

Compsci 004g, Fall 2005, Code Practice

These question are based on the code in the class [ProteinTools.java](#). You'll want to study the methods in this class and call them in writing the code that answers these questions.

1. An amino acid can be coded by several different codons. For example, the amino acid *Leucine* is coded by "TTA", "TTG", "CTT", "CTC", "CTA", and "CTG" whereas *Tryptophan* is only coded by "TGG".

The code fragment below can be completed by adding one line so that the call `codingCount("L")` prints

```
# codons coding for L = 6
```

since there are six codons that code for leucine and the call `codingCount("W")` should print

```
# codons coding for W = 1
```

since there is only one codon that codes for "W" which is tryptophan. The abbreviation is passed as a parameter to the method `codingCount` that you're completing.

```
public void codingCount(String abb){
    String[] list = ProteinTools.aminoAcidToCodons(abb);

    System.out.println("# codons coding for " + abb + " = " + count);

}
```

2. The method `getProtein` should return a string representing the protein coded for by the parameter DNA. For this method we're not looking for stop codons or start codons; for example, they may have been found in code written somewhere else and used before the method `getProtein` is called.

The call `getProtein("CATCCCTTTAAT")` should return the string "HPFN" and the class

`getProtein("GAAGGGCTGAC")` should return "EGAD".

The method has been started, you need to add appropriate code in three places: one a call of substring, one a call to a method in the class `ProteinTools`, and one to return the value that has been built up by concatenating protein abbreviations together.

```
public String getProtein(String dna){
    String pString = ""; // the protein string to return
    // loop over every codon, starting from first one
    for(int k=0; k < dna.length() - 2; k = k + 3) {
        String codon =                ; // use substring
        String protein = ProteinTools. ; // call
        pString = pString + protein;
    }
    return                            ;
}
```

3. There are many DNA strands that can code for a protein chain, e.g., the strands "CATCCCTTTAAT", "CACCCCTTTAAT", and "CATCCCTTTAAC" (among others) all code for the protein "HPFN". The method `possibleDNACount` returns the number of different DNA strands that could code for the protein passed as a parameter. Complete the method by adding code in the body of the for-loop so that the method will work as intended. Note that the number of strands coding for "HRT" is $2*6*4 = 48$ since there are two codons that code for H/histidine, six codons coding for R/arginine, and four for T/threonine. Thus the call `possibleDNACount("HRT")` should return/evaluate to 48.

```
public int possibleDNACount(String protein) {
    String[] firstCodons = ProteinTools.aminoAcidToCodons(protein.substring(0,1));
    int count = firstCodons.length; // # ways to code for first protein in chain
    for(int k=1; k < protein.length(); k++){
```

```

    }

    return count;

}

```

4. Complete the method `aminoAcidIndex` which returns the index in parameter `dna` at which the first occurrence of a codon coding the `aminoAcid` whose one-letter code is specified occurs. If there is no occurrence of the `aminoAcid` return -1. (*We're ignoring ORFs in this problem.*)

For example

method call	result returned	reason
<code>proteinIndex("CCCCAATTTCCAGT", "Q")</code>	3	"CAA" occurs at index 3, "Q" encodes Glutamine
<code>proteinIndex("CCCTAATTTCCAGT", "Q")</code>	10	"CAG" occurs at index 10, "Q" encodes Glutamine
<code>proteinIndex("GGTGGCGGAGGG", "G")</code>	0	"GGT" occurs at index 0, "G" encodes Glycine
<code>proteinIndex("CTTAGTAAGTTAGC", "R")</code>	-1	No Arginine, "R", coded by AGA and AGG

Complete the method below by adding code in the four places shown.

```

/**
 * @param dna is a sequence of upper-case 'A', 'G', 'T', 'C'
 * @param protein is a single-character string with valid protein code
 * @return index of first occurrence of a codon for protein or -1 if none
 */
public int proteinIndex(String dna, String aminoAcid)
{
    String[] codons = // complete

    int first = Integer.MAX_VALUE; // first is index of first codon for protein
    for(int k=0; k < codons.length; k++){
        String c = codons[k]; // codon for this amino acid
        int index = // complete, use indexOf(..)

        // add code, what if index == -1?

        first = Math.min(first, index);
    }

    // complete, add code here to return first, or -1 if never found

```

}

[Owen L. Astrachan](#)

Last modified: Tue Sep 27 14:16:17 EDT 2005