

CIS 479/579 Project 1
Summer 2019

Your first programming assignment will give you an opportunity to explore symbolic computation as you write a series of functions to explore some properties of single- and double-stranded DNA. A strand of DNA can be modeled using a linked list and is composed of four bases (adenine, thymine, guanine, and cytosine).

The list (A G G T C A T T G) corresponds to a strand that is nine bases long, using the first letter of each name as a symbol for the base. Each of the four bases has a complement with which it can form a pair. Adenine pairs with thymine, while guanine pairs with cytosine.

Two complementary single-strands of DNA can combine to form a double-stranded DNA. The strands (A G G T C A T T G) and (T C C A G T A A C) are complementary.

You are to write and test the following LISP functions:

1. COMPLEMENT-BASE takes a base as input and returns the matching base. (COMPLEMENT-BASE 'A) should return T and so forth for each base.

2. COMPLEMENT-STRAND returns the complementary strand of a sequence of single-stranded DNA.

(COMPLEMENT-STRAND '(A G G T)) should return (T C C A).

3. MAKE-DOUBLE takes a single strand DNA and returns a double-stranded version.

(MAKE-DOUBLE '(G G A C T)) should return
((G C) (G C) (A T) (C G) (T A)).

4. COUNT-BASES counts the number of bases of each type in either single- or double-stranded DNA and returns the result as a table.

(COUNT-BASES '((G C) (A T) (T A) (C G))) should return
((A 2) (T 2) (G 2) (C 2))

(COUNT-BASES '(A G T A C T C T)) should return
((A 2) (T 3) (G 1) (C 2)).

5. PREFIXP returns T if one strand of DNA is a prefix of another and NIL otherwise.

For example, (G T C) is a prefix of (G T C A T) but not of (A G G T C).

6. APPEARS returns T if one DNA strand appears anywhere within another. For example, (C A T) appears in (T C A T G) but not in (T C C G T A).

Hint: If X appears in Y then X is a prefix of Y or (CDR Y) or (CDR (CDR Y)) or ...

7. COVERP returns T if its first input, repeated some number of times, matches all of its second input. For example, (A G C) covers (A G C A G C A G C) but not (A G C T T G). You may assume neither input will be NIL.

8. PREFIX returns the leftmost N bases of a DNA strand.

(PREFIX 4 '(C G A T T A G)) should return (C G A T).

9. KERNEL returns the shortest prefix of a DNA strand that can be repeated to cover the strand.

(KERNEL '(A G C A G C A G C)) should return (A G C)

(KERNEL '(A A A A A)) should return (A)

(KERNEL '(A G G T C)) should return (A G G T C)

None of these functions is particularly long, if you keep in mind that you are working with Common LISP (not Java or C++). You should feel free to write any auxiliary functions that would make your work easier to do. In fact, it may be the case that, using one of the functions defined above will make your work go much more quickly (and help improve the structure of your code as well).

You should use XLISP, LispWorks, or Common Lisp for this assignment. You must create additional test cases to demonstrate the correctness of your work.

You will need to turn in a well commented source listing containing your function definitions, carefully organized test runs (perhaps using the Common Lisp DRIBBLE function and adding some annotations to the file using your favorite word processing program to make your output easier to follow), and a two-page memo discussing your solution quality (strengths, weaknesses, general approach, etc.)

Assigned: 5/08/19

Date due: 5/15/19