LISP Programming

LISP is one of the simplest computer languages in terms of syntax and semantics, and also one of the most powerful. It was developed in the mid-1950's by John McCarthy at M.I.T. as a "**LIS**t **P**rocessing language". Today, it is used for virtually all Artificial Intelligence programs and is the environment of choice for applications which require a powerful interactive working environment. LISP presents a very different way to think about programming from the "algorithmic" languages, such as BASIC, Fortran and Pascal.

As its name implies, the basis of LISP is a list. One constructs a list by enumerating elements inside a pair of parentheses. For example, here is a list with four elements (the second element is also a list):

(23 (this is easy) hello 821)

All statements in LISP are function calls with the following syntax: (*function* arg_1 arg_2 arg_3 ... arg_n). To evaluate a LISP statement, each of the arguments (possibly functions themselves) are evaluated, and then the function is invoked with the arguments. For example, (MULT (ADD 2 3) (ADD 1 4 2)) has a value of 35, since (ADD 2 3) has a value of 5, (ADD 1 4 2) has a value of 7, and (MULT 5 7) has a value of 35. Some functions have an arbitrary number of arguments; others require a fixed number. All statements return a value, which is either an atom or a list.

FUNCTION	RESULT
(ADD x_1 x_2 ...)	sum of all arguments
(MULT x_1 x_2 ...)	product of all arguments
(SUB *a b*)	*a-b*
(DIV *a b*)	*a*/*b*
(SQUARE *a*)	*a***a*
(EXP *a n*)	a^n
(EQ *a b*)	true if *a* and *b* are equal, NIL otherwise
(POS *a*)	true if *a* is positive, NIL otherwise
(NEG *a*)	true if *a* is negative, NIL otherwise

Some examples of these functions are as follows:

STATEMENT	VALUE
(ADD (EXP 2 3) (SUB 4 1) (DIV 54 4))	24.5
(SUB (MULT 3 2) (SUB 12 (ADD 2 2)))	-2
(ADD (SQUARE 3) (SQUARE 4))	25

LISP Programming - Worksheet

1. 01-02 C1 Lisp Programming 40
   Evaluate: (ADD (SUB 4 5) (ADD 6 3) (MULT 4 8))

2. 03-04 C1 Lisp Programming -10
   Evaluate: (MULT (ADD 2 3) (SUB 4 6))

3. 04-05 C1 Lisp Programming 1296
   Evaluate: (EXP (DIV (MULT (ADD 2 (SUB 4 2)) 3) 2) 4)

4. 05-06 C1 Lisp Programming 4
   Evaluate: (DIV (MULT (ADD 2 3) (SQUARE 2)) (SUB (EXP 2 3) (ADD 2 1)))

5. 06-07 C1 Lisp Programming 2
   Evaluate: (DIV (MULT (ADD 1 4 5) (SUB 7 2)) (EXP 5 2))

6. 07-08 C1 Lisp Programming 14
   Evaluate the following expression:
   (ADD (ADD 3 4) (SUB 5 2) (MULT 3 2) (EXP 2 3))

7. 08-09 C1 Lisp Programming 5
   Evaluate the following expression:
   (DIV (ADD (ADD 3 4) (MULT 4 2)) (SUB 8 5))

8. 09-10 C1 Lisp Programming 27
   Evaluate the following expression:
   (ADD (SUB 9 5) (EXP 2 3) (MULT (DIV 9 3) 5))