微机 CAD 技术

——AutoLISP 的开发应用——

董仁扬 主编

成都科技大学
成都工具研究所
成都发动机公司

1988 年 6 月

前　言

微型计算机的计算机辅助设计（CAD），由于它的经济性，在我国得到了普遍的重视，在机器零件，工装（如：模具，刀具等）的 CAD 中已开发了一批实用性的软件。在计算机辅助设计中，微机 CAD 有它的地位和应用范围。

要发展微机 CAD 就要有一套好的支撑软件和与之相应的一套实用的CAD 技术。AutoCAD 是国内公认的较好的支撑软件，国内已开发的微机 CAD 应用软件几乎都是在 AutoCAD 上开发的。AutoCAD 只是一套基础软件，用它形成一套实用的 CAD 技术，还要进行二次开发，以发挥它的优势，弥补它的不足。近两年来，在我们用 AutoCAD 开发一批刀具，模具 CAD 应用软件的过程中，解决了大量普遍性的问题，建立了一批专用函数，扩展了 AutoCAD，形成了一套较完整的 CAD 技术。本书介绍我们摸索出的一套 CAD 技术，这套 CAD 技术是用 AutoLISP 语言来实现的。

本书有十三章和五个附录。

第一章介绍 LISP 语言的基本概念，是学习以下各章的基础。其中，数据值的概念，是掌握 Lisp 的一把钥匙。

第二，三，四，五，七章介绍 AutoLISP 的内部函数，这部分增强了 AutoLISP 手册中的内容，介绍了一些我们使用的经验和窍门，并定义了一批实用函数。

递归是使 LISP 程序简洁的一个重要手段，也是学习 LISP 语言的难点，第六章着重介绍递归及其应用。

LISP 的数组本来很麻烦，而 AutoLISP 根本没有数组函数。我们巧用赋值函数 SET 在 AutoLISP 中建立起了使用方便的数组。第八章介绍这部分内容。巧用 SET 函数贯穿以下各章，也是本书的一个特点。

AutoLISP 只能使用 45K 的内存空间，是使用 AutoLISP 的一大障碍。我们创造了一套无用单元的回收技术，使 200K～300K 的大程序也能在 AutoCAD 内运行，第九章介绍这一套技术。

目前还没有一套适合微机 CAD 的专用数据库系统。我们创造了一类数据函数，建立了一套微机 CAD 的数据库系统，并用 AutoLISP 开发了一套该数据库的管理系统，建立数据库的效率很高。

第十一章介绍用 AutoLISP 编制变参数图形库的规范化方法，并编制了一套自动生成对称点坐标的函数，自动标注尺寸的函数，自动标注表面粗糙

度和形位公差的函数。用它们开发变参数图形库效率高。

第十三章介绍用快速传递数据的方法，实现 AutoCAD 与高级语言的联接。它方法简便、效率高，不需要接口软件，且 AutoCAD 与任何一种高级语言联接方法是相同的。

本书是以 AutoCAD 2.5 版的 AutoLISP 为基础编写的，AutoCAD2.6 版中的 AutoLISP 只增加了几个函数，这在第十三章作了介绍。AutoCAD 9.03 版中的 AutoLISP 与 2.6 版相同，没有增加函数，只是增强了几个函数并建立一套库函数（即自定义函数），这些函数是看得见的。所以本书对于使用 AutoCAD 2.5及以上版本的用户都是实用的。

本书不仅对 AutoCAD 的用户有指导作用，也是学习 LISP 语言的一本入门书。可以作为举办 AutoCAD 的高级班的教材，也可供高等学校师生学习计算机辅助设计和 LISP 语言作参考。

本书由董仁扬主编。参加的人员有：《复杂刀具 CAD 系统开发》专题组的成都科技大学与成都工具研究所的同志和航空航天部成都发动机公司工装 CAD 课题组的同志。

目 录

第一章　AutoLISP 的数据与程序

AutoLISP 语言是一种嵌入 AutoCAD 内部的 LISP 编程语言，2.18 及以上的版本都配备有。它与 CommonLISP（通用 LISP）很相近，是它的一个小子集。但是它在这个基础上又增加了很多独有的处理图形的功能。因此，它不仅具备了 LISP 语言的基本特点，而且具有很强的图形功能，是微机 CAD 的强有力的工具。

LISP 语言是一种表处理语言，有很强的符号处理能力。它是为搞人工智能而发展起来的，因此也有人工智能数学之称，也称它为人工智能语言。经过二十多年的发展，它也具有很强的数值计算功能。AutoLISP 还有如下特点：

1. 它用原子和表代替数和数组作为它的基本的数据结构；
2. LISP 程序也是表的结构。同其它数据一样可以对它进行操作、运算；
3. 它无语句和命令，而是用函数作为处理数据的工具，而函数也是表的形式；
4. 它对表的结构和表的表示方法非常重视；
5. 它非常重视递归；
6. 它的存储单元是动态分配的，具有无用单元回收的技术；
7. 它具有灵活、机动的表达能力，能够进行自我扩充；
8. 它语法简单，易学易懂，容易掌握。但编高级程序时技巧性很强；
9. 用它编写的程序特别短。

1.1　AutoLISP 的数据结构

数据是语言处理的对象。原子和表是 LISP 语言的基本数据结构，是它的词。原子和表按一定形式组合构成了符号表达式（也称为 S- 表达式），以后简称表达式，它是 LISP 语言的句子。用这些句子就可以构成 LISP 程序。AutoLISP 的数据类型有：原子，包括符号原子、数原子和串原子；表；内部函数；文件描述符；选择集；图素名；功能分页表。

1.1.1　符号（Symbol）

符号也称符号原子，它是按任意顺序排列的可印刷字符。字符包括：英文字母、0～9 的数字和专用符 ·、-、$ 等。但 AutoLISP 有如下限制：

一．大小写英文字母不分，如：A 与 a 视为同一字符。

二．符号作为变量（也称标识符）或函数名时，不能以数字开头，如：A1、A-B、*A 是正确的，1A、2B 是错误的。

三．符号作为文件名时，只能以字母开头，数字和专用符都不能打头，如：3name、*name 都是错误的。

四．以下专用符在 LISP 中有特殊的含意，不能作符号中的字符使用。它们是：

（ ）----左右圆括号，它是表示表的专用符。

. ----小数点，它是数中的小数点，也是点对的专用符。

' ----右撇，它是禁止求值函数的简写符号。

" ----引号，是表示字符串的专用符。

; ----分号，它是 LISP 表示注释的专用符。在 LISP 程序的一个编辑行中，";"号以后的字符都被当作注释，LISP 不予理采。

1.1.2 整数（Integer）

它是 -32768～+32767 范围内的整数，数的前面可以有 "+" 或 "-" 号。整数和下面的实数又统称为数原子。

1.1.3 实数（Real）

实数一般又称为浮点数，在 AutoLISP 中它有四种表示形式。以 17.5 为例：

一．科学表示	1.75E+01	（即 1.75×10^{1}）
二．十进制表示	17.50	
三．英尺、小数点英寸	1'-5.50"	
四．英尺、分数寸	1'-5½"	

一般运算只用前面两者，用十进制表示时，小数点后均为六位，在 LISP 运算中不能用改变精度的函数改变它小数点后的位数。注意：小于 1 的数，其小数点前的 0 不能省略，如 .4 是错误的，必须写成 0.4。

用改变精度的函数 rtos 可以改变数的精度（即小数点后的位数），但这时实数已转换成字符串的形式，它只能作为字符串来使用（如用在尺寸标注中）。一旦将这字符串转换成实数，其小数点后位数又恢复成六位。

1.1.4 字符串（String）

字符串简称字串，也称串原子。它是用双引号 "……" 内的一串字符表示，如："abc"。字符可以是 ASCII 码表中的任一字符。注意：

一．在字符串中，大小英文字母是不同的字符。如："A" 和 "a" 是

两个不同的字符串。

二．键盘上没有的字符，可采用扩充的 ASCII 码字符输入键，例如要敲入 Φ，它的 ASCII 码为 327，按下 "Alt" 键不动，再由右边的数字键盘敲 327，然后松开 "Alt" 键，即在屏幕上显示 Φ，但这些字符只有在文本中有效。

三．字符串内可以包含控制字符，控制字符有：

\e 换码； \n 新行； \r 回车； \t 制表符 (tab)； \nnn 表示八进制码为 nnn 的字符。例如：要显示字符串 "\nStart point:"，它将换行以后显示：

Start point:

又如："\60\101\141\142\143\144\145\146\147\150\340" 显示：

"0Aabcdefgh a "

关于字符的八进制代码与 ASCII 的对照，载入附录 E 中。可以看出用控制符也可在字符串中输入全部 ASCII 码。

前面的符号原子、数原子、串原子统称为原子。

1.1.5 表 (List)

表是由一对圆括号 (……) 和括号中的若干元素组成，元素与元素间用空格分开。表的元素可以是各类原子，也可以是表，也可以是 S- 表达式，即表可以嵌套。没有元素的表称为空表。

表实质上是一种特殊的点对，点对的结构形式为：左括号 "(" 接左元素，空格接句点 "." (即小数点)，空格接右元素再接右括号 ")"。例如：

(A . B)　(A . (B . (C . D)))

即点对也能嵌套。嵌套的点对：

(A . (B . (… (Y . Z) …))

LISP 总把它记为：

(A B … Y . Z)

其中 A 是左元素，(B … Y . Z) 是右元素。注意：点对中句点 "." 前后必须有空格，否则 AutoLISP 将告诉你是非法的点对。

点对的最后一个元素为 nil (空)，则嵌套的点对就变成表，例如：上例中的 D 为 nil，则点对变为表，即：

(A . (B . (C . nil)))

变为表：(A B C)。

AutoLISP 支持表，也支持点对，例如：表示图素数据的表，就是用点

对的形式表示的。

嵌套的表可以组成高级表，这时表的结构是很重要的。不同结构的表可以表示数组（表格）、树、集合等等。LISP 的所有函数（包括自定义函数），以至 LISP 程序都是表的结构，所以 LISP 语言是一种表处理语言而不是“数值处理”语言。足见表在 LISP 语言中的地位。下面介绍几个重要的高级表。

一．联接表----A 表

联接表简称 A 表，它是一个点对表，即它的元素是点对。其元素的形式为：

(A . B) 或 (A B)

它的左元素称为点对的关键字，它的右元素是关键字的值。A 表的形式为：

((<关键字1> <值1>) (<关键字2> <值2>) …)

二．框架

框架用 A 表表示，它是一个嵌套的 A 表。形式如下：

(<框架名1> (<槽1> (<侧面1> (<值1> <值2> …))
(<侧面2> (<值1> <值2> …)) …)
(<槽2> (<侧面1> (<值1> <值2> …))
(<侧面2> (<值1> <值2> …)) …) …)

它的值可以用检索函数 ASSOC（见 2.3.10）检索，所以它是建立高级数据库的有用工具，见第 10 章。

三．用表表示数组或表格

数组或表格可以用不同结构形式的表表示，下面以矩阵为例。例如：

```
⎛  0    1.2  0    1.4  0  ⎞
⎜  0    0    0    0    0  ⎟
⎜ 3.1   0    0    0    3.5⎟
⎝  0    4.2  4.3  0    0  ⎠
```

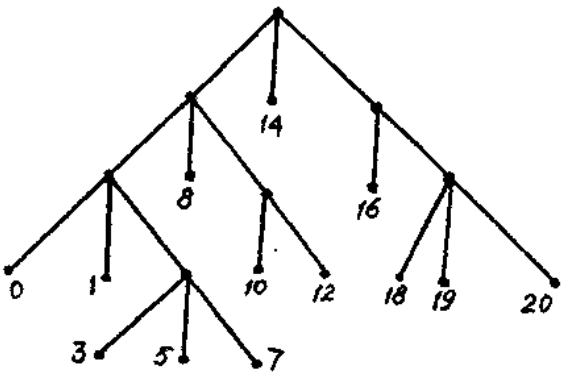
可以表示为 ((1 ((2 1.2) (4 1.4))) ; 第一行
(3 ((1 3.2) (5 3.5))) ; 第三行
(4 ((2 4.2) (3 4.3)))) ; 第四行

也可表示为 ((1 2 1.2) (1 4 1.4) ; 第一行
(3 1 3.1) (3 5 3.5) ; 第三行
(4 2 4.2) (4 3 4.3)) ; 第四行

四．用表表示树

树的形式如右图，也可以用表的形式将它的结构表示出来。

(((0 1 (3 5 7))
8 (10 12))
14 (16 (18 19 20)))



还可以定义一些函数对上述高级表进行操作或运算，这在后面将会看到。

在使用表时，须特别注意的是：左右括号必须在需要的位置配对。在调试程序时，所出的错误中，括号位置不对所占的比重很大，这是 LISP 语言比较麻烦的地方。

1.1.6 AutoLISP 的内部函数 (SUBR)

AutoLISP 的内部函数也是 AutoLISP 的一种数据类型，它实际上是一个子程序号，如：加法函数为 <Subr: #5356>，赋值函数 setq 为 <Subr: #82FC>。它确定了 AutoLISP 的功能，函数愈多，AutoLISP 的功能愈强。AutoLISP 的内部函数：2.18 版共有 122 个；2.5 版共有 146 个；2.6 版共有 152 个。这些函数的功能和用法，是本书的主要内容之一。

1.1.7 文件描述符 (FILE)

它是由 open 函数打开文件后产生的文件描述符，其形式如：< File: #D4C8 >、< File: #70C8 > 等，它表示被打开的文件在磁盘中的位置。它也是 AutoLISP 的一种数据类型。

1.1.8 AutoLISP 的图素名称 (ENAME) (2.5 版)

它表示图形数据库中图素的名称，其形式如:<Entity name: 60000014> 等。实际上，它表明了某图素在图形数据库中所占的位置。它也是 AutoLISP 的一种数据类型。

1.1.9 AutoLISP 的选择集 (PICKSET) (2.5 版)

它是图素名称的简单集合，其形式如象:<Selection set: 2> 等. 它也是 AutoLISP 的一种数据类型。

1.1.10 函数分页表 (PAGETB) (2.5 版)

请参阅 9.3 节页式虚拟储存。

1.2 AutoLISP 数据的值

LISP 解释程序的核心就是其求值程序，它对 AutoLISP 的各型数据求值。

LISP 程序的执行过程就是反复地读入原子和表，对它们进行求值，然后输出这些值的过程。所以，值是 LISP 的一个极重要的概念，要熟练地掌握和灵活地运用 AutoLISP 语言，必须对它的各类数据和表达式的值以及各种函数的值（即它的返回值）有明确的概念。AutoLISP 各类数据的值如下：

一．整数、实数、字符串、文件描述符、AutoLISP 的内部函数、图素名称、选择集、函数分页表等的值是它自身。如：123 的值就是123，"abc" 的值就是 "abc"。

二．符号的值是它当前的结合值。符号必须有值，否则会出错。符号的当前值是由赋值函数 setq 或 set 赋给的。给符号作第二次赋值，则用新值取代旧值。

T 和 nil 是两个特殊的符号，它的值是预先设置的，它的值是它本身。T 的值是 T，表示逻辑真；nil 的值是 nil，表示逻辑假或空表 "( )"。所以，用符号作变量时，不要用 T 或 NIL，否则容易出错。

三．表的值是它的第一个元素的值。求值表实际上是函数调用，它的第一个元素必须是函数名称，其余元素是它的自变量（它的值将作为实参），整个表被假定为函数定义。求值表的形式为：

(<函数名称> <自变量1> <自变量2> …)

<函数名称> 可以是 AutoLISP 的内部函数名，也可以是用户用 defun 定义的自定义函数名。例如：

(+ 10 20 30)

"+" 是 AutoLISP 的内部函数，作加法运算，10、20、30 是它的自变量，其值为它们的和 60。

求值表的值，即函数调用所得的值，称为它的返回值。

求值表还有另一种形式，即它的第一个元素是表，则这个表就作为函数的定义，其余元素是它的自变量。例如：

((+ N 1) 5)　　返回值 6

第一个元素 (+ N 1) 就是一个函数定义，它表示数 N 加 1，N 是形参。5 是实参。执行的结果是 5 加 1 等于 6。

一般意义的表必须是禁止求值的表，这时它的元素也就自然是禁止求值的。关于求值与禁止求值见 2.4。

1.3　AutoLISP 程序

LISP 程序是一种表的结构，即它是由若干表或嵌套表组合而成。

AutoLISP 程序都是用行编辑编写的。它有三种文本形式：LISP 文本（扩展名为 .lsp）；命令文本（扩展名为 .scr）；菜单文本（扩展名为 .mnu）。它们的格式、装入和运行方式都不同，且各有其特点。分别介绍如下：

1.3.1 LISP 文本

扩展名为 .lsp 是 lisp 文本的标志。lisp 文本是严格按照 lisp 语言的规则编写的文本，用 AutoLISP 编写时同其它 LISP 语言编写的格式相同。它与其它两种文本的区别主要是：

一．处理 AutoCAD 命令的方式不同

它必须将 AutoCAD 命令通过命令函数 command 转换成 lisp 函数才能使用，且点的表示方法、点的求值方式、连续空格的表示方式均不同。详见 3.1.1。

二．装入和调用方式不同

它必须用 load 函数装入计算机内才能执行，执行的方式有三种：若 lisp 文本是自定义函数的形式，则按函数的调用来运行；若 lisp 文本是自定义命令的形式，则按 AutoCAD 命令的方式使用；若不是上两种方式，则装入后立即执行。详见 7.2.3。

三．可以使用键输入函数进行人 - 机对话。详见 7.2。

四．lisp 文本在行编辑中的一个编辑行允许 255 个字符，超出部分将被丢掉。

lisp 文本的优点除能进行人 - 机对话外，主要是运行效率高，运行速度一般比其它两种文本快 2～3 倍。缺点是占 AutoLISP 空间较多。

1.3.2 命令文本

扩展名为 .scr 是命令文本的标志。它是由 2.18 以前版本中的变量与表达式发展起来的，它将 AutoLISP 函数与 AutoCAD 命令各按自己的格式混合编在一起。它颇象批处理文件，调入一句执行一句。它的特点是：

一．其中 lisp 函数的调用严格按照 lisp 语言的规则编写；AutoCAD 命令按它在屏幕上的调用方式编写。在 AutoCAD 命令中，用惊叹号“!”调值，坐标点用 X,Y 的方式输入，允许使用连续空格。空格在这三种文本中都很重要，它相当于回车，在调试 CAD 的程序中，空格不对的错误占的比重也很大，且不太好检查。

二．它运行前不须装入，直接用命令文件调用命令 script 调用，执行时调入一句执行一句。

三．命令文本的一个编辑行不能超过 80 个字符，超过部分将被丢掉，

须特别注意。

命令文本明显的优点是占 AutoLISP 的空间少。它的缺点是不能进行人 - 机对话，运行效率也较低。

LISP 文本中可以调用一个命令文本，而命令文本可调用多个 lisp 文本。若将它们混合使用，各取所长，则程序将更加合理。一般，lisp 文本用于主程序，进行人 - 机对话和组织调度子程序，以及用于 AutoCAD 与 AutoLISP 功能的扩充。大量的计算也可用 lisp 文本。而命令文本主要用于绘图程序和为绘图服务的少量计算。

1.3.3 菜单文本

扩展名为 .mnu 是菜单文本的标志，这里指的菜单文本是称为长菜单文件的文本。菜单文件是 AutoCAD 用来建立各种用户菜单而设立的，如用它来建立屏幕菜单、数值化板菜单等。菜单文件中的长菜单项也可包含若干的 AutoCAD 命令和 AutoLISP 函数，其作用有点象命令文本，但它的格式、运行方式均与命令文本不同。菜单文本的格式及规则在 AutoCAD 手册的附录 B 中已有详细的叙述，这里就不重复了。菜单文本用调菜单文件的命令 menu 装入，按菜单的运行方式运行。

第二章 基本函数

函数是 LISP 语言处理数据的工具，它起作一些计算机语言（如：BASIC、FORTRAN 等）中语句或命令的作用。函数是一张求值的表，一般格式为：

（<函数名> <自变量1> <自变量2> … ）

使用函数时要注意函数要求的自变量数目和每个自变量的数据类型，否则将出错。LISP 函数可以是 AutoLISP 的内部函数或用户自定义的函数。内部函数是 AutoLISP 本身所具有的函数；自定义函数是用户根据需要用 lisp 语言编写的函数，一般它起作子程序的作用。从本章起用几章来叙述AutoLISP的内部函数及我们使用它们的经验和窍门。

2.1 计算函数

它是处理数值计算的函数。lisp 的计算函数和 lisp 的其它函数一样，都采用前缀表示，即运算符为表的第一个元素。

2.1.1 连加

（+ <数> <数> … ）

它的返回值（即函数的值）是所有 <数> 的总和。<数> 可以是整数，也可以是实数。若所有 <数> 是整数，其结果也是整数；若其中一个 <数> 是实数，则其它的整数将转变为实数，结果也是实数。自变量数目不限，且 <数> 可以带符号（以下同）。例如：

(+ 1 2)	返回值	3
(+ 1 2 3 4.5)	返回值	10.500000
(+ 1 2 3 4.0)	返回值	10.000000
(+ 1 2 3 -4.0)	返回值	2.000000

2.1.2 连减

（- <数> <数> … ）

第一个 <数> 是被减数，其余的 <数> 为减数。它的返回值是第一个数减去其余 <数> 的和。若自变量只有一个，则将这个 <数> 变为负数。其 <数> 的数据类型与加法相同，例如：

(- 50 40)	返回值	10
(- 50 40.0 2.5)	返回值	7.500000
(- 8)	返回值	-8

2.1.3 连乘

(* <数> <数> …)

它的返回值是所有 <数> 的连乘积，其 <数> 的类型与加法相同。例如：

(* 2 3)　　返回值　6

(* 2 3 4.0)　　返回值　24.000000

(* 3 -4.5)　　返回值　-13.500000

2.1.4 连除

(/ <数> <数> <数> …)

第一个 <数> 是被除数，其余的 <数> 为除数。返回值是它的商。注意：整数除整数仍然是整数，若商为实数则小数被省去，商小于 1 为零，例如：

(/ 10.0 2 4.0)　　返回值　1.250000

(/ 135 360)　　返回值　0

2.1.5 加一

(1+ <数>)

它的返回值是将 <数> 增加 1 。<数> 可以是整数，也可以是实数。例如：

(1+ 5)　　返回值　6

(1+ -17.5)　　返回值　-16.500000

2.1.6 减一

(1- <数>)

它的返回值是将 <数> 减少 1 。<数> 可以是整数或实数。例如：

(1- 5)　　返回值　4

(1- -17.5)　　返回值　-18.500000

2.1.7 取绝对值

(abs <数>)

它的返回值为 <数> 的绝对值。<数> 可以是实数或整数。例如：

(abs 100)　　返回值　100

(abs -100)　　返回值　100

(abs -99.25)　　返回值　99.250000

2.1.8 e^x

(exp <数>)

它返回 e 的 <数> 次方，结果为实数。<数> 是 e^x 的次方 x，可为实数或整数；e 是自然对数的底。例如：

(exp 1)　　返回值　2.718282

(exp 2.2)　　返回值　9.025013

(exp -0.4)　　返回值　0.670320

2.1.9 a^x

(expt <数1> <数2>)

返回值为 <数1> 的 <数2> 次方，<数1> 与 <数2> 均为整数时其结果为整数，否则为实数。<数1> 为 a^x 中的底 a，<数2> 为 a^x 中的幂 x。例如：

(expt 2 4)　　返回值　16

(expt 3 2.0)　　返回值　9.000000

2.1.10 平方根

(sqrt <数>)

它的返回值是 <数> 的实平方根，结果是实数。而 <数> 可以是整数或实数。例如：

(sqrt 4)　　返回值　2.000000

(sqrt 2.0)　　返回值　1.414214

2.1.11 自然对数

(log <数>)

它的返回值是 <数> 的自然对数（以 e 为底），结果是实数。例如：

(log 4.5)　　返回值　1.504077

2.1.12 最大公约数

(gcd <整数1> <整数2> …)

它的返回值是多个 <整数> 的最大公约数，结果是整数。例如：

(gcd 81 37)　　返回值　3

(gcd 12 20 28 44)　　返回值　4

它可以有多个自变量，这在一般 AutoLISP 手册上没有明确指出。

2.1.13 余数

(rem <数1> <数2> …)

若自变量为两个，则返回值为 <数1> 除以 <数2> 的余数；若自变量为多个，则前两个数的余数再与后一个数求余数，以此类推，其返回值为最后求得的余数。<数> 可以是整数，也可以是实数，若都为整数，结果为整数，其中一个为实数，结果为实数。例如：

(rem 42 12)　　返回值　6

(rem 42 12 4)　　返回值　2

(rem 12.0 16)　　返回值　12.000000

(rem 12.0 16 7)　　返回值　5.000000

(rem 60 3)　　返回值　0

2.1.14 最大数

(max <数1> <数2> …)

它的返回值为 <数> 中的最大值，<数> 可以是整数或实数。例如：

(max 4.07 -144)　　返回值　4.070000

(max -88 19 5 2)　　返回值　19

2.1.15 最小数

(min <数1> <数2> …)

它的返回值为 <数> 中的最小值。<数> 可以是整数或实数。例如：

(min 683 -10.0)　　返回值　-10.000000

(min 73 2 48 5)　　返回值　2

2.1.16 逻辑位移

(lsh <整数1> <整数2>)

<整数1> 是被位移的整数，<整数2> 是位移的位数。它的执行过程是，先将 <整数1> 转化为二进制数，然后按位数左移或右移，最后将位移后的二进制数再转回十进制数，它的返回制就是这个十进制数。若 <整数2> 为正数，则向左移，为负数则向右移，且移入位为零，移出位丢弃。移入时若二进制数超出最高位（第十六位），<整数1> 将改变符号。

该函数执行的结果是，左移一位 <整数1> 扩大一倍，右移一位缩小一倍，结果是整数。例如：

(lsh 2 1)　　返回值　4

因 2 的二进制为 10，左移一位为 100，它的十进制为 4，4 就是返回值。

(lsh 2 -1)　　返回值　1

因它的二进制是 10，右移一位变为 1，它的十进制也是 1。

(lsh 40 2)　　返回值　160

(lsh 16384 1)　　返回值　-32768

2.1.17 正弦函数

(sin <角>)

它的返回值是 <角> 的正弦值，<角> 以弧度表示。例如：

(sin 1.0)　　返回值　0.841471