the node closest to the tail containing this word.

For example, if the list were

PROFESSOR-->EINSTEIN-->IS-->SMARTER-->THAN-->PROFESSOR-->NEWTON

after running it would be

PROFESSOR-->EINSTEIN-->IS-->SMARTER-->THAN-->NEWTON

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. Hint: your LLN method might want to return a boolean indicating whether a node was already deleted.

Northern Michigan University (Marquette Co, MI)
CS222-61-21W Computer Science I (Andrew A. Poe)
Practice Quiz 7
Thursday 18 March 2021 10:00 A.M. EDT
Time: 10 minutes

```
void LL::DeleteLastProfessor () {  
  
    if (!head) return;  
    if (!head->DeleteLastProfessor() && head->getdata()=="PROFESSOR") {  
        LLN *t = head;  
        head = t->getnext();  
        t->setnext (nullptr);  
        delete t;  
    }  
}  
  
bool LLN::DeleteLastProfessor () {  
  
    if (!next) return false;  
    bool b = next->DeleteLastProfessor ();  
    if (!b && next->getdata()=="PROFESSOR") {  
        LLN *t = next;  
        next = t->getnext();  
        t->setnext (nullptr);  
        delete t;  
        return true;  
    }  
    return b;  
}
```