INDENTATION AND SPACING GUIDELINES FOR LISP ASSIGNMENTS

Submitted files are expected to comply with the following rules when the files are viewed
on euclid using the less viewer. If you use tabs, be sure to follow rule 10 below!

1. When a list is written on more than one line, the first character of the 2nd and any
   subsequent lines in that list should be FURTHER TO THE RIGHT than the list's opening
   parenthesis. Here are two examples of BADLY written lists that VIOLATE this rule:
           (defun f (x)   and   '(the big black cat slept on the dark
           (* x 2))   red mat)
   CORRECT ways to write the above are:
           (defun f (x)   and   '(the big black cat slept on the dark
             (* x 2))   red mat)

2. Do NOT put a space immediately after an opening parenthesis or immediately before a
   closing parenthesis.
        BAD: (this is bad)   GOOD: (this is right)

3. When two consecutive elements of a list are on the same line, they should be
   separated by a space:
        BAD: ((this is)(a bad)example)   GOOD: ((this is) (a good) example)

4. Do NOT put closing parentheses on their own lines.
   For example, you should NOT write: (defun f (x y z)
                                                (+ x (* y z))
)
   You should write the following instead: (defun f (x y z)
                                                (+ x (* y z)))
   REASON: Putting closing parentheses on separate lines wastes screen space for no good
   reason and increases the amount of scrolling that is needed to read your code!

5. There should be at least one BLANK LINE after each function definition.

6. For each function call that is not written on a single line, make all the arguments of
   the call line up, as in:
        (myfunction argument1
                    argument2
                    argument3)
   or
        (myfunction
          argument1
          argument2
          argument3)
   In the second case the arguments should line up with one of the first few letters of
   the function name. If argument1, argument2, and argument3 are (func x y), (+ 2 w),
   and (anotherfunc w (+ y 1)), then you might write the call as follows:
        (myfunction
          (func x y)
          (+ 2 w)
          (anotherfunc w (+ y 1)))
   Another way to write the call would be:
        (myfunction (func x y)
                    (+ 2 w)
                    (anotherfunc w
                          (+ y 1)))

7. For each IF form that is not written on a single line, make the opening parentheses
   of the three arguments line up, as in:
```
       (if (evenp n)
           (func
             w
            (+ y 1))
           (- n 3))
```
   It is also OK to put the first two arguments on the same line as the symbol IF and
   put the third argument on the next line. In that case the third argument should line
   up with the first argument, as in the following example from p. 115 of Touretzky:
```
       (if (symbolp x) (list 'yes x 'is 'a 'symbol)
           (list 'no x 'is 'not 'a 'symbol))
```

8. For each COND form, make the opening parentheses of the clauses line up, as in:
```
       (cond ((eql e 0) '(integer-zero 0))
             ((numberp e) (list 'num-but-not-0 e))
             ((consp e) (list 'nonempty-list e))
             (t (list 'non-numeric-atom e)))
```
   This COND form could also be formatted as follows:
```
       (cond ((eql e 0)
               '(integer-zero 0))
             ((numberp e)
              (list 'num-but-not-0 e))
             ((consp e)
              (list 'nonempty-list e))
             (t (list 'non-numeric-atom e)))
```
   or
```
       (cond
         ((eql e 0)
           '(integer-zero 0))
         ((numberp e)
           (list 'num-but-not-0 e))
         ((consp e)
           (list 'nonempty-list e))
         (t (list 'non-numeric-atom e)))
```

9. For each LET or LET* form, put the first local variable initialization form on
   the same line as the symbol LET or LET*. If more than one local variable is
   initialized, make all the initialization forms line up. Indent the body of the
   LET or LET* form by two spaces. Here is an example (from p. 144 of Touretzky):
```
       (let* ((diff (- new old))
              (proportion (/ diff old))
              (percentage (* proportion 100.0)))
         (list 'widgets 'changed 'by percentage 'percent))
```

10. If you use tabs in your .lsp file, add a comment at the beginning of the file that
    tells me what -x (tab width) setting to use with the less viewer to view the
    file on euclid. Thus if your file is named doe-3.lsp and it was created in an editor
    with tab width set to 4 (i.e., in an editor whose tab stops were 4 columns apart),
    add a comment like the following at the beginning of the file:
    ;;;; THIS FILE CAN BE VIEWED ON euclid USING: less -x4 doe-3.lsp

Use the less command to view your file on euclid; check that you have followed rules
1 - 9. If you use tabs in your file, make sure the comment you added for rule 10 appears
correctly when you view the file on euclid.