CS272 Lab Assignment #4.

Due: Thursday, March 1, 11:30pm.

For this assignment you will need to use file CharNode.java. CharNode provides a node for a linked list with char data in each node. (CharNode is like IntNode, only elements are of type char instead of int.)

DNA and Protein Synthesis

DNA (deoxyribonucleic acid) is in the family of molecules referred to as *nucleic acids*. One strand of DNA has a backbone consisting of a polymer of the simple sugar deoxyribose bonded to something called a phosphate unit. Very unimpressively then, the backbone of a strand of DNA resembles this:

sugar-phosphate-sugar-phosphate-sugar-phosphate-….

What is impressive about DNA is that each sugar molecule in the strand also binds to one of four different nucleotide bases. These bases: Adenine (**A**), Guanine (**G**), Cytosine (**C**) and Thymine (**T**), are the beginnings of what we will soon see is a molecular alphabet. Each sugar molecule in the DNA strand will bind to one nucleotide base. Thus, as our description of DNA unfolds, we see that a single strand of the molecule looks more like this:

```
C               T               G               A               …
sugar-phosphate-sugar-phosphate-sugar-phosphate-sugar-phosphate-…
```

Each strand of DNA contains millions or even billions (in the case of human DNA) of nucleotide bases. These bases are arranged in a specific order according to our genetic ancestry. The order of these base units makes up the code for specific characteristics in the body, such as eye color or nose-hair length. Just as we use 26 letters in various sequences to code for the words you are now reading, our body's DNA uses 4 letters (the 4 nucleotide bases) to code for millions of different characteristics.

Task 0: Read Chapter 4 from the textbook. Create a new project in Eclipse. Import file CharNode.java to the project. Add a new class named DNA.java with main method to the project. Task 1 asks you to modify CharNode class. The rest of the tasks deal with DNA.java class.

Task 1: *Implememt toString() method in CharNode class to convert a linked list of characters into a String. You will use this method to output contents of your linked lists. The code for CharNode class is provided: CharNode.java*

You may read about toString() method here.

Task 2: *Use a linked list of characters to represent DNA. Use* CharNode.java *– implementation of a node for a linked list with char data in each node. Write a method that takes a String and converts it into a linked list of characters. The header of the method is the following:*

    public static CharNode stringToList(String userDNA)

If the string is empty then the method should return null.
Your program should prompt the user to enter a DNA strand and use this method to convert the string user entered into a linked list of characters.

Each molecule of DNA is actually made up of 2 strands of DNA cross-linked together. Each nucleotide base in the DNA strand will cross-link (via hydrogen bonds) with a nucleotide base in a second strand of DNA forming a structure that resembles a ladder. These bases cross-link in a very specific order: A will only link with T (and vice-versa), and C will only link with G (and vice-versa). Thus our picture of DNA now looks like this:

```
sugar-phosphate-sugar-phosphate-sugar-phosphate-sugar-phosphate-…
G              A              C              T              ….
|              |              |              |
C              T              G              A
sugar-phosphate-sugar-phosphate-sugar-phosphate-sugar-phosphate-…
```

Within this coil of DNA lies all the information needed to produce everything in the human body. A strand of DNA may be millions, or billions, of base-pairs long. Different segments of the DNA molecule code for different characteristics in the body. A **Gene** is a relatively small segment of DNA that codes for the synthesis of a specific protein. This protein then will play a structural or functional role in the body.
How does a gene code for a protein? Protein synthesis is a 2 part process that involves a second type of nucleic acid along with DNA. This second type of nucleic acid is RNA, ribonucleic acid. RNA differs from DNA in two respects. First, the sugar units in RNA are ribose as compared to DNA's deoxyribose. Because of this difference, RNA does not bind to the nucleotide base Thymine, instead, RNA contains the nucleotide base Uracil (U) in place of T (RNA also contains the other three bases: A, C and G).

Transcription: In the first step of protein synthesis, the 2 DNA strands in a gene that codes for a protein unzip from each other. Similar to the way DNA replicates itself, a single strand of messenger RNA (mRNA) is then made by pairing up mRNA bases with the exposed DNA nucleotide bases. For example:

DNA Sequence:	C	T	G	A	A	T
mRNA Sequence:	G	A	C	U	U	A

In other words, mRNA sequence can be produced from DNA sequence by replacing bases in the following manner. Base A is replaced with base U, base G is replaced with base C, base C is replaced with base G, base T is replaced with base A.

Task 3: *Write a method that takes a DNA strand (a linked list of characters) and produces the corresponding mRNA sequence (a linked list of characters). The method should have one* *parameter (CharNode) – the head of a DNA strand, and return the head of the corresponding mRNA sequence (CharNode). The header of the method is the following:*

public static CharNode dnaToRNA(CharNode dnaList)

If the value of the parameter is null then the method should return null.
Your program should call this method and output the corresponding mRNA sequence.

Sequence similarity

Determining the similarity between two sequences is a common task in computational biology. Identifying similarities is important, as it is assumed that similar structure leads to similar biological functions.

A subsequence of a sequence is a list of characters from the sequence in the same order that they appear in that sequence. For example, sequence TTC is a subsequence of GATGTAACCTA since characters T, T, and C are the 3rd, the 5th, and the 8th characters in GA**T**G**T**AA**C**CTA. On the other hand, ACCG is not a subsequence of GATGTAACCTA since characters A, C, C, G appear in a different order in GATGTAACCTA (there is no G after C in the longer sequence).

When a sequence X is a subsequence of another sequence Y, then it can be a subsequence of different segments in Y. For example, let X be CAT and let Y be ACAGTGCGTATAT. Then, X appears as a subsequence of Y in many ways, including the following:

- A**CA**G**T**GCGTATAT – a subsequence of the segment **CA**G**T** of length 4 (2nd through 5th characters of Y),
- ACAGTG**C**GT**A**TA**T** – a subsequence of the segment **C**GT**A**TA**T** of length 7 (7th through 13th characters of Y),
- ACAGTG**C**GT**AT**AT – a subsequence of the segment **C**GT**AT** of length 5 (7th through 11th characters of Y),
- A**C**AGTGCGT**AT**AT – a subsequence of the segment **C**AGTGCGT**AT** of length 10 (2nd through 11th characters of Y), and so on.

Task 4: *Write a method called subSequence that given two linked lists of characters determines whether the shorter list (let us call it S) is a subsequence of the longer list (let us call it L).*
If the shorter list is a subsequence of the longer list, then the method should return the length of the shortest segment in L that contains S as a subsequence. For example, for sequences CAT and ACAGTGCGTATAT from the above example, the method should return 4.
If the shorter list is not a subsequence of the longer list, then the method should return value -1.
The method should have two parameters (of type CharNode) – heads of the two linked lists.
It is not known in advance which of the parameters (the first one or the second one) contains a shorter list. The length of the lists must be checked first to determine which of the two given lists is shorter.
(Note: The goal of this task is to practice working with linked lists. Converting lists into strings and using String methods is not permitted.)
In your program you should prompt the user to enter two strings, convert the strings into linked lists using the method which you wrote before (to convert a String into a linked list of characters),

call the method to determine whether a shorter list is a subsequence of a longer list and output the result.

To summarize: You need to implement *toString()* method in *CharNode* class (task 1). You need to implement methods *stringToList* (task 2), *dnaToRNA* (task 3), and the *subSequence* method (task 4).
In your program you should prompt the user to enter a DNA strand, convert it to a linked list using *stringToList* method, find the corresponding mRNA using *dnaToRNA* method and output it, prompt the user to enter two strings, convert them into linked lists, call the method *subSequence* to determine whether a shorter list is a subsequence of a longer list and output the result.
An output produced by your program may look like the following:

```
Please enter a DNA strand: CTAGACTACGGTATAGTTATCA
You entered: CTAGACTACGGTATAGTTATCA


Your DNA list is CTAGACTACGGTATAGTTATCA
Corresponding mRNA is GAUCUGAUGCCAUAUCAAUAGU

Please enter sequence 1: CAACCTGGCT
You entered: CAACCTGGCT
Please enter sequence 2: CCGG
You entered: CCGG

CCGG is a subsequence of a segment of length 5 of CAACCTGGCT
```

Note: Specifications for all the methods you write should be included as comments in your code. Also, please use inline comments, meaningful variable names, indentation, formatting and whitespace throughout your program to improve its readability.

What to submit:

1. Submit your source code (*.java files) electronically using Canvas.