

Given the following classes:

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

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

write the method

```
void LL::SkipEveryOther ();
```

This method removes every other node from the linked list. It keeps the head, skips the next, keeps the next, skips the next. If the linked list were A-->B-->C-->D-->E, after running it would be A-->C-->E.

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::SkipEveryOther () {  
  
    if (head) head->SkipEveryOther ();  
}  
  
void LLN::SkipEveryOther () {  
  
    if (!next) return;  
    LLN *t = next;  
    if (t->getnext()) t->getnext()->SkipEveryOther ();  
    next = t->getnext();  
    t->setnext(nullptr);  
    delete t;  
}
```