

Given the following classes:

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

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

write the method

```
void LL::DeleteMatchingLength () ;
```

This method deletes the head node and all nodes whose string length matches the length of the first string.

For example, if the list were

GREAT-->FLEAS-->HAVE-->LITTLE-->FLEAS

after running it would be

HAVE-->LITTLE

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::DeleteMatchingLength () {  
  
    if (!head) return;  
    head = head->DelAll(head->getdata().length());  
}  
  
LLN * LLN::DelAll (int n) {  
  
    if (next)  
        next = next->DelAll (n);  
    if (data.length()==n) {  
        LLN *t = next;  
        next = nullptr;  
        delete this;  
        return t;  
    }  
    return this;  
}
```