

# Chapter 9

## Lists and Sequences

### Creating Lists

#### To enter a list from the keyboard

1. Press  $\{\}$ . The braces indicate the beginning and end of a list.
2. Enter the elements of the list, separating each with a comma ( $\{ \}$ ).
3. Press  $\text{ENTER}$ .

Note that the list is displayed without commas.

#### To assemble a list from a set of stack objects

If you are working in RPN mode, you can assemble a list from a contiguous set of objects already on the stack.

1. Enter the objects on the stack (or place them at the bottom of the stack: level 1, level 2, and so on).
2. Enter the number of objects you want in the list on the first level of the stack.

3. Execute the  $\rightarrow\text{LIST}$  command to convert the objects to a list.

The  $\text{TO LIST}$  command can be executed by pressing  $\leftarrow \text{PRG LIST} \rightarrow \text{LIST}$ .

#### To append a new object to the beginning of a list

1. Enter or select the new object.
2. Press  $\oplus$ .
3. Enter or select the list.
4. Press  $\text{ENTER}$ .

**In RPN mode:** follow steps 1, 3, and 2.

## To append a new object to the end of a list

1. Enter or select the list.
2. Press  $\oplus$ .
3. Enter or select the new object.
4. Press  $\text{ENTER}$ .

**In RPN mode:** follow steps 1, 3, and 2.

## List Processing

### To apply a one-argument command to each element in a list

The order in which you enter the command and the list depends on the type of function it is: *prefix* or *postfix*.

A **prefix function** is one whose name or abbreviation usually comes before its arguments; for example,  $\text{SIN}(x)$  and  $\text{SQ}(x, y)$ .

A **postfix function** is one whose name or abbreviation usually comes after its arguments; for example,  $6!$ .

The following example shows how to apply a one-argument prefix command to a list if you are working in algebraic mode. The table below the example gives the order of the steps for postfix functions, and for when you are working in RPN mode.

1. Enter or select the command.

For example,  $\text{SQ}()$ .

2. Enter or select the list.

For example,  $\text{SQ}(\{3, 4, 5\})$ .

3. Press  $\text{ENTER}$ .

The answer is a new list, with each element the square of each element in the original list:  $\{9, 16, 25\}$ .

If you are executing a postfix function, or working in RPN mode, the steps you should take are given in the following table.

	Prefix	Postfix
Algebraic	1, 2, 3	2, 1, 3
RPN	2, 3, 1	2, 1

**Another example:** To find the factorial of 3, 4, and 5 while working in algebraic mode.

1. Enter or select the list: {3, 4, 5}.
2. Select the factorial command:  $\leftarrow$  (MTH) PROBABILITY !
3. Press (ENTER).

The answer is {6, 24, 120}.

## To add corresponding elements of two lists

1. Enter or select the first list.  
For example, {4, 7, 9}.
2. Select the ADD command.  
 $\leftarrow$  (MTH) LIST ADD
3. Enter the second list.  
For example, {2, -4, -8}.  
The number of elements in the second list must be the same as the number of elements in the first list.
4. Press (ENTER).  
The answer is {6, 3, 1}.

**In RPN mode:** follow steps 1, 3, and 2.

## To concatenate two lists

To concatenate two lists is to form a list made up of the elements of both lists. The order of the elements in each sub-list is preserved.

1. Enter the list whose elements will form the first part of the concatenated list.

For example, {1, 2, 3}

2. Press  $\oplus$ .

3. Enter the list whose elements will form the latter part of the concatenated list.

For example, {4, 5, 6}.

4. Press  $\text{ENTER}$ .

The answer is {1, 2, 3, 4, 5, 6}.

**In RPN mode:** follow steps 1, 3, and 2.

## To subtract, multiply, or divide corresponding elements of two lists

1. Enter the first list.

2. Press  $\ominus$ ,  $\otimes$ , or  $\div$ .

3. Enter the second list.

The number of elements in the second list must equal the number of elements in the first list.

4. Press  $\text{ENTER}$ .

**In RPN mode:** follow steps 1, 3, and 2.

For example, {2, 4, 7}  $\otimes$  {3, 5, -4} yields {6, 20, -28}.

# Applying a function or program to a list


The DOLIST command enables you to run programs or execute functions on *groups* of lists.

## To run a program or execute a function on lists

When you are operating on a number of lists—especially long lists—it may be easier and clearer if you are in RPN mode (as in the case of the following example).

1. Enter the lists.
2. Enter the number of lists to be operated on. This number is the number of elements in each list operated on in each iteration of the function or program.
3. Enter a program or function.
4. Execute DOLIST.

**Example:** Apply the function defined by  $a_n + (b_n \times c_n)$  to three lists,  $a$ ,  $b$ , and  $c$ .

1. Enter the lists.  
For example, {1, 2, 3, 4}, {4, 5, 6, 7}, and {7, 8, 9, 11}.
2. Enter the number of lists to be operated on (for example, 3).
3. Enter the function.  
« \* + »
4. Execute the DOLIST command.  
 PRG LIST PROCEDURES DOLIST  
The answer is {29, 42, 57, 81}.

## To apply a procedure sequentially to elements of a list

The DOSUBS command enables you to apply a program or function to the elements in the list one after the other. In RPN mode, the procedure is:

1. Enter the list.
2. Enter the frame index. This is the number of elements affected by each iteration of the function. For example, entering 3 would take 3 elements from the list and use them as arguments for the function.
3. Enter a program or function.
4. Execute DOSUBS.

**Example:** Find the 2-element moving average of {2, 4, 8, 16, 50}.

1. Enter the list.
2. Enter the frame index.

In this example, the number is 2, as you want to find the average of *two* numbers each time.

3. Enter the program.

« + 2 / »

4. Execute the DOSUB command.

⏮ (PRG) LIST PROCEDURES DOSUBS.

The answer is {3, 6, 12, 33}.

## To execute a function on every element of a list

The STREAM command enables you to apply a function recursively to every element in a list. The command takes the first two elements, executes a function, takes the result and the next element, executes the function again, and so on, until there are no more arguments for the function to work on.

In RPN mode, the steps are:

1. Enter the list.

For example, { 1 2 3 4 5 }.

2. Enter a program or function that takes two arguments and returns one result.

For example, « \* ».

3. Execute STREAM.










⏮ (PRG) LIST PROCEDURES STREAM.





In this example, the function multiplies all the elements in the list. The answer is 120.

You can also use the product list command to calculate the product of all the elements in a list. See “To find the product of the elements in a finite list” on page 9-10.

# List Manipulations

The following functions provide ways to manipulate the elements of a list:

Function	Description
 (MTH) LIST SORT	Sorts the elements in a list in ascending order. In RPN mode, the list must be on level 1.
 (MTH) LIST REVLIST	Reverses the order of the elements in a list. In RPN mode, the list must be on level 1.
 (PRG) LIST ELEMENTS HEAD	Returns the first element in the list. In RPN mode, the list must be on level 1.
 (PRG) LIST ELEMENTS TAIL	Returns a list of all the elements in the list except the first element. In RPN mode, the list must be on level 1.
 (PRG) LIST ELEMENTS GET	Returns the element in the list (argument 1/level 2) corresponding to a specified position (argument 2/level 1). For example, GET({1, 4, 7, 8}, 3) returns 7, since 7 is the third element in the list.
 (PRG) LIST ELEMENTS GETI	Similar to the GET command (see above) but also returns the specified position number incremented by 1 (and the original list). For example, GETI({1, 4, 7, 8}, 3) returns {1, 4, 7, 8}, 4 (the specified position number + 1), and 7 (the third element in the list).
 (PRG) LIST ELEMENTS PUT	Replaces an element at a particular position (argument 2/level 2) of a list (argument 1/level 3), with a new element (argument 3/level 1). For example, PUT({1, 2, 3}, 2, 5) returns {1, 5, 3}.
 (PRG) LIST ELEMENTS PUTI	Similar to the PUT command (see above) but also returns the specified position number incremented by 1. For example, PUTI({1, 2, 3}, 2, 5) returns {{1, 5, 3}, 3}.
 (PRG) LIST ELEMENTS SIZE	Returns the number of elements in a list. In RPN mode, the list must be on level 1.

Function (Continued)	Description
 (PRG) LIST ELEMENTS POS	<p>Returns the position of the first occurrence of an element (argument 2/level 1) in a specified list (argument 1/level 2). For example:</p> <p>POS({2, 4, 6, 1, 2, 3, 4}, 4) returns 2 (since the first occurrence of 4 is at position 2 in the list).</p>
 (PRG) LIST OBJ→	<p>RPN command to disassemble a list into its elements and return (to level 1) the number of elements that were in the list. Each element is placed on a separate level of the stack.</p>
 (PRG) LIST SUB	<p>Creates a sub-list of elements from a list (argument 1/level 3) starting with the element at one position (argument 2/level 2) and ending with the element at another position (argument 3/level 1). For example, SUB({1, 2, 3, 4, 5}, 2, 4) returns {2, 3, 4}.</p>
 (PRG) LIST REPL	<p>Replaces the elements of a list (argument 1/level 3) with the elements from another list (argument 3/level 1) starting at a specified position (argument 2/level 2). For example, REPL({6, 7, 8, 2}, 2, {1, 2, 3}) returns {6, 1, 2, 3 }.</p>

# Sequences

Sequence commands automate the generation of a list from the repeated execution of a function or program.

## To generate a sequence

In algebraic mode:

1. Specify the sequential calculation command.

$\leftarrow$  (PRG) LIST PROCEDURES SEQ.

2. Enter the function or program (or its name).
3. Enter the index variable name.
4. Enter the initial value for the variable.
5. Enter the final value for the variable.
6. Enter the step size of the increment.

The number of elements generated is the integer portion of  $((final - initial)/step) + 1$ .

7. Press (ENTER).

**In RPN mode:** follow steps 2, 3, 4, 5, 6, and 1.

**Example:** Generate a list of the squares of the integers from 15 to 19.

1. Press  $\leftarrow$  (PRG) LIST PROCEDURES SEQ.

2. Enter  $\leftarrow$  ( )  $\leftarrow$  (X<sup>2</sup>) X  $\rightarrow$   $\rightarrow$  and press  $\leftarrow$  ( ) .

Note that tick marks are needed to delimit the algebraic object.

3. Enter X and press  $\leftarrow$  ( ) .

$x$  is the variable that will be evaluated at each iteration.

4. Enter 15 and press  $\leftarrow$  ( ) .

15 is the initial value, that is, the value used in the first iteration.

5. Enter 19 and press  $\leftarrow$  ( ) .

19 is the final value, that is, the value used in the last iteration.

6. Enter 1.

1 is the step value, that is, by how much the initial value will be incremented after each iteration.

7. Press (ENTER).

The result is {225, 256, 289, 324, 361}.

If you had entered 2 for the step value at step 6, then every second integer in the specified range would have been included in the iterations, and the result would have been {225, 289, 361}.

## To find the sum of the elements in a finite list

1. Select the list sum command.

$\leftarrow$  (MTH) LIST SLIST

2. Enter or select the list.

For example, {2, 4, 7}.

3. Press (ENTER).

The answer is 13.

**In RPN mode:** follow steps 2 and 1.

## To find the product of the elements in a finite list

1. Select the list product command.

$\leftarrow$  (MTH) LIST PLIST

2. Enter or select the list.

For example, {2, 3, 4}.

3. Press (ENTER).

The answer is 24.

**In RPN mode:** follow steps 2 and 1.

## To find the set of first differences of a finite sequence

The first differences for the list  $\{x_1 \ x_2 \ \dots \ x_n\}$  is defined as  $\{x_2 - x_1 \ \dots \ x_n - x_{n-1}\}$ .

1. Select the list differences command.

$\leftarrow$  (MTH) LIST DLIST

2. Enter or select the list.

For example, {2, 5, 11, 13, 33}.

3. Press (ENTER).

The answer is {3, 6, 2, 20}.

**In RPN mode:** follow steps 2 and 1.