CS 4390 Assignment 3.14: In this assignment, you will have to work on a general purpose doubly linked list that can store derisive comments about the Vikings and perform parsing functions on each element. It utilizes the bug free doubly linked list ADT that you constructed back in assignment zero. You will have to perform your own range checking. A bug-ridden test program is included to help you exterminate any bugs you might have. Obviously, this assignment is very tough, so start early. Due July 4, 1776 at midnight. Submit the assignment in Spanish to your section leader's uncle's nephew's daughter's dog. Today's secret message: This assignment will self destruct in 10 seconds.

Probabilistic grammars can be used to generate somewhat natural sounding text, such as the one above. In this assignment, you will use your knowledge about strings and recursion to write a program that implements a probabilistic grammar. Then you will use your program to generate samples of random (but hopefully coherent-looking) text. In the appendix you'll find an example of a probabilistic grammar.

Each rule in the grammar describes how to generate a string, possibly using other rules to do so. An example is the following:

```
{  <start>
    I need an extension because <plea> . ;
}
```

The rule above can be coded by the following Java method:

```java
public static String start(){
    return "I need an extension because " + plea()+ ".";
}
```

For each term of the form `<term>` (which we call a non-terminal) in our grammar, there is a rule that describes the strings that it can generate. So, to describe the strings that can be generated from the term `<start>` we also need rules to define the strings that can be generated from `<plea>`.

When a rule terminal can generate more than one possible sequence of terms, we write each of the possible sequences followed by a ";". An example is the rule:

```
{  <athletic-event>
    an alligator wrestling meet  ;
    an intramural monster truck meet  ;
    the Winter Olympics  ;
    the four-square semi-finals  ;
}
```

This means that the non-terminal `<athletic-event>` can generate four possible sequences of terms: “an alligator wrestling meet”, “an intramural monster truck meet”, “the Winter Olympics”, and “the four-square semi-finals”. We assume that all sequences in a rule have the same probability. In this case, since we have four alternatives, each has a probability of one fourth. Therefore, we can code the rule as follows:

```java
public static String athletic-event(){
    Random m= new Random();
```
int choice=m.nextInt(4);
switch(choice) {
    case 0: return " an alligator wrestling meet ";
    case 1: return " an intramural monster truck meet ";
    case 2: return " the Winter Olympics ";
    case 3: return " the four-square semi-finals ";
    default:return "[ERROR] on athletic-event, invalid option";
} // end switch

To implement a grammar, you need to write a method of type String for every non-terminal in the grammar of your choice. Every sentence will be generated by calling the start method, which in turn will call the other methods as appropriate. Your main program will contain a loop that will ask the user if she/he wants to continue generating texts or quit the program.

Your assignment consists of implementing one of the grammars described in http://www-cs-faculty.stanford.edu/~zelenski/rsq/grammars/ (which includes the one used to generate the first paragraph in this assignment) and also to design and implement your own (hopefully funny) grammar about a topic of your choice. Make sure you are not implementing the same grammar as any other team in the class.
Appendix

The complete grammar from which the rules described above were taken is the following:

{  
  <start>  
  I need an extension because <plea> . ;  
}  

{  
  <plea>  
  <dubious-excuse> ;  
  <dubious-excuse> ;  
  <dubious-excuse> ;  
  <dubious-excuse> ;  
  <dubious-excuse> ;  
  <dubious-excuse> , and then <plea> ;  
  <dubious-excuse> , and on top of that <plea> ;  
  <dubious-excuse> , and as if that wasn't enough <plea> ;  
  <dubious-excuse> , and then get this, <plea> ;  
  <dubious-excuse> , and just then <plea> ;  
  <dubious-excuse> , and, well I am a little embarassed about this, but <plea> ;  
  <dubious-excuse> , and I'm sure you've heard this before, but <plea> ;  
  <dubious-excuse> , and then, just when my mojo was getting back on its feet, <plea> ;  
  <dubious-excuse> , and just then <plea> ;  
  <dubious-excuse> , and right about then <plea> ;  
  <dubious-excuse> , and then if I recall correctly <plea> ;  
  <dubious-excuse> , oh, and then <plea> ;  
}  

{  
  <dubious-excuse>  
  my disk got erased ;  
  my dog ate my <something> ;  
  my roommate ate my <something> ;  
  I didn't know I was in this class ;  
  I thought I already graduated ;  
  my dorm burned down ;  
  I spent all weekend hung-over ;  
  I had <lots-of-work> ;  
  I had <lots-of-work> ;  
  well, it's all a haze ;  
  I had to go to <athletic-event> ;  
  I had to practice for <athletic-event> ;  
  I had to worry about <athletic-event> ;  
  I lost a lot of money on <athletic-event> ;  
  I forgot how to write ;  
  all my pencils broke ;  
  the bookstore was out of erasers ;  
  I used up all my paper ;  
  I had to go to this really worthwhile event ;  
  I had to make up a lot of documentation for the Navy in a big hurry ;  
  I got stuck in a blizzard at Tahoe ;  
  I skied into a tree ;  
  my karma wasn't good ;  
  I just didn't feel like working ;  
  it was just too nice outside ;  
  the programming language was inadequately abstract ;  
  I had to do laundry ;  
  I lost my <something> ;  
}
my <something> had a problem of a private nature 
my <something> was confiscated by res-ed 
my <something> was set alight and thrown off the roof 
my <something> was enshrowded in a mysterious fog for three days and then vanished 
I had recurring dreams about my <something>

{ lots-of-work
<impressive-number> midterms 
<impressive-number> midterms and <impressive-number> papers 
to finish my doctoral thesis 
<impressive-number> programs in <impressive-number> different languages
}

{ athletic-event
an alligator wrestling meet 
an intramural monster truck meet 
the Winter Olympics 
the four-square semi-finals
}

{ <impressive-number
4 
7 
like, a billion; 
tons of; 
 mega 
like, a bunch,
}

{ <something
disk 
backpack 
 mind 
sense of purpose 
book 
notes 
Mac 
computer 
module specification 
printout 
dreams 
motivation
}