MASSACHUSETTS INSTITUTE OF TECHNOLOGY

ARTIFICIAL INTELLIGENCE LABORATORY

December 1974

A.I. Memo 315

LOGO Memo 14

A GLOSSARY OF LOGO PRIMITIVES

Hal Abelson
Jim Adams

ABSTRACT

This is a brief description of the primitives in PDP 11 LOGO. It is intended to provide a quick reference for users who are already familiar with LOGO basics. For a more detailed and comprehensive description of LOGO, consult the LOGO Manual (A.I. Memo 313, LOGO Memo 7).

This work was supported by the National Science Foundation under grant number GJ-1049 and conducted at the Artificial Intelligence Laboratory, a Massachusetts Institute of Technology research program supported in part by the Advanced Research Projects Agency of the Department of Defense and monitored by the Office of Naval Research under Contract Number N00014-70-A-0362-0002.

A Glossary of LOGO Primitives

This is a brief description of the primitives in PDP 11 LOGO. It is intended to provide a quick reference for users who are already familar with LOGO basics. For a more detailed and comprehensive description of LOGO, consult the LOGO Manual (LOGO MEMO No. 7).

LOGO primitives are generally divided into operations (which output) and commands (which don't). There are also special modifying words, which are only used in combination with certain other primitives, and noise words. Noise words are optional and generally do not affect the evaluation of the LOGO expression in which they appear (as long as they are in the right place!), but can be used to make the syntax of the expression more like English.

ALL

Special modifying word. Allows the user to reference all existing items of a given type with a single command. See, for example, ERASE ALL and PRINTOUT ALL.

ALSO

Takes one input, the name of a device. This command allows the user to control more than one device at a time on LOGO. For example, in order to use both a display and a floor turtle, type

```
CLEARSCREEN
ALSO TURTLE <number>
```

The CNTRL command is used to specify which device the following commands will refer to (if this is not used, the default value is the first device that you "owned").

AND

Noise word. Used to separate inputs to a procedure as in:

```
SUM OF 3 AND 4
```

ASIZE

Takes one input, the name of an array, and outputs the dimensions of the array.

BACK (BK)

Takes one input. Commands the display or floor turtle to move backwards the number of units designated by the input.

BELL

Takes no inputs. It rings the bell on the console.

BOTH

Takes two inputs and outputs TRUE if both inputs are true and FALSE if one or both are false. Each input must be a conditional (evaluate to either TRUE or FALSE).

BTOUCH

Takes no inputs. It checks the back touch sensor on the floor turtle, and outputs TRUE if it has been activated, FALSE otherwise.

BURY

Takes one input, a procedure name, and effectively "hides" that procedure in the workspace. For example, the buried procedure "FOO" will not be printed in PO ALL commands, nor will it be affected by any procedure commands not explicitly directed to it. This protects it from accidental deletion through ERASE ALL. To reference a buried procedure, the name of the procedure must be specified explicitly in the command, as in POF "FOO or ERASE "FOO.

To unbury a procedure, ERASE BURY <procedure> is used. BURY also accepts ALL as an input, in which case all the procedures in the workspace will be buried.

BUTFIRST (BF)

Takes one input. If the input is a list, outputs all but the first word of the list. If the input is a word, outputs all but the first character of the word.

```
BUTFIRST "JOHN outputs OHN
BUTFIRST [JOHN MARY PAUL] outputs [MARY PAUL]
```

BUTLAST (BL)

Takes one input, a word or list. If a word, outputs all but the last letter of the word. If a list, outputs all but the last word of the list.

```
BUTLAST "HARRY outputs HARR
BUTLAST [THE NEW BICYCLE] outputs [THE NEW]
```

.CASESW

Takes no inputs. If you are using a terminal with lower case characters, LOGO normally translates lower case into upper case. .CASESW switches this feature off and on.

CLEARSCREEN (CS)

Takes no inputs. Commands the computer to clear the display screen. It also returns the display turtle to its original position at the center of the screen [0,0], and pointing straight up (HEADING 0).

CLOCK

Takes no inputs. Outputs a number which is incremented approximately every 1/60th of a second.

.CLOSE

Takes one input, a device specification (see .TYO), and releases that device from the curent user's control, so that another user may access it.

.CLOSEF

Takes no inputs. Closes the file currently open (no more than one file can be open at a time). If no files are open, the command is ignored. See .OPENR and .OPENW.

CNTRL

When a user is contolling more than one device, the CNTRL command specifies the device to which a given command is directed. Takes one input, the name of a device the user currently "owns", and specifies that device as the object of the next command typed. All commands will be directed to this device until another CNTRL command is typed. See also ALSO.

CONTENTS

No inputs. Outputs a list containing the titles of all procedures in the workspace.

CONTINUE (CO)

Takes no inputs. Continues the execution of a procedure that is currently PAUSEd, starting at the next line. (See PAUSE).

COS

Takes one numeric input, and returns the cosine of the angle of <input> degrees.

COUNT

Takes one input, a word or a list. If the input is a list, outputs the number of words in the list. If it is a word, outputs the number of characters in the word.

```
COUNT "ELEPHANT outputs 8
COUNT [ L E PHANT ] outputs 3
```

CRINDEX

CRINDEX is used to create an index in the filing system (see INDEX). It takes one argument, the name of the index to be created. This index is added onto the branch of the filing tree specified by the previous USE command. If the input is a list, however, the index may be even farther down the tree. See SETINDEX.

.CTYI

Takes no inputs. Outputs the character code of the next character typed in at the user's console.

.CTYO

Takes one numeric input which specifies a character code. This character is printed as output on the user's console.

CTYOWAIT

Used to cause LOGO to WAIT a until a console has finished typing out. Normally takes no inputs, in which case the pause ends as soon as the console has finished its output. If an input is given, this is an additional amount of time for LOGO to wait after the console has finished.

DATE

Takes no inputs. Outputs a list of 3 elements which is MO/DAY/YEAR. Currently unimplemented.

DEBUG

Takes no inputs. Switches the debugging system on and off. When the system is on, errors cause a PAUSE rather than a BREAK to the top-level. During this pause, the user may execute any LOGO commands and then continue execution by using the CONTINUE command.

DEFINEARRAY (DEFAR)

This command is used to create an array. Depending on the number of dimensions of the array, DEFINEARRAY takes up to five inputs, in this order: the array name, the dimensions of the array, and the array type (0 for integer, 1 for floating point, 2 for pointer). Up to three dimensions are allowed. If the array has more than one dimension, DEFINEARRAY must be treated as having a variable number of inputs, i.e. the command and its inputs must be enclosed in parentheses.

DIFFERENCE

Takes two inputs, which must be numbers, and outputs their difference (subtracts the second from the first).

```
DIFFERENCE 3 1 outputs 2
```

DISPLAY

Takes one input which must be a SNAP (i.e., something created by the SNAP operation) and shows it on the screen at a location determined by the current position of the turtle. The SNAP always appears in the orientation in which it was originally drawn.

.ECHOSW

Takes no inputs. Turns off and on a feature which inhibits the echoing of characters typed at the console.

EDIT (ED)

Takes one input, the name of a procedure, and puts the user in editing mode. Allows the user to change the definition of the specified procedure. When done editing, type END to get out of editing mode.

EDIT LINE (EDL)

Takes one input, the line number of a procedure. Tells the computer which line in a procedure you wish to edit. The specified line is put into the edit buffer where it can be manipulated using the special control characters for editing (see EDITING CHARACTERS near the end of this glossary). EDIT LINE can be used only when defining or editing a procedure.

EDIT TITLE (EDT)

Takes no inputs, but puts the title of the procedure currently being defined/edited into the edit buffer, where it can be changed using the editing control characters. Or the user can simply redefine the title. For example,

```
>EDIT TITLE
>TO JOHN
```

makes JOHN the title of the procedure no matter what it was before. EDIT TITLE can only be used when editing or defining a procedure.

EITHER

Takes two inputs and outputs TRUE if at least one is true and FALSE if both are false. Both inputs must be conditionals (evaluate to either TRUE or FALSE).

ELSE

Can be used with IF and THEN to allow an alternative course of action to take place if the conditional in an IF-THEN pair is FALSE.

```
IF :X>5 THEN PRINT "GOOD ELSE PRINT "BAD
```

EMPTYP

Takes one argument. Outputs TRUE if the argument is the empty word or the empty list, FALSE otherwise.

END

Takes no inputs. Tells the computer that you are finished defining or editing a procedure.

EQUAL

Takes two inputs and outputs TRUE if both arguments evaluate to be the same thing. Otherwise outputs FALSE.

```
EQUAL "JOHN "JOHN   outputs TRUE
EQUAL 2 1+1  outputs TRUE
EQUAL 1 3  outputs FALSE
```

ERASE (ER)

Takes one input, the name of a procedure, and erases that procedure from the workspace.

ERASE ALL (ER ALL)

Clears out the entire workspace.

ERASE ARRAY (ER ARRAY)

Takes one input, the name of an array, and erases it from the workspace.

ERASE ARRAYS (ER ARRAYS)

Takes no inputs. Erases all arrays in the workspace (arrays cannot be saved in the filing system).

ERASE BURY (ER BURY)

Takes one input, the name of a buried procedure (or ALL) and unburies it (or all buried procedures). See BURY.

ERASE FILE (ER FILE)
Takes one input, the name of a file, and removes it from the file system.

ERASE LINE (ERL)
Takes one input, a line number, and gets rid of that line. Can only be used while defining or editing.

ERASE NAME (ER NAME)
Takes one input, a name, and erases that name from the workspace.

ERASE NAMES (ER NAMES)
Erases all defined names.

ERASE PROCEDURES (ER PROCEDURES)
Gets rid of all procedures (but leaves the names.)

ERASE STEP (ER STEP)
Takes one input, a procedure name (or ALL) and causes that procedure (or all STEPped procedures) to no longer be traced (see STEP).

ERASE TRACE (ER TRACE)
Takes one input, a procedure name (or ALL), and causes that procedure (or all TRACEd procedures) to no longer be traced (see TRACE).

ERBRK
Takes no inputs. Outputs 1 if the previous interruption of the procedure being executed was caused by pressing cntrl-G, -1 if it was caused by pressing cntrl-Z, 0 otherwise.

ERCLR
Takes no inputs. Deactivates the ERSET command. See ERSET.

ERLIN
Takes no inputs. Outputs the procedure line number in which the last error occurred.

ERLOC
Takes no inputs. Outputs the location in the computer's core at which the last error occurred.

ERNAM
Takes no inputs, outputs the name of the last error.

ERNUM
Takes no inputs, outputs the number of the last error.

ERPRO
Takes no inputs, outputs the name of the procedure in which the last error occurred.

ERRET
Takes one input, a line number, and returns execution to that line of the procedure in which the last error occurred.

ERSET
Takes one input, a procedure name. Causes the procedure given as input to be executed every time an error occurs. If this procedure does not output, LOGO prints the normal error message after the execution is completed. If the procedure does output, the output is printed instead of the normal error message. System bug errors cannot be overidden by an ERSET.

ERTOK
Takes no inputs, outputs the "token number" at which the previous error occurred in the given line. Not very useful unless the user is acquainted with the LOGO evaluator.

.FILEP
Takes as input a list which contains the desired information to be written into a file. .FILEP can be used only when a file has been opened for writing through the use of .OPENR or .OPENA commands. The input to .FILEP can also be a procedure name, in which case the procedure is executed and its output is written into the file. If there is no output, nothing is written.

.FILER
Takes no inputs, returns as output the current line of the file being read. .FILER can be used only when a file has been opened for reading through the use of .OPENR. Each time .FILER is used, an internal pointer in the file is incremented, so that the next use of .FILER will read the next line in the file, and so on. When the end of the file is reached, one blank line will be printed out by .FILER, and then the file is automatically closed. If the line being read is a procedure name, that procedure will be executed.

FILE
Special word. See PRINTOUT. For information on how to create a file, see WRITE.

FIRST (F)
Takes one input, a word or list. It outputs the first character of a word or the first word of a list.

```
PRINT FIRST "HELLO outputs H
```

FORWARD (FD)
Takes one input. Tells the floor or display turtle to move ahead

the distance designated in the input.

```
            FORWARD 10 moves the turtle ahead 10.
```

FPRINT

Like PRINT except that it also prints the top-level brackets around a list.

```
            ?PRINT [HELLO THERE]
            HELLO THERE
            ?FPRINT [HELLO THERE]
            [HELLO THERE]
```

FPUT

Takes 2 arguments, the second of which must be a list (The first may be either a word or a list). Outputs a new list whose FIRST is the first arg to FPUT and whose BUTFIRST is the second arg to FPUT.

```
            FPUT "HERE [I AM] outputs [HERE I AM] See also LPUT.
```

FTOUCH

Takes no inputs. It outputs TRUE or FALSE depending on whether the turtle has activated its forward sensor.

GET

Used to reference a specific element of an array. Takes up to 4 inputs, depending on the dimensions of the array. These inputs are: the array name, and the coordinates of the desired element. (Coordinates are indexed using 0-origin, i.e., the first coordinate in any dimension is always 0, and the nth element of a dimension has coordinate n-1. GET outputs the value of the specified element of the array. As in DEFINEARRAY, if the array has more than one dimension, GET and its inputs must be enclosed in parentheses.

GO

Takes one input, the number of a line in a procedure. It is used in a procedure to transfer control to that line.

GOODBYE

Takes no inputs, and is effectively the same as HELLO, although LOGO responds with a different message. Results in a re-initialization of the user's workspace.

GREATER

Takes two numeric inputs and outputs TRUE if the first argument is greater than the second; if this is not so it outputs FALSE.

```
            GREATER 4 2 outputs TRUE
```

.GUN

Takes one arg which is a user number and resets that user's device. Its use without that user's consent is deemed highly anti-social.

HEADING

Takes no inputs. It asks the computer to output the heading of the display turtle, i.e., what direction it is pointing in (in degrees).

HELLO

Takes no inputs. HELLO tells the computer to erase everything in the workspace of the console you are using. HELLO should be typed before you begin writing procedures to clear out any garbage that is left in the workspace from the last person who used the console.

HERE

Outputs a list of 3 elements, the XCOR, YCOR and HEADING of the display turtle.

HIDETURTLE (HT)

Takes no inputs. Tells the computer to get rid of the little triangle which is the turtle on your display screen. The turtle will still leave a track even though you cannot see him. If you don't want to see the turtle track, type PENUP (see below).

HOME

Takes no inputs. Equivalent to SETT [0 0 0].

IF

Takes one argument which must evaluate to TRUE or FALSE and causes execution of the rest of the LOGO line to be conditional upon the evaluation of the input. If the input evaluates to TRUE, the rest of the line is executed, otherwise control passes directly to the next linbe in the procedure, and the remaining part of the line is ignored.

```
IF :N=7 THEN PRINT "HELLO
```

THEN is optional. See also ELSE.

IFFALSE (IFF)

Takes no inputs. Executes the rest of the line if the result of the previous TEST was FALSE.

IFTRUE (IFT)

Executes the rest of the line if the result of the previous TEST was TRUE.

ILINE

Takes no inputs, outputs the last line typed in at the console.

INDEX

Special word which refers to a part of the file system structure. All files belonging to a single user are said to be grouped under that user's index. Indices can occur at multiple levels within the user's main index. See CRINDEX and PRINTOUT INDEX.

INTEGER

Takes one numeric input, a floating point number, in decimal or exponential form, and outputs this number converted to an integer.

IS

Synonym for EQUAL.

LAMPOFF

Takes no inputs. Turns the light on the floor turtle off.

LAMPON

Takes no inputs. Turns the light on the floor turtle on.

LAST

Takes one input, a word or list. It outputs the last word of a list or the last character of a word.

```
            LAST [DOG AND CAT] outputs CAT
```

LEFT (LT)

Takes one numeric input. Causes the floor or display turtle to rotate to the left the number of degrees given as input.

LESS

Takes two numeric inputs. It outputs TRUE if the first argument is less than the second argument, FALSE otherwise.

LEVEL

Takes no inputs. Outputs a number which tells "how many procedures deep" current execution is. For example,

```
            ?PRINT LEVEL
            0
            >TO WHAM
            >10 PRINT LEVEL
            >20 WHAM
            >END
            WHAM DEFINED
            ?WHAM
            1
            2
```

etc. (Note that this procedure will not stop by itself).

LIGHT

Takes no inputs. Outputs the amount of light perceived by the "eye" of the floor turtle. This is a number between 0 and 63, inclusive.

LINE

Special word referring to a line in a procedure. See ERASE, PRINTOUT and EDIT.

LIST

Takes two inputs, each of which may be either a word or a list. Outputs a two-element LIST whose elements are its inputs. This operation also accepts a variable number of inputs (see below).

LISTP

Takes one input. Outputs TRUE if the input is a list, FALSE otherwise.

LOCAL

Takes one input, which is a name. Used in a procedure it causes the scope of that name to be local to the procedure (i.e. the name is undefined outside of the procedure).

LOGIN

Takes one input, a user name (quoted). This "tells the system who you are" and is used by the PEEK command. Also performs an automatic USE. In addition, searches the user's file system for an INIT file and performs all commands written into this file (if it exists), then searches for the MAIL file and prints it out, if the user so desires.

LPUT

Takes 2 arguments, the second of which must be a list. Outputs a list whose LAST is the first arg to LPUT and whose BUTLAST is the 2nd arg to LPUT:

```
LPUT "HERE [I AM] outputs [I AM HERE]
```

LTOUCH

Takes no inputs. Outputs TRUE if the floor turtle's left touch sensor has been activated, otherwise FALSE.

MAIL

Takes one input, the user name of the person to whom the mail is to be sent. The computer responds with a back-arrow (←) which indicates that anything typed in at the console is regarded as mail. To end the message, type a line with a single period followed by a carriage return. The completed message is then placed in the filing system of the user to whom it is sent, contained in a file named MAIL.

MAKE

Takes two inputs. The first input is the NAME, the second is the THING. MAKE assigns the NAME to the THING.

MCLEAR

Music box command. Takes no inputs, clears out the music box buffer.

MLEN

Music box command. Takes no inputs, outputs the length of the music compiled for the longest voice. See also VLEN.

MUCTRL

Music box command. Takes one input which specifies to the music box hardware how many voices you wish to load. This is similar to the NVOICES command in the standard music system. This is used in conjunction with MUTYO to generate real-time music.

MUTYO

Music box command for real-time music. Enables the user to bypass the music buffer and the PM command. Takes two inputs, each a music box pitch, and makes the music box play the pitches. Takes the same pitches as NOTE.

NAMES

Special word. A NAME is a LOGO word which is associated with a value, or THING. The use of the special word NAMES allows the user to access all names in the workspace with a single command. See PRINTOUT and ERASE.

NEWSNAP

Takes no inputs. Causes the image currently on the screen not to be part of subsequent snaps. Also sets the starting location of subsequent snaps to the current position of the turtle rather than (0,0). See also DISPLAY and SNAP.

NODISPLAY

Takes no inputs and turns off the display.

NOMUSIC

Music box command. Takes no inputs, releases the music box so that others may use it.

NOPLOTTER

Takes no inputs and turns off the plotter.

NOT

Takes one input, which must be a conditional. Outputs TRUE if the input is FALSE and FALSE if the input is TRUE.

NOTE

Music box command. Takes two numeric inputs, the first specifies the pitch and the second the duration of one note of music. Pitches are numbered chromatically from -24 to 36 with 0 being middle C. There are also three special "pitches":

-28 is a silence

-27,-26 are the percussion sounds "boom" and "ssh"

-25 is not a valid pitch. Durations must be between 0 and 127 units. Each unit is normally about 1/8 second. If the duration is zero, NOTE generates nothing. If it is 1, generates a pitch 1 unit long. If greater than 1, generates a pitch <duration-1> units long followed by 1 unit of rest (so that music will not sound "slurred"). If the pitch is -26 or -27, NOTE generates a sound for 1 unit followed by <duration-1> units of rest.

NOTURTLE
Takes no input. NOTURTLE tells the computer that you are finished with the floor turtle. You must use this command to release the turtle before someone else can use it.

NOWRAP
Takes no inputs. Tells the computer not to allow scenes on the display to go out of bounds. See WRAP.

NUMBERP
Takes 1 input. Outputs TRUE if the input is a number and FALSE otherwise.

NVOICES
Takes one input, a number which specifies the voices to which music output is sent. The music system normally multiplexes output among four voices, but it is also possible to send output to only one or two voices, using NVOICES:

NVOICES 1 outputs only to voice 1
NVOICES 2 outputs only to voices 1 and 2
NVOICES 4 outputs to all four voices (the normal mode) Since the music box is fed at a constant rate, NVOICES 1 causes the basic unit of duration to be 1/4 as long as normal, while NVOICES 2 causes it to be 1/2 as long.

OF
Noise word. Used to separate inputs from commands or operations, as in:

```
SUM OF 3 4
```

.OPENA
Takes one input, the name of a file (quoted), and opens the file for writing. The actual writing is done with the .FILEP command, and all information written is added (appended) onto the end of the file. None of the original contents of the file are altered. If the file given as input does not exist, a new file with the specified name will be created, in which case the command is identical to .OPENW.

.OPENR
Takes one input, a file name (quoted) and opens that file for reading. The actual reading is done with .FILER. Once the file has been automatically closed by .FILER, .OPENR must be typed in order to read the file again.

.OPENW
Similar to .OPENA, but the writing done with .FILEP is written starting at the beginning of the file. The file name which is given as input may be a non-existent file, in which case the file is created. If

the file already exists, however, the user will be asked to delete it before writing, since the old information contained in the file will be destroyed by the input to .FILEP.

OUTPUT

Takes one input. Can only be used in a procedure; returns control to the calling procedure and outputs the specified argument.

PAUSE

Takes no inputs. When the DEBUG switch is on, executing this command within a procedure will cause LOGO to pause. The user may then execute any commands he wishes. To resume the execution of the procedure, use the CONTINUE command.

PEEK

Takes no inputs. Prints system status information.

PENDOWN (PD)

Takes no inputs. When used for either a floor turtle or a display turtle, it causes the turtle to draw a line when it moves.

PENP

Takes no inputs. Used with the display turtle, it outputs TRUE if the pen is down, FALSE if the pen is up.

PENUP (PU)

Takes no inputs. When used with either a floor turtle or display turtle, causes the turtle to not draw a line when it moves.

PLOTTER

Takes no inputs, turns on the plotter. All other plotter commands are the same as display commands. However, certain display cokmmands do not apply to the plotter and will be ignored if typed to the plotter. Most of these are fairly obvious; they include: CLEARSCREEN, WIPECLEAN, WRAP, HIDETURTLE, SHOWTURTLE, SNAP, WIPE, and DISPLAY. The plotter also has a more restricted argument range than the displays. NOPLOTTER turns off the plotter.

PM

Music box command. Takes no inputs. Causes the output of previous NOTE commands to be played on the music box. As the music is played, it is erased form temporary storage and must be recompiled before being played again.

PRINT (PR)

Takes one input, which evaluates to either a word or a list. Prints out the evaluated input on the console.

```
?PR SUM 4 4
8
?PR [SUM 4 4]
```

PRINTOUT (PO)

Takes one input, the name of a procedure, and prints the text of the procedure on the console.

PRINTOUT ALL (PO ALL)

Takes no inputs and prints out all names, procedures, and arrays currently defined in the workspace.

PRINTOUT ARRAY (PO ARRAY)

Takes one input, the name of an array, and prints out the array's dimensions and type.

PRINTOUT ARRAYS (PO ARRAYS)

Takes no inputs. Performs PO ARRAY for all currently defined arrays.

PRINTOUT FILE (PO FILE)

Takes as input a file name and prints out the contents of the file.

PRINTOUT INDEX (POI)

Takes no inputs. Prints the names of all files in the user's current index (see USE).

PRINTOUT LINE (POL)

Takes one input, a line number, and prints out the specified line. Can be used only while defining or editing.

PRINTOUT NAMES (PO NAMES)

No inputs. Prints out all currently defined names and their values.

PRINTOUT PROCEDURES

No inputs. Prints out the definitions of all procedures currently defined in the workspace.

PRINTOUT TITLE (POT)

No inputs. Used only while defining or editing, prints the title of the procedure currently being defined/edited.

PRINTOUT TITLES (POTS)

Takes no inputs. Prints out the titles of all procedures defined in the workspace.

PRINTOUT TREE (PO TREE)

Takes no inputs. Prints out the tree structure of the filing system, starting with the current index. See SETINDEX.

PROCEDURES

Special word. See PRINTOUT and ERASE.

PRODUCT

Takes two numeric inputs and outputs their product.

```
PRINT PRODUCT 2 3   prints 6.
```

QUOTIENT

Takes two numeric inputs and outputs their quotient. (The first input is divided by the second).

```
PRINT QUOTIENT 4 2   prints 2.
```

RANDOM

No inputs. Outputs a one-digit random integer.

READ

Takes one input, the name of an existing file (stored on disk), and reads it into the workspace.

READPTR

Takes no inputs. Reads into the workspace from the paper tape reader.

REMAINDER

Takes two numeric inputs, outputs the remainder of the first number divided by the second.

```
REMAINDER 27 5  outputs 2
```

REQUEST

No inputs. When encountered in the execution of a procedure, causes LOGO to pause and wait for a line to be typed in. This input is treated as a list and printed on the console.

RIGHT (RT)

Takes one numeric input. Commands the floor or display turtle to turn to the right (clockwise) the number of degrees which you give as an input.

```
RIGHT 30 tells the turtle to turn 30 degrees to the right.
```

RTOUCH

Takes no inputs. Checks to see whether the right sensor on the floor turtle has been activated by touching something. If so, outputs TRUE, otherwise FALSE.

RUN

Takes one input, a list, and evaluates this list just as if it were typed in at the console.

```
?RUN [PRINT SUM 18 5]
23
```

SEND

Takes two inputs. The first is the number of a console, the second is a list which is printed out as a message on the specified console. Used to send messages to users currently on the system.

SENTENCE (SE)

Takes two inputs. If both are lists it puts the elements of the lists together to make a single list and outputs that list. If either of its inputs is not a list it first changes the input to a one-element list and then proceeds as above.

```
                SENTENCE [FOO] [FOO BAR] outputs [FOO FOO BAR]
                SENTENCE [WHAT IS YOUR] "NAME? outputs [WHAT IS YOUR NAME?]
```

SENTENCE also has the variable number of inputs feature.

```
                            (SENTENCE [PIECES] "OF "A [BIG LIST])
                outputs [PIECES OF A BIG LIST]
```

SETASIZE

Used to allocate additional space for arrays. Takes one input, the amount of space to be allocated, in wods (every word stores one pointer array element, every two words store one integer or floating point array element.) The maximum input is aprroximately 11000, which makes it possible to expand array space by a factor of three. It should be remembered, however, that the use of SETASIZE involves a re-initialization of the user's workspace, which means that a "WELCOME TO 11LOGO" message will be printed upon completion of the command.

SETHEADING

Takes one numeric input. Specifies the direction (in degrees) in which the display turtle points. Zero is straight up.

SETINDEX (SETI)

Takes one input, the name of an index. The input may be a list if there are one or more indices separating the desired index from the current one. In this case the list contains the names of all the indices on the "path" which must be followed through the file tree in order to reach the new index. This index becomes the new root index, which means that all filing commands such as POI and PO TREE refer only to that portion of the file system below the desired index.

SETTURTLE (SETT)

Takes one input which is a list of 3 numbers. This input assigns, in order, the x-coordinate, the y-coordinate, and the heading of the display turtle.

SETX

Takes one numeric input. Moves the display turtle horizontally to the specified coordinate.

SETXY

Takes two numeric inputs. Moves the display turtle to the position designated.

SETY

Takes one numerical input. It moves the display turtle vertically to the specified coordinate.

SHOWTURTLE (ST)

Takes no inputs. It tells the computer to put the display turtle (a little triangle) on the display screen.

SIN

Takes one numeric input (representing degrees) and outputs the sine of the specified angle.

SNAP

Takes no inputs. Outputs a reference to "the stuff on the display screen." For example,

MAKE "PIC SNAP causes :PIC to refer to whatever is currently on the screen. Later you can cause another copy to appear by saying:

DISPLAY :PIC or erase it by saying
WIPE :PIC

Each SNAP has associated with it a "starting location" and an "ending location" which determine where the picture appears when you say DISPLAY. The starting location is normally th center of the screen (but see also NEWSNAP) and the ending location is the position of the turtle when the SNAP command is given. SNAP's cannot be saved with the WRITE command (they are ignored).

STEP

Takes one input, the name of a procedure (or ALL). Causes the procedure to be TRACEd, and in addition, before each line of the procedure is executed, the line is typed out and LOGO waits for the user to respond. There are three options:

1) Typing a carriage return causes the line to be executed and goes on to the next line.

(2) Typing cntrl-G terminates execution (as always).

(3) Typing cntrl-Z generates a PAUSE, which may be exited by using the CONTINUE command.

A STEP is removed by typing ERASE STEP <procedure name (or ALL)>.

STARTDISPLAY

Takes one input, either 0 or 1. Tells the computer to give you a display turtle. The input determines the size of the display. 0 is a half-size display, 1 is a full-size display.

STOP

No inputs. Used in a procedure, it terminates execution of the procedure and transfers control to the calling procedure.

STORE

Used to assign a value to a specific element of an array. Takes the same number of inputs as the corresponding DEFINEARRAY command which created the array. These inputs are: the array name, the coordinates of the desired element, and the value to be assigned. Zero-origin indexing is in effect here also (see GET). If the array has more than one dimension, STORE and its inputs must be enclosed in parentheses.

SUM

Takes two numeric inputs and outputs their sum. Also accepts multiple inputs (see below).

SYSPR

Takes as input a list which is printed out on all system consoles currently in use.

TEST

A command it takes an argument (which must evaluate to TRUE or FALSE) and puts the result in a "test box." You can then ask IFTRUE or IFFALSE to look in the box.

```
        TEST 2 = 2 IFTRUE PRINT "TRUE     prints TRUE
        TEST 2 = 5 IFTRUE PRINT "TRUE     prints nothing
```

TEXT

Takes one input, a procedure name, and outputs the text of the procedure as a list. (Note the distinction between TEXT and PRINTOUT. The latter is a command; it does not output).

THEN

Noise word, which separates the conditional clause of an IF-expression from the statement to be conditionally executed. (see IF.)

THING

Takes one input, a name, and outputs the value assigned to that name.

TIME

Outputs a 3-element list which is supposed to be HR/MIN/SEC. Curently unimplemented.

TITLE

Special word, refers to the title of a procedure. See PRINTOUT and EDIT.

TO

Used to define procedures. Takes a variable number of inputs, the first of which is the name of the procedure to be defined. The rest are names of inputs to the procedure.

TOOT

Takes one input. Tells the floor turtle to toot its horn <input> times.

TOPLEVEL

Takes no inputs. Used only in a procedure, it returns control in the procedure immediately to the top level.

TRACE

Takes one input which is a procedure name (or ALL). Causes the computer to print out a message each time the procedure is executed, indicating the inputs to the procedure and the output, if any. To get rid of it, type ERASE TRACE followed by the procedure name (or ALL).

TURTLE

Takes one numeric input, the number of the turtle to be used. Gives the user access to the specified turtle. Although it is possible to have up to four turtles on the system, normally only 1 and 2 will be acceptable turtle numbers.

.TYI

Takes one input, a device specification. See .TYO. The next piece of information received from the device is interpreted as a number and given as output.

.TYO

Takes two inputs. The first is a device specification. This may be either a device number or device name. The device names LOGO recognizes are the following:

"LIGHT	for the light box
"TUR1	for turtle 1
"TUR2	for turtle 2
"MUSIC	for the music box
"PLOTTER	for the plotter
"TTY	for your own console

The second input is a character code which specifies a character to be relayed to the given device as output.

TYOWAIT

Similar to CTYOWAIT, but takes one extra input (which comes first). This input is the specification of the device to wait for. These specifications are the same as for .TYO.

TYPE

Essentially the same as PRINT, but does not carriage return.

TYPEIN

Equivalent to FIRST of REQUEST.

USE

One input, which is the name of an index. If this is not the user's root index, the index may be specified by a list which contains the "path" of indices through the file tree leading to the desired index. This command specifies the default index to be used for all READ and WRITE commands, as well as POI and PO TREE.

.VERSION

No inputs. Outputs a number which tells which version of LOGO is currently running.

VLEN

Music box command. Takes no inputs. Outputs the total length of music compiled for the current voice.

VOICE

Music box command. Takes one input which must be a number between 1 and 4. Directs the output of subsequent NOTE commands to the various voices. The current voice remains the same for all subsequent NOTE commands until the next VOICE command is given. If no VOICE command is given, the system outputs to voice number 1.

WAIT

Takes one numeric input, causes LOGO to pause for the specified length of time (given in 1/60ths of a second.)

WIPE

Takes one input, a snap, and erases all appearances of the snap from the display screen.

WIPECLEAN

Takes no inputs. It tells the computer to get rid of everything on the screen, leaving the turtle where it is.

WORD

Takes two inputs, which must be words. WORD takes the two inputs and puts them together to make one word.

WORDP

Takes one input and tests to see if the input is a word. If so, outputs TRUE, otherwise FALSE.

WRAP

Takes no inputs. Causes an image which goes out of bounds on the display screen to "wrap around" to the other side.

WRITE

Takes one input, the name of a file to be created. Saves the current contents of the workspace under that file name.
will put everything in your workspace into a file named X.

WRITEPTP

Takes no inputs. It writes the contents of your workspace onto a paper tape, for permanent storage. (Make sure the paper tape printer is ready!)

XCOR

Takes no inputs. Outputs the present X-coordinate of the display turtle.

YCOR

Takes no inputs. Outputs the present Y-coordinate of the turtle.

VARIABLE NUMBER OF INPUTS

Certain primitives can take a variable number of inputs. To use this feature, the primitive and its inputs must be grouped together in parentheses. For example,

(SUM 5 6 7 8) outputs 26

The primitives having this feature are:

```
 FPRINT
 FPUT   -(FPUT "A "B "C [WOW] outputs [ A B C WOW]
 LIST
 LPUT   - (LPUT "A "B "C [WOW]) outputs [WOW A B C ]
 PRINT
 PRODUCT
 SENTENCE
 SING   -  Number of inputs must be even
 SUM
 TYPE
 WORD
```

SPECIAL CHARACTERS

```
 ! Used for comments.  Anything appearing after it on a line is ignored.
 # Takes one input which must be a word and evaluates its input, e.g.,
               #(WORD "POOH 5) will execute the procedure named POOH5.
 ( ) used for grouping.
 [ ] Used to indicate lists.
 * infix PRODUCT
 + infix SUM
 - infix DIFFERENCE
 / infix QUOTIENT
 : Same as THING, except that it does not evaluate its argument.
 > infix GREATER
 < infix LESS
 = infix EQUAL
 \ infix MOD
 ← infix MAKE, ie., "A ← 5 is equivalent to MAKE "A 5

 ↑A  Echoes as carriage return
 ↑B  Echoes as blank
 ↑G  Break
 ↑H  Echoes as backspace
 ↑I  Echoes as tab
 ↑J  Echoes as linefeed
 ↑Q  Superquotes the following character (i.e. prevents it from being
evaluated)
```

↑Z When DEBUG is off it is equivalent to ↑G. When DEBUG is on, it causes a pause.

EDITING CHARACTERS

rubout Deletes the pevious character.
↑W Deletes the previous word.
↑Y Places the previous line typed in the edit buffer

When you have something in the edit buffer (after using EDL or ↑Y) the following can be used:

↑C Copy the next character.
↑N Copy the next word.
↑S Skip the next word.
↑R Copy the rest of the edit buffer.