

Chapter 8

列表操作

列表是一种计算器对象，可用于数据处理和编程。本章介绍了列表操作的示例。

Definitions

在计算器的上下文中，列表是一系列对象，它们在大括号之间用空格 (SPC), RPN模式或逗号 (→ ,), 在两种模式下。可以包括在列表中的对象是数字，字母，字符串，变量名称和/或运算符。列表对于操作数据集和某些编程应用程序非常有用。列表的一些示例是：

```
{ t 1 }, { "BETA" h2 4 }, { 1 1.5 2.0 },  
{ a a a a }, { { 1 2 3 } { 3 2 1 } { 1 2 3 } }
```

在下面显示的示例中，我们将仅限于数字列表。

创建和存储列表

要在ALG模式下创建列表，首先输入大括号键 (↵) (associated with the (+) key), 然后键入或输入列表中的元素，用逗号(→ ,)。以下击键将进入列表{1 2 3 4}并将其存储到变量L1中。

↵ () / → , 2 → , 3 → , 4
▶ STO▶ ALPHA (L) / ENTER

The screen will show the following:

```
{1,2,3,4}▶L1  
+SKIP|SKIP+|+DEL|DEL+|DEL L|INS=
```

```
{1. 2. 3. 4.}▶L1  
{1. 2. 3. 4.}  
+SKIP|SKIP+|+DEL|DEL+|DEL L|INS=
```

左边的图显示了按 (ENTER) 之前的屏幕，而右边的图显示了将列表存储到L1后的屏幕。 请注意，在按下 (ENTER) 之前，列表会显示分隔其元素的逗号。 但是，在按下(ENTER)之后，逗号将替换为空格。

在RPN模式下输入相同的列表需要以下按键：

↵ () / SPC 2 SPC 3 SPC 4 ENTER
ALPHA (L) / ENTER STO▶

下图显示了在按下 **(STO▶)** 键之前的RPN堆栈:

5:					
4:					{1. 2. 3. 4.}
3:					'L1'
2:					
1:					
EQ Mpar LVARI TITLE a b					
TRIAD MES1 STRS PROJ GRAV PIPE					

编写和分解列表

组合和分解列表仅在RPN模式下有意义。在这种操作模式下，通过使用函数 **OBJ→** 来实现分解列表。使用此函数，RPN堆栈中的列表将分解为其元素，堆栈级别为1: 显示列表中的元素数。接下来的两个屏幕截图显示了在应用功能**OBJ→**之前和之后带有小列表的堆栈:

4:					
3:					
2:					
1:					{3 -2 0}
EQ Mpar LVARI TITLE a b					

4:					
3:					
2:					
1:					3
EQ Mpar LVARI TITLE a b					

请注意，在应用**OBJ→**之后，列表的元素占用级别4: 到2:，而级别1: 显示列表中的元素数量。

要在RPN模式下组成列表，请将列表的元素放在堆栈中，输入列表大小，然后应用函数 **→LIST** (从函数目录中选择它，如下所示: **(F) CAT (F) →**，然后使用上下箭头键 (**▲ ▼**) 定位函数 **→LIST**)。以下屏幕截图显示了在应用函数 **→LIST**之前和之后的4个列表的元素列表:

5:					
4:					
3:					
2:					
1:					
EQ Mpar LVARI TITLE a b					

5:					
4:					
3:					
2:					
1:					{5 6 3 2}
EQ Mpar LVARI TITLE a b					

Note: Function **OBJ→** 应用于ALG模式的列表只需再现列表，并向其添加列表大小:

:OBJ→({3 2 -1})					
					{3 2 -1 3.}
EQ Mpar LVARI TITLE a b					

使用数字列表进行操作

为了演示带有数字列表的操作，我们将创建一些其他列表，除了上面创建的列表L1: L2 = { - 3,2,1,5}，L3 = { - 6,5,3,1,0, 3, -4}，L4 = {3, -2,1,5,3,2,1}。在ALG模式下，输入列表L2, L3, L4后屏幕将如下所示:

```

: (-3 2 1 5) ► L2
: (-6 5 3 1 0 3 -4) ► L3
: (3 -2 1 5 3 2 1) ► L4
: (3 -2 1 5 3 2 1)
L4 | L3 | L2 | L1 | TRIAN MES1

```

在RPN模式下，以下屏幕显示准备存储的三个列表及其名称。要在这种情况下存储列表，您需要按 **(STOP)** 三次。

改变标志

当应用于数字列表时，符号更改键 **(+/-)** 将更改列表中所有元素的符号。例如：

```

: L1
: (1. 2. 3. 4.)
: -L1
: (-1. -2. -3. -4.)
L2 | L3 | L4 | L1 | TRIAN MES1

```

加法，减法，乘法，除法

通过单个数字对列表进行乘法和除法分布在列表中，例如：

```

: -5 L2
: (15 -10 -5 -25)
: L1 / 5
: (.2 .4 .6 .8)
L2 | L3 | L4 | L1 | TRIAN MES1

```

从列表中减去单个数字将从列表中的每个元素中减去相同的数字，例如：

```

: L2
: (-3 2 1 5)
: L2 - 10
: (-13. -8. -9. -5.)
L2 | L3 | L4 | L1 | TRIAN MES1

```

在列表中添加单个数字会生成一个由数字增加的列表，而不是向列表中的每个元素添加单个数字。例如：

```

: L1
: (1 2 3 4)
: L1 + 6
: (1 2 3 4 6)
L2 | L3 | L4 | L1 | TRIAN MES1

```

相同长度的数字列表的减法，乘法和除法产生具有逐项运算的相同长度的列表。例子：

:L1-L2						:L1-L2					
						(4. 0. 2. -1.)					
:L1.L2						(-3. 4. 3. 20.)					
L2	L3	L4	L1	TRIAN	MES1	L2	L3	L4	L1	TRIAN	MES1

除法L4 / L3将产生无穷大项，因为在L3的要素之一是零：

:L4					
:L3					
(-1 -2 1 5 * 2 -1)					
(2 5 3 3 3 4)					
L2	L3	L4	L1	TRIAN	MES1

如果操作中涉及的列表具有不同的长度，则会生成错误消息（错误：无效维度）。

当应用于列表时，加号 (⊕) 作为连接运算符，将两个列表放在一起，而不是逐个添加它们。例如：

:L1+L2					
(1 2 3 4 -3 2 1 5)					
L2	L3	L4	L1	TRIAN	MES1

为了逐个添加两个相同长度的列表，我们需要使用运算符ADD。可以使用函数目录 (⏏_CAT) 加载此运算符。下面的屏幕显示ADD的应用程序，用于逐个添加列表L1和L2：

:L1 ADD L2					
(-2 4 4 9)					
L2	L3	L4	L1	TRIAN	MES1

键盘上的实数功能

键盘上的实数功能 (ABS, e^x, LN, 10^x, LOG, SIN, x², √, COS, TAN, ASIN, ACOS, ATAN, y^x) 可用于列表。这里有些例子：

ABS

```
:L2
(-3 2 1 5)
:IL2I
(3 2 1 5)
L2 | L3 | L4 | L1 | TRIAN | MES1
```

EXP and LN

```
:e L1
(e1 e2 e3 e4)
:LN(L1)
(0 LN(2) LN(3) 2·LN(2))
L2 | L3 | L4 | L1 | TRIAN | MES1
```

LOG and ANTILOG

```
:LOG(L1)
(0 LOG(2) LOG(3) LOG(4))
:ALOG(L2)
(1/1000 100 10 100000)
L2 | L3 | L4 | L1 | TRIAN | MES1
```

SQ and square root

```
:SQ(L1)
(1 4 9 16)
:√L2
(√-1 √3 √2 1 √5)
L2 | L3 | L4 | L1 | TRIAN | MES1
```

SIN, ASIN

```
:SIN(L1)
(SIN(1) SIN(2) SIN(3) SIN(4))
:ASIN(L2)
(-.304692654015 .20135)
L2 | L3 | L4 | L1 | TRIAN | MES1
```

COS, ACOS

```
:COS(L2)
(COS(3) COS(2) COS(1) COS(5))
:ACOS(L1)
(1.47062890563 1.36943)
L2 | L3 | L4 | L1 | TRIAN | MES1
```

TAN, ATAN

```
:TAN(L1)
(TAN(1) TAN(2) TAN(3) TAN(4))
:ATAN(L2)
(-ATAN(3) ATAN(2) π/4 ATAN(4))
L2 | L3 | L4 | L1 | TRIAN | MES1
```

INVERSE (1/x)

```
:INV(L1)
(1 1/2 1/3 1/4)
L2 | L3 | L4 | L1 | TRIAN | MES1
```

MTH菜单中的实数功能

从MTH菜单中感兴趣的功能包括HYPERBOLIC菜单: SINH, ASINH, COSH, ACOSH, TANH, ATANH和REAL菜单: %, %CH, %T, MIN, MAX, MOD, SIGN, MANT, XPON, IP, FP, RND, TRNC, FLOOR, CEIL, D→R, R→D。采用单个参数的一些函数如下所示应用于实数列表:

SINH, ASINH

```
:SINH(L1)
(SINH(1) SINH(2) SINH(3) S)
:ASINH(L2)
(-.295673047563 .19869)
SINH | ASINH | COSH | ACOSH | TANH | ATANH
```

COSH, ACOSH

```
:COSH(L2)
(COSH(3) COSH(2) COSH(1) C)
:ACOSH(L1)
(0 ACOSH(2) ACOSH(3) ACOS)
SINH | ASINH | COSH | ACOSH | TANH | ATANH
```

TANH, ATANH

```
:TANH(L2)
(-TANH(3) TANH(2) TANH(1) )
:ATANH(L1)
(ATANH(1) ATANH(2) ATANH(3))
SIGN | ASINH | COSH | ACOSH | TANH | ATANH
```

SIGN, MANT, XPON

```
:SIGN(L1)
(1 1 1 1)
:MANT(100/L2)
(3. 2. 1. 5.)
:XPON(L1/100)
(2. 2. 2. 2.)
ABS | SIGN | MANT | XPON | IP | FP
```

IP, FP

```
:IP((1.2 2.3 -1.5))
(1. 2. -1.)
:FP((1.2 2.3 -1.5))
(.2 .3 -.5)
ABS | SIGN | MANT | XPON | IP | FP
```

FLOOR, CEIL

```
:FLOOR((1.2 2.3 -1.5))
(1. 2. -2.)
:CEIL((1.2 2.3 -1.5))
(2. 3. -1.)
RND | TRNC | FLOOR | CEIL | D→R | R→D
```

D→R, R→D

```
:D→R((30 60 90))
(.523598775598 1.04719)
:R→D(( $\frac{\pi}{6}$   $\frac{\pi}{3}$   $\frac{\pi}{2}$ ))
(30. 60. 00000000002 90.0)
RND | TRNC | FLOOR | CEIL | D→R | R→D
```

使用两个参数的函数示例

下面的屏幕截图显示了函数%的应用列表参数。函数%需要两个参数。前两个示例显示了两个参数中只有一个是列表的情况。

```
:%({10 20 30},1)
(.1 .2 .3)
:%(5,{10 20 30})
(5.1 5.2 5.3)
Z | ZCH | ZT | MIN | MAX | MOD
```

结果是根据list参数分配函数%的列表。例如，

$$\%(\{10, 20, 30\}, 1) = \{\%(10, 1), \%(20, 1), \%(30, 1)\},$$

while

$$\%(5, \{10, 20, 30\}) = \{\%(5, 10), \%(5, 20), \%(5, 30)\}$$

在以下示例中，函数%的两个参数都是相同大小的列表。在这种情况下，执行参数的逐项分布，即，

$$\%(\{10,20,30\},\{1,2,3\}) = \{\%(10,1),\%(20,2),\%(30,3)\}$$

```

: %({10 20 30},{1 2 3})
      {10·1 20·1 30·3 }
      {10·100 20·50 30·100}
2 | ZCH | ZT | MIN | MAX | MOD

```

列表参数的函数%的这种描述显示了当一个或两个参数是列表时，具有两个参数的任何函数的一般评估模式。功能RND的应用示例如下所示：

```

: RND({1/3 1/6 1/3},2)
      { .33 .17 .33 }
: RND(1/3,{2 3 4})
      { .33 .333 .3333 }
INTEG POLY MODUL PERM DIVIS FACTO

```

复数列表

下面的练习展示了如何创建一个复数列表，给出两个相同长度的列表，一个表示实部，另一个表示复数的虚部。使用L1 ADD i·L2。

```

RAD MYZ HEX C= 'X'      ALG
CHOME3
      {1/3 1/6 1/3}
      { .33 .17 .33 }
: RND(1/3,{2 3 4})
      { .33 .333 .3333 }
: L1 ADD i·L2
      {1+i-3 2+i·2 3+i 4+i·5}
L4 | L3 | L2 | L1 | CASDI

```

诸如LN，EXP，SQ等的功能也可以应用于复数列表，例如，

```

: SQ(L5)
(SQ(1+i-3) SQ(2+i·2) SQ(3+i·5))
: JL5
      { ((3+i)·√2-2·i·√5)·√1+√10 } (1)
L5 | L2 | L3 | L4 | L1 | TRIAN

```

```

: L5
(e^{1+i-3} e^{2+i·2} e^{3+i 4+i·5})
: LN(L5)
(LN(1+i-3) LN(2+i·2) LN(3+i·5))
L5 | L2 | L3 | L4 | L1 | TRIAN

```

```

:ALOG(L5)
:(ALOG(1+i,-3) ALOG(2+i,2) ▶
:LOG(L5)
:(LOG(1+i,-3) LOG(2+i,2) LO▶
:INV(L5)
:
$$\left\{ \frac{1}{1+i-3} \frac{1}{2+i,2} \frac{1}{3+i} \frac{1}{4+i,5} \right\}$$

L5 L2 L3 L4 L1 TRIAN

```

```

:SIN(L5)
:(SIN(1+i,-3) SIN(2+i,2) SI▶
: SINH(L5)
:(SINH(1+i,-3) SINH(2+i,2) ▶
:ASIN(L5)
:(ASIN(1+i,-3) ASIN(2+i,2) ▶
L5 L2 L3 L4 L1 TRIAN

```

以下示例显示复数的函数RE（实部），IM（虚部），ABS（幅度）和ARG（参数）的应用。结果是实数列表：

```

:RE(L5)
:IM(L5)
:IL5)

$$\left\{ \sqrt{10} \ 2\sqrt{2} \ \sqrt{10} \ \sqrt{41} \right\}$$

L5 L2 L3 L4 L1 TRIAN

```

```

:ARG(L5)

$$\left\{ -\text{ATAN}(3) \ \frac{\pi}{4} \ \text{ATAN}\left(\frac{1}{3}\right) \ \text{ATAN}\left(\frac{1}{3}\right) \right\}$$

L5 L2 L3 L4 L1 TRIAN

```

代数对象列表

以下是应用SIN函数的代数对象列表的示例：

```

:
$$\left\{ \frac{f}{2} \ \alpha-\beta \ \frac{(x-y)^2}{4} \right\}$$


$$\left\{ \frac{f}{2} \ \alpha-\beta \ \frac{(x-y)^2}{4} \right\}$$

L5 L2 L3 L4 L1 TRIAN

```

```

:
$$\left\{ \frac{f}{2} \ \alpha-\beta \ \frac{(x-y)^2}{4} \right\}$$

:SIN(ANS(1))

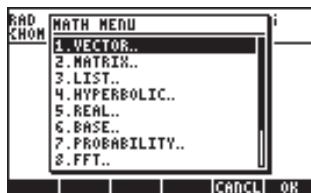
$$\left\{ \text{SIN}\left(\frac{f}{2}\right) \ \text{SIN}(\alpha-\beta) \ \text{SIN}\left(\frac{(x-y)^2}{4}\right) \right\}$$

L5 L2 L3 L4 L1 TRIAN

```

The MTH/LIST menu

MTH菜单提供了许多专门用于列表的功能。将标志117设置为CHOOSE框：



接下来，系统标志117设置为SOFT菜单：

VECTRMATR: LIST MYP REAL BASE

ΔLIST ELIST MLIST SORT REVL ADD

此菜单包含以下功能：

- ΔLIST : 计算列表中连续元素之间的增量
- ΣLIST : 计算列表中元素的总和
- ΠLIST : 计算列表中元素的乘积
- SORT : 按顺序对元素进行排序
- REVLIST : 撤消列表顺序
- ADD : 运算符用于逐个添加两个相同长度的列表（上面显示了此运算符的示例）

下面显示了在ALG模式下应用这些功能的示例：

```

:L3      {-6 5 3 1 0 3 -4}
:ΔLIST(L3)
      {11 -2 -2 -1 3 -7}
ΔLIST ELIST MLIST SORT REVL ADD
:L3      {-6 5 3 1 0 3 -4}
:ΣLIST(L3)
      {-6 -4 0 1 3 3 5}
ΔLIST ELIST MLIST SORT REVL ADD

```

```

:L3      {-6 5 3 1 0 3 -4}
:ΣLIST(L3)
      2
ΔLIST ELIST MLIST SORT REVL ADD
:L3      {-6 5 3 1 0 3 -4}
:REVLIST(L3)
      {-4 3 0 1 3 5 -6}
ΔLIST ELIST MLIST SORT REVL ADD

```

可以组合SORT和REVLIST以按递减顺序对列表进行排序：

```

:L3      {-6 5 3 1 0 3 -4}
:REVLIST(SORT(L3))
      {5 3 3 1 0 -4 -6}
ΔLIST ELIST MLIST SORT REVL ADD

```

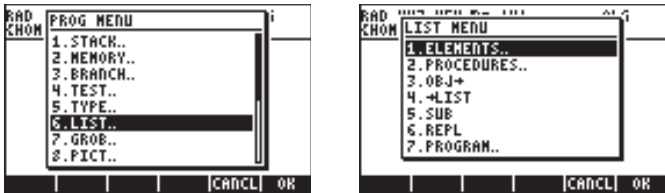
如果您在RPN模式下工作，请将列表输入堆栈，然后选择所需的操作。例如，要计算列表L3中连续元素之间的增量，请按：

[L] [3] [ENTER] [←] MTH [▽] [▽] [OK] [OK]

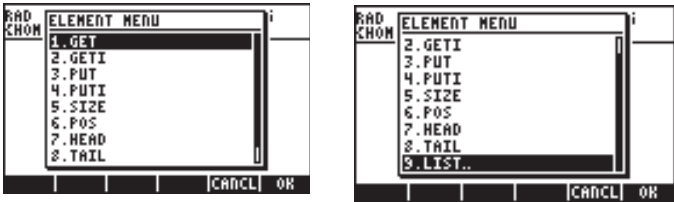
这将L3置于堆栈上，然后从MTH菜单中选择 ΔLIST操作。

操纵列表的元素

PRG（编程）菜单包括LIST子菜单，其具有许多用于操纵列表元素的功能。系统标志117设置为CHOOSE框：

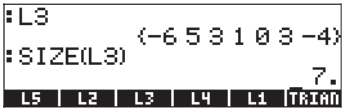


Item 1. ELEMENTS ..包含以下可用于操作列表中元素的函数：



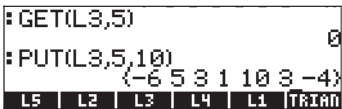
List size

来自PRG / LIST / ELEMENTS子菜单的函数SIZE可用于获得列表的大小（也称为长度），例如，



在列表中提取和插入元素

要提取列表的元素，我们使用PRG / LIST / ELEMENTS子菜单中的函数GET。函数GET的参数是列表和要提取的元素的编号。要将元素插入列表，请使用功能PUT（也可在PRG / LST / ELEMENTS子菜单中使用）。函数PUT的参数是列表，要替换的位置以及要替换的值。功能GET和PUT的应用示例如下面的屏幕所示：



功能GETI和PUTI，也可在子菜单PRG / ELEMENTS /中使用，也可用于提取和放置列表中的元素。然而，这两个函数主要用于编程。函数GETI使用与GET相同的参数并返回列表，元素位置加1，以及请求位置的元素。函数PUTI使用与GET相同的参数，并返回列表和列表大小。

列表中的元素位置

为了确定列表中元素的位置，使用具有列表和感兴趣元素作为参数的函数POS。例如，

```
:L3
      { -6 5 3 1 0 3 -4 }
:POS(L3,5)
      2.
L5 | L2 | L3 | L4 | L1 | TRAIL
```

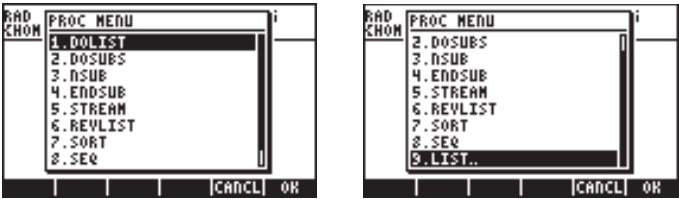
HEAD and TAIL functions

HEAD函数提取列表中的第一个元素。TAIL函数删除列表的第一个元素，返回剩余的列表。下面显示了一些示例：

```
:L3
      { -6 5 3 1 0 3 -4 }
:HEAD(L3)
      -6
:TAIL(L3)
      { 5 3 1 0 3 -4 }
HEAD | TAIL |      |      | LIST
```

The SEQ function

项目2. PRG / LIST菜单中的PROCEDURES ..包含以下可用于对列表进行操作的功能。



函数REVLIST和SORT早先作为MTH / LIST菜单的一部分介绍。函数DOLIST, DOSUBS, NSUB, ENDSUB和STREAM被设计为用于在RPN模式下操作列表的编程函数

函数SEQ可用于生成给定特定表达式的值列表，并在此处更详细地描述。

SEQ函数将索引，索引的名称，索引的起始，结束和增量值作为参数作为参数，并返回一个列表，该列表包含对索引的所有可能值的表达式的求值。函数的一般形式是SEQ（表达式，索引，开始，结束，增量）。

在以下示例中，在ALG模式下，我们识别 $\text{expression} = n^2$ ， $\text{index} = n$ ， $\text{start} = 1$ ， $\text{end} = 4$ 和 $\text{increment} = 1$ ：

```
:SEQ(n^2,n,1,4,1.)
{1. 4. 9. 16.}
```

┌───┴───┐
| SORT | SEQ | | | | LIST |

产生的列表对应于值 $\{1^2, 2^2, 3^2, 4^2\}$ 。在RPN模式下，您可以列出该函数的不同参数，如下所示：

```
5: 2.
4: n
3: 1.
2: 4.
1: 1.
```

┌───┴───┐
| SORT | SEQ | | | | LIST |

在应用函数SEQ之前。

The MAP function

通过命令目录(\rightarrow CAT), 提供的MAP函数将数字列表和函数f (X) 或形式 $\llcorner \rightarrow a \dots \gg$, 并生成一个列表，该列表包含将该函数或程序应用于数字列表。例如，以下对函数MAP的调用将函数SIN (X) 应用于列表{1,2,3}:

```
:MAP({1 2 3},SIN(X))
{SIN(1) SIN(2) SIN(3)}
```

┌───┴───┐
| CASCH | HELP | | | | |

In ALG mode, the syntax is:

```
(ALPHA) (ALPHA) (M) (A) (P) (ALPHA) (←) ( ) ( / ) (→) ( , ) ( 2 ) (→) ( , ) ( 3 ) (▶) (→)
( , ) ( SIN ) (ALPHA) ( X ) ( ENTER )
```

In RPN mode, the syntax is:



```
(←) ( ) ( / ) (→) ( , ) ( 2 ) (→) ( , ) ( 3 ) ( ENTER ) ( , ) ( SIN ) (ALPHA) ( X ) ( ENTER ) (ALPHA) (ALPHA) (M)
(A) (P) ( ENTER )
```

在这两种情况下，您都可以输入MAP命令（如上例所示）或从CAT菜单中选择命令。

以下对函数MAP的调用使用程序而不是函数作为第二个参数：

```
: MAP({0,1,2},{x → x 'x  
^2-1' *})  
(-1 0 3)  
CASCM/HELP
```

定义使用列表的函数

在第3章中，我们介绍了使用DEFINE函数 ( DEF) 来创建具有一个或多个参数的实数函数。使用DEF定义的函数也可以与列表参数一起使用，除了包含加法的任何函数必须使用ADD运算符而不是加号 ()。例如，如果我们定义函数 $F(X, Y) = (X-5) * (Y-2)$ ，这里以ALG模式显示：

```
: DEFINE('F(X,Y)=(X-5.)(Y-2)  
NOVAL  
F L2 L1
```

我们可以使用列表（例如，本章前面定义的变量L1和L2）来评估函数，从而产生：

```
: DEFINE('F(X,Y)=(X-5.)(Y-2)  
NOVAL  
: F(L1,L2)  
(20. 0. 2. -3.)  
F L2 L1
```

由于函数语句不包含任何添加，因此将函数应用于列表参数非常简单。但是，如果我们定义函数 $G(X, Y) = (X + 3) * Y$ ，尝试使用列表参数 (L1, L2) 评估此函数将失败：

```
: DEFINE('G(X,Y)=(X+3.)*Y')  
NOVAL  
G(L1,L2)  
G F L2 L1
```

```
: DE  
Invalid Dimension  
: G(L1,L2)  
"Invalid Dimension"  
G F L2 L1
```

为了解决这个问题，我们可以编辑变量的内容，我们可以使用   在堆栈中列出

```
:DEFINE('G(X,Y)=(X+3.)*Y')
NOVAL
:G(L1,L2)
"Invalid Dimension"
« → X Y '(X+3.)*Y' »
G | F | L2 | L1 |
```

用ADD替换加号 (+) :

```
:G(L1,L2)
"Invalid Dimension"
« → X Y '(X+3.)*Y' »
: « → X Y '(X ADD 3.)*Y' »
Y' »
« → X Y '(X ADD 3.)*Y' »
»
+SKIP+SKIP+ →DEL DEL→DEL L|Ins
```

接下来，我们将编辑的表达式存储到变量中 :

```
: « → X Y '(X ADD 3.)*Y' »
Y' »
« → X Y '(X ADD 3.)*Y' »
»
:ANS(1.)▶G
« → X Y '(X ADD 3.)*Y' »
»
G | F | L2 | L1 |
```

评估G (L1, L2) 现在产生以下结果:

```
:G(L1,L2)
{-12. 10. 6. 35.}
G | F | L2 | L1 |
```

As an alternative, you can define the function with ADD rather than the plus sign (+), from the start, i.e., use `DEFINE('G(X,Y)=(X ADD 3)*Y')` :

作为替代方案，您可以从一开始就使用ADD而不是加号 (+) 定义函数，

```
:DEFINE('G(X,Y)=(X ADD 3)*Y')
NOVAL
:G(L1,L2)
{-12. 10. 6. 35.}
G | F | L2 | L1 |
```

您还可以将函数定义为 $G(X, Y) = (X - 3) * Y$.

列表的应用

本节显示了几个列表的应用程序，用于计算样本的统计信息。通过示例，我们了解了值列表，例如{s1, s2, ..., sn}。假设感兴趣的样本是列表

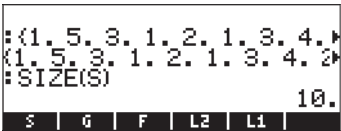
{1, 5, 3, 1, 2, 1, 3, 4, 2, 1}

是列表并且我们将它存储到一个名为S的变量中（下面的屏幕截图显示了ALG模式下的这个动作，但RPN模式下的过程非常相似。请记住，在RPN模式下在激活函数之前将函数的参数放在堆栈中）：



列表的调和平均值

这是一个足够小的样本，我们可以在屏幕上指望元素的数量（n = 10）。对于更大的列表，我们可以使用函数SIZE来获得该数字，例如，

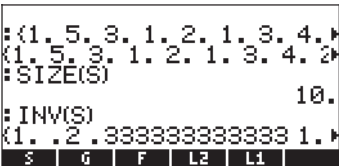


假设我们想要计算样本的调和平均值，定义为

$$s_h = \frac{1}{\frac{1}{n} \sum_{k=1}^n \frac{1}{s_n}} = \frac{1}{\frac{1}{n} \left(\frac{1}{s_1} + \frac{1}{s_2} + \dots + \frac{1}{s_n} \right)}$$

要计算此值，我们可以按照以下步骤操作：

- 1.将函数INV（）应用于列表S:



- 2.将函数ΣLIST（）应用于结果列表in1。

```

{1. 5. 3. 1. 2. 1. 3. 4. 2}
:SIZE(S)
10.
:INV(S)
{1. 2. 333333333333 1.}
:ΣLIST(ANS(1.))
6.11666666666
◀LIST|ΣLIST|TLIST|SORT|REVLI|ADD

```

3.将上面的结果除以n = 10:

```

:INV(S)
{1. 2. 333333333333 1.}
:ΣLIST(ANS(1.))
6.11666666666
:ANS(1.)
10.
.611666666666
S | G | F | L2 | L1 |

```

4.将INV () 函数应用于最新结果:

```

:ΣLIST(ANS(1.))
6.11666666666
:ANS(1.)
10.
.611666666666
:INV(ANS(1.))
1.6348773842
S | G | F | L2 | L1 |

```

因此，列表S的调和平均值为sh = 1.6348

列表的几何平均值

样本的几何平均值定义为

$$x_g = \sqrt[n]{\prod_{k=1}^n x_k} = \sqrt[n]{x_1 \cdot x_2 \cdots x_n}$$

要查找存储在S中的列表的几何平均值，我们可以使用以下过程:

1. 函数 TLIST() 应用于列表S:

```

:ANS(1.)
10.
.611666666666
:INV(ANS(1.))
1.6348773842
:TLIST(S)
720.
S | G | F | L2 | L1 |

```

2. 应用函数XROOT (x, y) , 即按键 $\boxed{\rightarrow} \boxed{\sqrt{x}}$, to the result in 1:


```

10.
: INV(ANS(1.)) .611666666666
: TLIST(S) 1.6348773842
: XROOT(ANS(1.),10) 720.
S G F L2 L1

```

```

.611666666666
: INV(ANS(1.)) 1.6348773842
: TLIST(S) 720.
: ANS(1.)√10. 1.00320315402
S G F L2 L1

```

因此，列表S的几何平均值是 $s_g = 1.003203 \dots$

加权平均

假设上面定义的列表S中的数据，即：

$$S = \{1, 5, 3, 1, 2, 1, 3, 4, 2, 1\}$$

由权重的影响，

$$W = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$$

如果我们将权重列表定义为 $W = \{w_1, w_2, \dots, w_n\}$ ，我们注意到上面列表W中的第k个元素可以由 $w_k = k$ 定义。因此，我们可以使用函数SEQ生成此列表，然后将其存储到变量中  as follows:

```

: ANS(1.)√10. 720.
: SEQ(k,k,1,10,1) 1.00320315402
(1. 2. 3. 4. 5. 6. 7. 8. 9)
: ANS(1.)√W
(1. 2. 3. 4. 5. 6. 7. 8. 9)
SORT| SEQ| | | LIST

```

给定数据列表 $\{s_1, s_2, \dots, s_n\}$ 和权重列表 $\{w_1, w_2, \dots, w_n\}$ ，S中数据的加权平均值定义为

$$s_w = \frac{\sum_{k=1}^n w_k \cdot s_k}{\sum_{k=1}^n w_k}$$

要使用列表W中的权重计算列表S中数据的加权平均值，我们可以使用以下步骤：

1. 乘以列表S和W:

```

1.00320315402
: SEQ(k,k,1,10,1)
(1. 2. 3. 4. 5. 6. 7. 8. 9)
: ANS(1.)√W
(1. 2. 3. 4. 5. 6. 7. 8. 9)
: S*W
(1. 10. 9. 4. 10. 6. 21. )
W S G F L2 L1

```

2. 在此结果中使用函数ΣLIST来计算 s_w 的分子:

```

(1. 2. 3. 4. 5. 6. 7. 8. 9)
:ANS(1.)▶W
(1. 2. 3. 4. 5. 6. 7. 8. 9)
:S.W
(1. 10. 9. 4. 10. 6. 21. ▶
:ΣLIST(ANS(1.))
121.
ΣLIST ΣLIST nLIST SORT REVL1 ADD

```

3. 再次使用函数 Σ LIST 来计算 s_w 的分母:

```

(1. 2. 3. 4. 5. 6. 7. 8. 9)
:S.W
(1. 10. 9. 4. 10. 6. 21. ▶
:ΣLIST(ANS(1.))
121.
:ΣLIST(W)
55.
W S G F L2 L1

```

4. 使用表达式 $\text{ANS}(2) / \text{ANS}(1)$ 计算加权平均值:

```

:ΣLIST(ANS(1.))
121.
:ΣLIST(W)
55.
:ANS(2.)
:ANS(1.)
2.2
W S G F L2 L1

```

因此，列表W中具有权重的列表S的加权平均值是 $s_w = 2.2$.

Note: ANS (1) 指的是最近的结果 (55)，而ANS (2) 指的是前一个结果 (121)。

分组数据统计

分组数据通常由表格给出，该表格显示数据类或数据库中数据的频率 (w)。每个类或bin由类标记表示，通常是类的中点。下面显示了分组数据的示例:

	Class	Frequency
Class	mark	count
boundaries	s_k	w_k
0 - 2	1	5
2 - 4	3	12
4 - 6	5	18
6 - 8	7	1
8 - 10	9	3

类标记数据可以存储在变量S中，而频率计数可以存储在变量W中，如下所示：

```
:SEQ(2-k-1,k,1,5,1)
                                (1 3 5 7 9)
:ANS(1)►S
                                (1 3 5 7 9)
:(5 12 18 1 3)►W
                                (5 12 18 1 3)
W | S | G | F | L2 | L1
```

Given the list of class marks $S = \{s_1, s_2, \dots, s_n\}$, and the list of frequency counts $W = \{w_1, w_2, \dots, w_n\}$, the weighted average of the data in S with weights W represents the mean value of the grouped data, that we call \bar{s} , in this context:

给定类别标记S = {s1, s2, ..., sn}的列表，以及频率计数列表W = {w1, w2, ..., wn}，S中具有权重W的数据的加权平均值表示平均值 在这种情况下，分组数据的值，我们称之为S

$$\bar{s} = \frac{\sum_{k=1}^n w_k \cdot s_k}{\sum_{k=1}^n w_k} = \frac{\sum_{k=1}^n w_k \cdot s_k}{N}$$

where $N = \sum_{k=1}^n w_k$ 表示总频率计数。

因此，列表S和W中的数据平均值可以使用上面针对加权平均所概述的过程来计算，即，

```
:ΣLIST(W*S)
ΣLIST(W)
                                55
                                13
:→NUM(ANS(1))
                                4.23076923077
W | S | G | F | L2 | L1
```

我们将此值存储到名为XBAR的变量中：

```
ΣLIST(W)
                                55
                                13
:→NUM(ANS(1))
                                4.23076923077
:ANS(1)►XBAR
                                4.23076923077
XBAR | W | S | G | F | L2
```

该分组数据的方差定义为

$$V = \frac{\sum_{k=1}^n w_k \cdot (s_k - \bar{s})^2}{\sum_{k=1}^n w_k} = \frac{\sum_{k=1}^n w_k \cdot (s_k - \bar{s})^2}{N}$$

要计算最后的结果，我们可以使用以下内容：

```

: ANS(1) ► XBAR
4.23076923077
: ΣLIST(W(S-XBAR)2)
156.923076923
: ΣLIST(W)
39
XBAR | W | S | G | F | L2

```

```

: ANS(1) ► XBAR
4.23076923077
: ΣLIST(W(S-XBAR)2)
156.923076923
: ΣLIST(W)
39
XBAR | W | S | G | F | L2

```

分组数据的标准差是方差的平方根：

```

: ΣLIST(W)
39
: ANS(2)
: ANS(1)
4.02366863905
: √ANS(1)
2.00590843237
XBAR | W | S | G | F | L2

```