Northern Michigan University (Marquette Co, MI)
CS222-61-21W Data Structures (Andrew A. Poe)
Quiz 9
Friday 9 April 2021 10:00 A.M. EDT
Time: 10 minutes

Given the following classes:

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

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

write the method

`void LL::DeleteFirstOfLength (int l);`

Of all strings of length l (assume l > 0) in the linked list, this method removes those nodes that come alphabetically first.
For example, if DeleteFirstOfLength(4) were called on:
I-->LIKE-->TRAFFIC-->LIGHTS-->THAT-->IS-->WHAT-->I-->SAID
after running it would be
I--> TRAFFIC-->LIGHTS-->THAT-->IS-->WHAT-->I-->SAID

with LIKE deleted, since LIKE is the alphabetical first string of length 4. If there had been more than one LIKE in the list, all would have been deleted.

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.

Northern Michigan University (Marquette Co, MI)
CS222-61-21W Data Structures (Andrew A. Poe)
Quiz 9
Friday 9 April 2021 10:00 A.M. EDT
Time: 10 minutes

Northern Michigan University (Marquette Co, MI)
CS222-61-21W Data Structures (Andrew A. Poe)
Quiz 9
Friday 9 April 2021 10:00 A.M. EDT
Time: 10 minutes

```
void LL::DeleteFirstOfLength (int l) {

 if (!head) return;
 string s = head->getfirstoflength(l);
 if (s != "") head = head->DelAll(s);
}

string LLN::getfirstoflength (int l) {

 string s = "";
 if (next) s = next->getfirstoflength (l);
 if (data.length() != l) return s;
 if (s=="") return data;
 if (data < s) return data;
 return s;
}

LLN * LLN::DelAll (string s) {

 if (next)
  next = next->DelAll (s);
 if (data==s) {
  LLN *t = next;
  next = nullptr;
  delete this;
  return t;
 }
 return this;
}
```