

Given the following classes:

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

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

write the method

```
string LL::FirstAfter (string w);
```

This method returns the alphabetical first word in the linked list of all the words that alphabetically come after the word in *w*. For example, if the linked list were I-->ATE-->A-->GINORMOUS-->AMOUNT-->OF-->PANFRIED-->SAUSAGE, and I called the method with "BACON", the method should return "GINORMOUS". If there is no qualifying word in the list, the method should return the empty string.

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.

```
string LL::FirstAfter (string w) {
```

```
    if (!head) return "";  
    return head->FirstAfter (w);
```

```
}
```

```
string LLN::FirstAfter (string w) {
```

```
    string s = "";  
    if (next) s = next->FirstAfter (w);  
    if (data <= w) return s;  
    if (s=="") return data;  
    if (data < s) return data;  
    return s;
```

```
}
```