Problems related to BioLingua and Loops

Note:

- *I've added links in the Intro to Programming web page to help you get to BioLingua and help pages.*
- *There are some in your midst who already have a good handle on BioLingua. Make use of them! Aaron, Andrew, and Maitrik have past experience with the language, and Cailin has experience with a similar language. Kyle may as well.*

1. Write a loop!

   1a. Use the `DEFINE` function to define the variable ***n*** as 47.

   1b. Use the `DEFINE` function to define the variable ***n-squared*** as ***n*** multiplied by itself.[*]

   1c. Use the `DISPLAY-LINE` function to display on the screen ***n*** followed by ***n*** squared.

   Hint: `(HELP Display-line)` will give you the syntax for the function and also let you know how to separate ***n*** from ***n*** squared.

   1d. Take the lines you wrote in **1b** and **1c** and wrap them within the following loop:

```
(FOR-EACH n FROM 1 TO 10
      [1b and 1c go here]
  )
```

   Run the loop. Note that you did not copy the line you wrote in **1a** because the loop iteration control takes care of that.

   1e. Change the loop so that it no longer uses `DEFINE`, replacing that function with `AS`. Peak ahead to the next problem to see an example of `AS` in action.

   1f. Using your knowledge of BioLingua Syntax,[†] explain why `DEFINE` is enclosed in parentheses but `AS` is not.

2. You want to simulate a board game that uses pairs of dice. Accordingly, you write a loop that will generate pairs of digits from 1 to 6. Here's a test of that code, intended to simulate 10 rolls of the dice:

```
(FOR-EACH roll FROM 1 TO 10
     WITH die1 = (RANDOM-INTEGER FROM 1 TO 6)
     AS die2 = (RANDOM-INTEGER FROM 1 TO 6)
     DO (DISPLAY-LINE die1 *tab* die2))
```

   (you used `WITH` for one die and `AS` for the other because you're not sure which one to use… so compromise!). Copy the code into the Program Window[‡] of BioLingua and run it. How do you explain the results you get? What is the difference between `WITH` and `AS`?

---

[*] If you can't get arithmetic to work right, then go to the BioLingua Help page, and click on *Description of Functions*, and then *Arithmetic Functions*.

[†] Which might be increased by looking at the notes on the subject posted at the web site

[‡] If you don't know what I mean by "Program Window", then go to the BioLingua Help page, and click on *The BioLingua Listener*.

3. You have identified a protein in *Anabaena* PCC 7120 that is not found in the closely related *Anabaena variabilis* or *Nostoc punctiforme*. You wonder whether it is a part of a transposon (a piece of DNA that is capable of moving itself from one place to another). If it is, you'd like to know what this transposon is about, that is the extent of the DNA that comprises the transposon. Here is the beginning of the sequence of the protein:

   MLVFETKLEGTNEQYQLLDEAIKTARFVRNACLRYWMDNQNIGRYD

   3a. Look for the gene that encodes this protein by copying the sequence above and going to the National Center for Biotechnology Information (NCBI). You can get there through the following link:

   http://www.ncbi.nlm.nih.gov/

   Click on BLAST (top bar on page)
   Click on Protein-protein BLAST (blastp), under **Protein**
   Paste in the sequence into the search window
   Press BLAST!
   Press Format! (and wait probably many tens of seconds)
   Interpret the results

   3b. Alternatively, go to BioLingua and enter in the program window:

```
(PROTEINS-SIMILAR-TO
   "MLVFETKLEGTNEQYQLLDEAIKTARFVRNACLRYWMDNQNIGRYD"
   IN A7120)
```

   (`A7120` is a nickname for *Anabaena* PCC 7120) and press EVAL.

   If you look carefully (and don't panic), you will discern that both strategies tell you which five proteins of *Anabaena* PCC 7120 match the 46 amino acids you submitted. You might like one format over the other, depending on what you're looking for. The fact that there are five matching proteins lends support to your idea that they are transposases (and the annotation of the proteins shown in the NCBI output is even more gratifying).

   But never mind… the goal was not merely to identify the proteins but to identify the ***transposon***, the unit of DNA containing the transposase gene that moves. How to proceed?

   Here's an idea: Take the DNA sequence of the gene encoding the transposase plus a few hundred nucleotides on either side of it. Then compare that DNA segment to the entire genome of *Anabaena*. That portion of the DNA segment that is found in several places in the genome must be the transposable unit. Good plan. How to execute it?

   In NCBI, you're stuck. Can't go any further (without an enormous amount of work). So let's try BioLingua.

   3c. Extract the DNA surrounding the gene encoding the first protein on the list (p-Alr4104) and assign it to a variable, as follows:

```
(DEFINE tn-region AS (SEQUENCE-OF alr4104 FROM -200 TO-END +200))
```

3d. Find all instances of this DNA segement within the genome of A7120:

```
(SEQUENCES-SIMILAR-TO tn-region IN A7120)
```

Note that you've found five instances where the region surrounding alr4104 are matched by similar sequences. Highly suspicious… five exact matches of the protein sequence you started with and five exact matches of the region surrounding alr4104, a gene encoding one of the five proteins. Do the other four DNA regions correspond to the other four proteins? Let's figure out how to answer this question in one case and then use a loop to generalize to all five cases.

3e. Assign to a variable called ***hit-info*** the first line of the results of the protein search you did in **3b**. You can do this in the following way:

```
(ASSIGN hit-info = (FIRST (RESULT n)))
```

(where ***n*** is the number in the History Window[§] preceding the command you issued in **3b**). You'll get a link to an informational frame. Click on the link and you'll see all the information on the first line of the protein search (and more). All you want right now, however, is the name of the protein (p-all4104), which is in the row labeled **subject**.

3f. Assign to a variable called ***protein*** the name of the protein:

```
(ASSIGN protein = (GET-ELEMENT subject FROM hit-info))
```

(you'll get the official, full name of the protein, #$A7120.p-all4104, but don't worry about that).

3g. Assign to a variable called ***gene*** the gene that encodes the protein:

```
(ASSIGN gene = (GENE-OF protein))
```

You'll get a link to an informational frame. Click on the link and note the information contained about the gene. In particular, note that it contains the extent of the gene, i.e. the earliest coordinate (in the **From** field) and the latest coordinate (in the **To** field).

3h. Get the coordinates defining the extent of the gene:

```
(ASSIGN (from to) = (GET-ELEMENTS (from to) FROM gene))
```

Note that you're allowed to assign two variables at the same time and to get two values from a frame. Only the value of the first variable is displayed on the screen (a defect currently in the system), but both were assigned. Test this by entering the name of each variable as atoms to learn their values.

3i. Display the coordinates along with the gene:

```
(DISPLAY-LINE ...
```

Now compare the coordinates you found in this way from the protein search to the coordinates found by DNA sequence search (in **3d**). They should be close (why not identical?) If you can do it for the first protein, you can do it for all five. Collect the statements from **3f** to **3i** and wrap them within a loop:

[§] If you don't know what I mean by "History Window", then go to the BioLingua Help page, and click on *The BioLingua Listener*.

```
(FOR-EACH hit-info IN (RESULT n)
    [3f through 3i go here]
  )
```

3j. Do the coordinates of the genes that encode the protein correspond to the coordinates of the segments found in the DNA search of **3d**?

3k. Are the DNA segments in 3d all of the same length, as you'd expect if they were multiple instances of the same transposon? Yes you could do the subtraction to find out, but hey, we have a computer here, modify the loop you just wrote to answer this question.