LISP FUNCTIONS

Summarized below are several basic functions used in the Lisp programming language. In some cases, both a short form and a long form are given (separated by or).

Basic List Operations
(car X) - return the head (first element) of the list X
(cdr X) - return the tail of the list X (X with the first element removed)
(cons X Y) - return the list created by inserting the list X onto the front of list Y
(append X Y) - return the list created by concatenating lists X and Y
(list X_1 X_2 ... X_n) - return the list created by concatenating lists X_1, X_2, ... X_n
(’ X) or (quote X) - return the list X (treated as a literal)

Arithmetic Operations
(+ E_1 E_2 ... E_n) - return E_1+E_2+ ... +E_n
(– E_1 E_2 ... E_n) - return E_1–E_2– ... –E_n
(* E_1 E_2 ... E_n) - return E_1*E_2* ... *E_n
(/ E_1 E_2 ... E_n) - return E_1/ E_2/ ... / E_n
(1+ X) - return X + 1
(1- X) - return X – 1

Relational Operations
(zerop X) - if X = 0 then return T else return NIL
(relation X Y) - if X relation Y then return T else return NIL (relation is =, /=, <, <=, >, >=)
(equal X Y) - if X = Y then return T else return NIL
(eq X Y) - if X and Y are the same object, then return T else return NIL

Logical Operations
(and X_1 X_2 ... X_n) - return X_1 ∧ X_2 ∧ ... ∧ X_n
(or X_1 X_2 ... X_n) - return X_1 ∨ X_2 ∨ ... ∨ X_n
(not X) - return ¬ X
Control Operations

(setq X Y) - assign the value of Y to X and return the value of Y
(set X Y) - same as setq except that X can be a computed value

(cond (test1 expression1)
     (test2 expression2)
     ...
     (testn expressionn))
     if test1 then return expression1
     else if test2 then return expression2
     ... else if testn then return expressionn

(print X) - print X

Function Manipulation Operations

(defun F (X1 X2 ... Xn) E) - define function F on arguments X1, ..., Xn to be the value of expression E

(apply F L) - apply the function denoted by F to the argument list L

(eval E) - evaluate the expression denoted by E

(prog (X1 X2 ... Xn) E1 E2 ... Em)
     define local variables X1, ..., Xn to be used in evaluating expressions E1, ..., Em, the value of the last being returned as the value of the PROG expression

(return E) - return the value of E (usually used in a prog expression)