M6800 LISP 1.5 PROGRAMMERS MANUAL

FRITS VAN DER WATEREN.

Getting started with LISP.

It is most likely that you have to patch the I/O part of LISP to your environment. For this purpose a listing of this section is supplied. This will be clear enough to patch the neccesary changes.
Only a few notes on the device-table,which starts at $106.
LISP talks to a device via software I/O ports. Each port has four link-pointers in the device-table.

Link #1 should point to a routine that reads one character,which is returned in the A-accumulator.

Link #2 should point to a routine that outputs one character, which is supplied in the A-accumulator.

Link #3 should point to an OPEN-routine,which can be used either to initialise the handler or to open a file on file-oriented devices. In that case a filename can be supplied. The pointer to that name is in 'ARG2' ($16). When the filename is omitted, the contents of 'ARG2' is NIL.
The OPEN-routine must return T or NIL in the X-register.
T in the case that the operation succeeds and NIL when it failed.
NIL=$0000 and T=$0DCC

Link #4 should point to a CLOSE-routine,that closes the current file on this device. This routine must also return T or NIL in X.
A special case is the close-routine for device #1,which checks if the ETX (ctrl/C) key is pressed. On ETX it jumps back to the LISP-interpreter otherwise it continues. This routine is called several times during evaluation,but only via link-address #4 of device #1.

How to start LISP.

The start address of LISP is $100. When started it immediately allocates all contiguous RAM from the last location of LISP.
It reserves the upper 1/8 part of memory for stack. One stack element takes always two bytes.
And the lower 7/8 part of memory for cell-storage. One cell is 4 bytes.
It then initialises the OBLIST, i.e. all user entered ATOM'S are deleted. Furthermore a garbage collection is forced, by setting the FREE-LIST to NIL. After all this work LISP is ready ror input and types a prompt (*) on the systems console (dev #1).
To restart LISP, go at address $103. All ATOM'S on the OBLIST are then maintained. Only a garbage collection is forced to clean up the cell-storage.

Input - Output.

This LISP version is capable of handling more than one device. This is done by adding a port-number as an argument to I/O-functions. However when this argument is NIL or it is omitted, then I/O-port 1 is assumed.
Port-1 is the systems console (full duplex) and is interfaced by an 'ACIA' located at $FF00 and $FF01.
Port-2 is an high-speed reader interfaced by the A-section of a 'PIA' at $FF10 and $FF11.
Port-3 is a high-speed punch interfaced by the B-section of a 'PIA' at $FF12 and $FF13.
Port-4 and 5 are unused,but may be patched to user I/O routines.

Functions that can use I/O-ports are:

```
(PRIN1 X DEV)
(PRINT X DEV)
(TERPRI DEV)
(READCH DEV)
(READ1 DEV)
(READ DEV)
(OPEN DEV filename)
(CLOSE DEV)
```

Where 'DEV' is the port number.
OPEN or CLOSE performed on port 3 result in punching of about 15 inch of blank tape.
The input format is very free. You can insert spaces,tabs,comma's and carriage-returns anywhere in the input string as separators to make it more readable. But an ATOM must be a contiguous string of characters ofcourse, without any of these special characters. But when you want to use these seperator charcaters and/or the special characters:left-parenthesis,right-parentesis and a dot, then you have to super-quote the ATOM. This is because the function QUOTE will not work on these characters. As superquotes the characters ' and " are used. The string to be quoted must be enclosed by either of them. However when this string is closed by a carriage-return it closes the quoting too, and the carriage-return is included as last character in the string.
So when you want for-instance a single carriage-return, then write: "↵
where ↵ stands for carriage-return.

Error recovery.

When you have typed an error,you can delete the whole line by typing CTRL/X (cancel). The system echoes a @ and continues reading on the next line.
Previous characters can be deleted by typing a CTRL/H (back-space) The system then echoes this backspace,but only when there are characters in the buffer.
When your terminal has no back-space feature,you can use the RUBOUT-key. The deleted characters are then echoed in reverse order,enclosed by square brackets.
An evaluation can be aborted by pressing CTRL/C (ETX).

Errors during runtime.

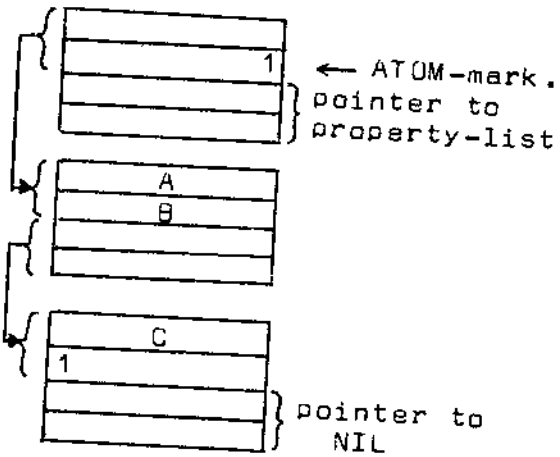
The following are the possible errors and their meaning.

*SYNTAX ERROR	The input string is not a legal S-expression.
*ILLEGAL NUMBER	Illegal number structure in input string
NON ATOMIC ARG:<list>	This error occurs when a function wants an atomic argument. Value is NIL and evaluation continues.
ATOMIC ARG:<atom>	Atomic argument for CAR. Value is NIL and evaluation continues.
*ILLEGAL FUNCTION:<list>	This function is not available in the OBLIST.
*ILLEGAL GO:<atom>	This label is not inside the current PROG. Only local labels are allowed.
*ILLEGAL DEVICE:<dev>	This device is not available in the DEVICE-TABLE.
OVERFLOW	An arithmetic overflow has occured.The value is taken modulo 2^15,and evaluation continues.
TOO MANY ARGS	There are too many arguments supplied with a SUBR or FSUBR type of function. The remaining arguments are ignored and evaluation continues.
TOO LITTLE ARGS	There are too little arguments supplied with a FSUBR-type of function. The missing arguments are taken to be NIL and evaluation continues.
TOO MANY ARGS FOR:<list>	There are too many arguments supplied with an EXPR or FEXPR-type of function.
TOO LITTLE ARGS FOR:<LIST>	There are too little arguments supplied with an EXPR or FEXPR-type of function.
*NON NUMERIC ARG: <list>	The argument must be a number for: PLUS, MINUS,TIMES,QUOTIENT and GREATERP.
*STACK OVERFLOW	The stack has been used up. The system restarts at $103
*MEMORY FULL	All cells have been used up. The system restart at $103

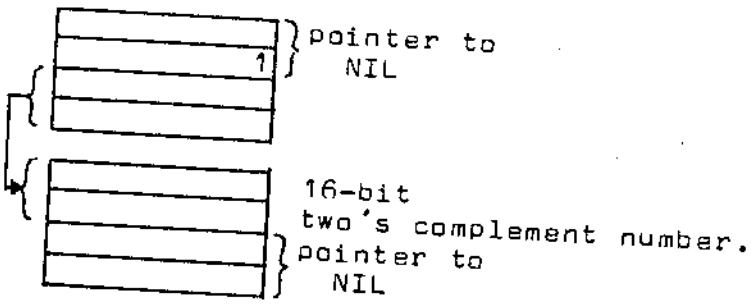
note: All errors marked with a * are fatal errors, that is: the current evaluation is aborted and a prompt is typed.

A brief description of the cell structures.

The basic element in LISP is the cell ,which requires four bytes in this implementation. The first two bytes represent the CAR-address,and the last two bytes the CDR-address. All addressing of the cells is even-word,that is: bits-0 and 1 are always zero. However bit-0 of the CAR-part is used as an ATOM-mark and bit-0 of the CDR-part is used as a mark by the garbage collector. Now an ATOM looks like:



When the print-name consists of an odd number of characters,a filler (negative byte) is used.
A numeric ATOM always consists of two cell's and looks like:



Numbers in LISP are represented as a 16-bit signed integer. And are recognized as such when an Atom begins with a numeric character or a + or - sign.
The internal number representation in LISP is an ATOM with a print-name of NIL.

Elementary functions.

(CAR X)	value is the 'CAR-part' of X. The CAR of an ATOM is illegal, and will result in an error message. Never access the property of a SUBR or FSUBR indicato because it contains the pointer to a machine language routine.
(CDR X)	value is the 'CDR-part' of X.
(CONS X Y)	value is the list: (X . Y)
(QUOTE X)	value is X litteraly
(RPLACA X Y)	replace the 'CAR-part' of X by Y;value is X
(RPLACD X Y)	replace the 'CDR-part' of X by Y;value is X Be very careful with these two functions, because you alter an existing list-structure.

I/O functions.

(READCH DEV)	read one single character from 'DEV'. The value is an ATOM of this single character.
(READ1 DEV)	read an atom; value is this atom.
(READ DEV)	read an S-expression; value is this expression.
(TEREAD)	flush the input buffer; value is NIL.
(PRIN1 DEV)	print atom X;value is NIL.
(PRINT X DEV)	print S-expression X; value is NIL. When there are more than 55 characters on a line then the first occurence of a space will be replaced by a carriage-return and linefeed.
(TERPRI DEV)	print a CR and LF on DEV; value is NIL
(OPEN DEV NAME)	open a file on DEV with NAME as file name. filename may be omitted for non file oriented devices.
(CLOSE DEV)	close file on DEV

Predicates.

(ATOM X)	if X is an atom,value is T else NIL.
(NUMBER X)	if X is a number,value is T else NIL.
(NULL X)	if X is NIL, value is T else NIL.
(EQ X Y)	if X is the same as Y,value is T else NIL
(GREATER X Y)	if X is greater than Y,value is T else NIL.

Arithmetic functions.

(PLUS X Y)	value is X + Y
(MINUS X Y)	value is X - Y
(TIMES X Y)	value is X * Y
(QUOTIENT X Y)	value is X / Y

Some miscelaneous functions.

(SETQ X Y)	X is set to the value of Y;this is also the value of this function. X is looked up first on the association-list, before altering the permanent value of X.
(PUTPROP X Y Z)	The property-list of atom X is extended by property Y under an indicator Z. If the indicator already exists,then its property is altered;value is Y
(GET X Y)	Get the property saved under the indicator Y from the property-list of atom X.
(SASSOC X Y)	Lookup X on the association-list Y, when found return its value;else NIL.
(ALIST)	return the current systems association-list.
(COND (A B)(X Y))	General conditional. Each argument of COND is a conditional expression. The number of these expressions may be infinite. A conditional expression consists of two items; the first item (A) is evaluated and when NIL, the next conditional expression is taken,etc; otherwise the second item (B) is evaluated, which is then the final result of COND. When there is no second item, then the value of the first item is the result, when not NIL. When COND runs out of its argument list, then the result is NIL; rather than an error message.
(LIST A B C...Z)	The result is a list of all its evaluated arguments. The number of arguments may be infinite.
(EVAL X Y)	Evaluate X with Y as association-list. In fact this is the LISP-interpreter itself. When a variable is evaluated, then the assocation-list is always looked up first (by SASSOC) rather than to take the permanent value from its property-list.
(APPLY X Y Z)	Apply the argument Y to the function X with Z as initial association-list.
(PROG (A B....) L (statement1) (statement 2))	With this function we are capable of writing programs in LISP. The first argument of PROG is a list of variables used inside the PROG. These are the so called 'local variables' and are initialy set to NIL. The remainder is a list of labels and statements. PROG's may be nested to any level.
(GO L)	Goto label L (litterally). GO is restricted to local labels only. A reference outside a PROG will result in an error.
(RETURN X)	Return from a PROG with the value of X. RETURN is the only legal exit from a PROG.

(FUNCTION X) The result is the list: (FUNARG X alist)
where 'alist' is the association-list at the
time FUNCTION is called. When this list is
scanned a next time, then X is evaluated with
'alist' as association-list rather than the
current association-list.
This is the so called FUNARG meganism.

OBLIST OBLIST is an ATOM who's permanent value
is a list of all ATOM's, known by the
system so far.
Some other objects are:
SUBR,FSUBR,EXPR,FEXR,LAMBDA,FUNARG,NIL,T
All these objects have their self as value.

note: All underlined arguments are taken litterally.
All other arguments are evaluated before they are applied
to the calling function.

I hope you will enjoy LISP.
And if any problems with LISP are encountered, or when you have
any suggestions for improvements, which can be implemented in
a next version, please let me know.

Frits van der Wateren
van 't Hoffstraat 140
NL 2014 BK Haarlem
The Netherlands

```
*
******************    LISP 1.5    ******************
*
*    COPYRIGHT 1978 BY:
*
*    FRITS VAN DER WATEREN
*    VAN 'T HOFFSTRAAT 140
*    2014 RK  HAARLEM
*    THE NETHERLANDS
*
*
*
*
* LISP STARTS AT $100.
*      AND AUTOMATICALY ALLOCATES ALL CONTIGUOUS MEMORY
*      AND RESERVES 7/8 FOR CELL STORAGE (4 BYTES PER CE
*      AND 1/8 FOR STACK (2 BYTES PER WORD).
*
* RESTART IS AT $103
*      THE CURRENT OBLIST IS MAINTAINED BUT THE
*      A-LIST BINDINGS ARE LOST.
*      FURTHERMORE A GARBAGE COLLECTION IS FORCED.
*
*THE DEVICE TABLE IS LOCATED AT $106
*      AND CAN BE EXTENDED BY TWO MORE DEVICES.
*      IMPLEMENTED HANDLERS ARE:
*       #1 TTY    WITH AN ACIA AT $FF00
*       #2 READER WITH A PIA AT $FF10 (A)
*       #3 PUNCH  WITH A PIA AT $FF10 (B)
*
```

```
0000                    ORG     0

0000                NIL     RMB     4       ATOM NIL
0004                FWAM    RMB     2       FIRST WORD OF CELL STORAGE
0006                LWAM    RMB     2       LAST WORD OF CELL STORAGE
0008                STACK   RMB     2       BOTTOM OF STACK
000A                LIMIT   RMB     2       STACK-LIMIT REGISTER
000C                OBLSTB  RMB     2       POINTER TO OBJECT-LIST.
000E                N       RMB     2
0010                FREE    RMB     2       POINTER TO FREE LIST
0012                CURCEL  RMB     2       POINTER TO CURRENT CELL
0014                ARG1    RMB     2       HOLDS ARGUMENT 1 DURING EVALUAT
0016                ARG2    RMB     2
0018                ARG3    RMB     2
001A                PROGB   RMB     2       BEGIN OF CURRENT PROG-LIST
001C                RUNP    RMB     2       RUNNING POINTER ON PROG-LIST
001E                OBLSTP  RMB     2
0020                OP      RMB     2
0022                AP      RMB     2
0024                NUM1    RMB     2
0026                NUM2    RMB     2
0028                SAVEX   RMB     2
002A                TEMPX   RMB     2
002C                MP      RMB     2
002E                PNAME   RMB     2
0030                STEPC   RMB     1
0031                SIGN    RMB     1
0032                CP      RMB     2       POINTER TO CURRENT INPUT CHAR.
0034                DELFLG  RMB     1
0035                DEVICE  RMB     2       POINTS TO I/O HANDLER LINK ADDRE
0037                CCOUNT  RMB     1       CHARACTER COUNT FOR OUTPUT.

0001                M       EQU     1       MARKER
0080                F       EQU     $80     FILL CHARACTER

0038                FIRSTC  EQU     *       REMAINDER IS INPUT-BUFFER

0100                        ORG     $100
00FF                LASTC   EQU     *-1
```

BPTR 2 — FOR BUFFER WRITE
OUTLL 1
ERRLOC 2

```
0100 7E 0130          JMP     START   COLD START.
0103 7E 0182          JMP     RSTART  WARM START.
                *
                * DEVICE TABLE
                *
                * EACH ENTRY CONSISTS OF FOUR POINTERS
                * TO I/O HANDLERS.
                * POINTER 1 IS INPUT HANDLER
                * POINTER 2 IS OUTPUT HANDLER
                * POINTER 3 IS OPEN OR INITIALISE
                * POINTER 4 IS CLOSE
                *
                * DEVICE #1 IS THE SYSTEM DEVICE
                *
                *
0106 02         DEVTBL  FDB     TTYIN,TTYOUT,TTYOPN,ABORT
010E 02                 FDB     RDRIN,DUMMY,RDROPN,TR
0116 02                 FDB     DUMMY,PUNOUT,PUNOPN,PUNCLS
011E 00                 FDB     0,0,0,0         #4 FOR EXTENTION.
0126 00                 FDB     0,0,0,0         #5 FOR EXTENTION.
012E 00                 FDB     0       ZERO ENDS THE TABLE
```

```
0130 CE 0D95    START   LDX     #NAMNIL+M  *     SET ATOM NIL AT $0000
0133 DF 00              STX     NIL        *
0135 CE 0D9C            LDX     #PRPNIL    *
0138 DF 02              STX     NIL+2      *
013A CE 0DAC            LDX     #OBLIST
013D DF 0C              STX     OBLSTB     BEGIN OF OBLIST
013F CE 1278            LDX     #LISPSP
0142 DF 04              STX     FWAM       FIRST FREE CELL.
0144 86 8B              LDAA    #%10001011 *
0146 A7 00      SEEK    STAA    0,X     ALLOCATE ALL
0148 A1 00              CMPA    0,X      CONTIGUOUS MEMORY
014A 26 05              BNE     END       RIGHT AFTER LISP.
014C 6F 00              CLR     0,X
014E 08                 INX
014F 20 F5              BRA     SEEK
0151 35         END     TXS
0152 9F 08              STS     STACK
0154 96 08              LDAA    STACK   RESERVE STACK SPACE-
0156 D6 09              LDAB    STACK+1
0158 D0 05              SUBB    FWAM+1
015A 92 04              SBCA    FWAM
015C 44                 LSRA
015D 56                 RORB
015E 44                 LSRA
015F 56                 RORB
0160 44                 LSRA
0161 56                 RORB
0162 D0 09              SUBB    STACK+1
0164 92 08              SBCA    STACK
0166 43                 COMA
0167 97 06              STAA    LWAM    LWAM=-(STACK/8-STACK)
0169 C6 80              LDAB    #$80
016B D7 07              STAB    LWAM+1  ON PAGE-BOUNDARY + 128
016D 4C                 INCA             STACK-LIMIT ON PAGE BOUNDARY.
016E 97 0A              STAA    LIMIT    128 BYTES FOR STACK OVERFLOW
0170 7F 000B            CLR     LIMIT+1
0173 CE 0DC4    RSTR    LDX     #OBL1
0176 DF 16              STX     ARG2
0178 CE 0DAC            LDX     #OBLIST
017B EE 02              LDX     2,X
017D EE 02              LDX     2,X
017F BD 077E            JSR     RPLACA  RESTORE OBLIST
0182 9E 08      RSTART  LDS     STACK
0184 CE 000E            LDX     #N
0187 6F 00      CLRLOC  CLR     0,X     CLEAR (NIL) WORK SPACE.
0189 08                 INX
018A 8C 001E            CPX     #OBLSTP
018D 26 F8              BNE     CLRLOC
018F DE 0E              LDX     N
0191 DF 14              STX     ARG1
0193 BD 05F0            JSR     OPEN    OPEN DEV #1
0196 CE 0D82            LDX     #HED    PRINT: LISP 1.5 AND
0199 BD 03A5            JSR     PMESSG   VERSION NUMBER.
```

```
                ***********************************
                *                                 *
                *       THE LISP INTERPRETER      *
                *                                 *
                ***********************************
```

154

```
019C 9E 08    LISP    LDS     STACK
019E DE 0E            LDX     N
01A0 BD 06F7          JSR     TEREAD
01A3 BD 06EF          JSR     TERPRI
01A6 BD 0437          JSR     READ    READ ONE S-EXPRESSION FROM DEV #
01A9 DF 14            STX     ARG1
01AB DE 0E            LDX     N
01AD DF 16            STX     ARG2
01AF DF 1A            STX     PROGB
01B1 DE 14            LDX     ARG1
01B3 BD 08B9          JSR     EVAL    EVALUATE
01B6 DF 14            STX     ARG1
01B8 DE 0E            LDX     N
01BA DF 16            STX     ARG2
01BC BD 0602          JSR     PRINT   AND PRINT RESULT ON DEV #1
01BF 20 D8            BRA     LISP
```

ARG1 = 1A9C

```
                    *
                    * GET A CELL FROM THE FREE LIST
                    * ON EXIT X POINTS TO THIS CELL
                    *
01C1 DE 10  1C3 CELL    LDX     FREE
01C3 27 0D      1D1     BEQ     GCOL        FREE LIST IS EMPTY
01C5 DF 12              STX     CURCEL      GET CELL
01C7 EE 02              LDX     2,X         ADVANCE POINTER TO NEXT
01C9 DF 10              STX     FREE         FREE CELL ON LIST
01CB DE 12              LDX     CURCEL
01CD 6F 02              CLR     2,X         CLEAN UP THIS CELL!
01CF 6F 03              CLR     3,X
01D1 39                 RTS
                    *
                    * GARBAGE COLLECTOR.
                    *
01D2 DE 14  1EF GCOL    LDX     ARG1
01D4 8D 65              BSR     MARKL       MARK 3 CURRENT ARGS.
01D6 DE 16              LDX     ARG2
01D8 8D 61              BSR     MARKL
01DA DE 18              LDX     ARG3
01DC 8D 5D              BSR     MARKL
01DE DE 1A              LDX     PROGB
01E0 8D 59              BSR     MARKL       MARK PROG
01E2 DE 0C              LDX     OBLSTB
01E4 8D 55              BSR     MARKL       MARK OBLIST, SAVEX
01E6 30                 TSX
01E7 09                 DEX
01E8 9C 08  216 GCOL1   CPX     STACK
01EA 27 18              BEQ     GCOL3
01EC BD 03C7            JSR     PUSHX       MARK ALL ACTIVE LISTS
01EF A6 01              LDAA    1,X          ON THE STACK
01F1 E6 02              LDAB    2,X
01F3 D0 0D              SUBB    OBLSTB+1
01F5 92 0C              SBCA    OBLSTB
01F7 2B 04              BMI     GCOL2       POINTS IN SYSTEM AREA.
01F9 EE 01              LDX     1,X
01FB 8D 3E              BSR     MARKL
01FD BD 03E5    GCOL2   JSR     PULLX
0200 08                 INX
0201 08                 INX
0202 20 E4              BRA     GCOL1
0204 DE 0E      GCOL3   LDX     N
0206 DF 10              STX     FREE
0208 DE 06              LDX     LWAM
```

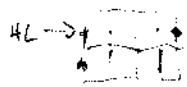
* PUSH ANY OTHER ACTIVE PTRS
(NUM1, NUM2, &SAVEX)
‡ KILL STK ON EXIT
≠ NO CHK FOR > SYS AREA !
N.B. DIES IF ODR ON STACK (NOT JUST FULL CELLS)

```
020A 9C 0C      SWEEP   CPX     OBLSTB  NOW SWEEP ALL UNMARKED CELLS
020C 27 1D              BEQ     SWPDON   ONTO THE FREE LIST.
020E 09                 DEX
020F 09                 DEX
0210 09                 DEX
0211 09                 DEX
0212 A6 03              LDAA    3,X
0214 67 03              ASR     3,X     UNMARK CELL MARK IN CDR
0216 68 03              ASL     3,X
0218 46                 RORA
0219 25 EF              BCS     SWEEP
021B 96 10              LDAA    FREE    LINK AN UNMARKED CELL
021D D6 11              LDAB    FREE+1   TO THE FREE-LIST
021F 6F 00              CLR     0,X
0221 6F 01              CLR     1,X
0223 A7 02              STAA    2,X
0225 E7 03              STAB    3,X
0227 DF 10              STX     FREE
0229 20 DF              BRA     SWEEP

022B DE 10      SWPDON  LDX     FREE
022D 27 03              BEQ     FULL    NO FREE-LIST!
022F 7E 01C1            JMP     CELL

0232 CE 0D5B    FULL    LDX     #FL.MS
0235 BD 03A5    SYSERR  JSR     PMESSG
0238 7E 0182            JMP     RSTART
```

GOOD



FREE

```
                *
                * MARK ONE LIST
                *
023B 27 41  282 MARKL   BEQ     MARKEX  EMPTY LIST
023D 4F         MARKL   CLRA
023E 36                 PSHA            SET BOTTOM OF WORKSTACK TO NIL.
023F 36                 PSHA
0240 8C 0001    MARK1   CPX     #1      NUMERIC CELL?
0243 27 26              BEQ     MRKNUM  YES
0245 DF 10              STX     FREE
0247 96 10              LDAA    FREE
0249 D6 11              LDAB    FREE+1
024B D0 0D              SUBB    OBLSTB+1
024D 92 0C              SBCA    OBLSTB  NO; LIST IN SYSTEM AREA?
024F 2B 21              BMI     MARK2   YES; UP ONE LEVEL
0251 56                 RORB            NO, ATOM CELL?
0252 25 12              BCS     MRKATM  YES
0254 A6 03              LDAA    3,X
0256 46                 RORA            CELL MARKED?
0257 25 19              BCS     MARK2   YES,UP ONE LEVEL
0259 6C 03              INC     3,X     NO,MARK CELL
025B BD 03C7            JSR     PUSHX 3F9
025E EE 00              LDX     0,X     DOWN ONE LEVEL
0260 20 DE              BRA     MARK1

0262 6C 03  266 MRKNAM  INC     3,X     MARK PRINTNAME OF ATOM
0264 EE 02              LDX     2,X
0266 09         MRKATM  DEX
0267 26 F9              BNE     MRKNAM
0269 20 07              BRA     MARK2
026B BD 03E5 2FF MRKNUM JSR     PULLX   MARK ONE NUMERIC CELL
026E EE 02              LDX     2,X     ALREADY MARKED
0270 6C 02              INC     2,X
0272 BD 03E5    MARK2   JSR     PULLX   UP ONE LEVEL
0275 27 07              BEQ     MARKEX  TOP LEVEL! SO EXIT.
0277 EE 02              LDX     2,X
0279 09                 DEX             TAKE CARE OF MARK
027A 27 F6              BEQ     MARK2
027C 20 C2              BRA     MARK1
027E 39         MARKEX  RTS
```

MARKL : IF X=NIL THEN RETURN
ELSE PUSH NIL

MARK1 : IF X=NUMERIC THEN: MARK, POP X, POP X

```
                        *
                        * INPUT ONE CHAR FROM TTY
                        *
027F B6 FF00    TTYIN   LDAA    ACIACS
0282 47                 ASRA
0283 24 FA              BCC     TTYIN
0285 B6 FF01            LDAA    ACIADA
0288 84 7F              ANDA    #$7F
028A 39                 RTS
                        *
                        * OUTPUT ONE CHAR. ON TTY
                        *
028B F6 FF00    TTYOUT  LDAB    ACIACS
028E 57                 ASRB
028F 57                 ASRB
0290 24 F9              BCC     TTYOUT
0292 B7 FF01            STAA    ACIADA
0295 39                 RTS
                        *
                        * OPEN TTY (INITIALISE)
                        *
0296 86 01      TTYOPN  LDAA    #%1
0298 B7 FF00            STAA    ACIACS
029B 20 10              BRA     TR


                        *
                        * ABORT IF CTRL/C (ETX) IS PRESSED
                        *
029D B6 FF00    ABORT   LDAA    ACIACS  IF CTRL/C IS PRESSED
02A0 47                 ASRA            THEN ABORT EVALUATION.
02A1 24 0A              BCC     TR
02A3 B6 FF01            LDAA    ACIADA
02A6 81 03              CMPA    #3
02A8 26 03              BNE     TR
02AA 7E 019C            JMP     LISP
02AD 7E 06EB    TR      JMP     TRUE

     FF00       ACIACS  EQU     $FF00
     FF01       ACIADA  EQU     $FF01
```

```
                        *
                        * INPUT ONE CHARACTER FROM HIGH SPEED READER.
                        *
02B0 B6 FF11    RDRIN   LDAA    RDRC
02B3 2A FB              BPL     RDRIN
02B5 B6 FF10            LDAA    RDR
02B8 43                 COMA
02B9 84 7F              ANDA    #$7F
02BB 27 F3              BEQ     RDRIN   IGNORE NULLS
02BD 81 7F              CMPA    #$7F
02BF 27 EF              BEQ     RDRIN   IGNORE RUBOUT
02C1 81 0A              CMPA    #$A
02C3 27 EB              BEQ     RDRIN   IGNORE LF
02C5 39         DUMMY   RTS
                        *
                        * OPEN READER (INITIALISE)
                        *
02C6 7F FF11    RDROPN  CLR     RDRC
02C9 7F FF10            CLR     RDR
02CC 86 2E              LDAA    #%101110
02CE B7 FF11            STAA    RDRC
02D1 B6 FF10            LDAA    RDR     SET FLAG
02D4 20 D7              BRA     TR
                        *
                        * OUTPUT ONE CHARACTER ON PUNCH.
                        *
02D6 F6 FF13    PUNOUT  LDAB    PUNC        Po
02D9 2A FB              BPL     PUNOUT
02DB B7 FF12            STAA    PUN
02DE 39                 RTS
                        *
                        * OPEN PUNCH (INITIALISE) AND PUNCH LEADER.
                        *
02DF 7F FF13    PUNOPN  CLR     PUNC
02E2 7F FF12            CLR     PUN
02E5 86 2E              LDAA    #%101110
02E7 B7 FF13            STAA    PUNC
                        *
                        * CLOSE PUNCH; PUNCH TRAILER.
                        *
02EA 86 96      PUNCLS  LDAA    #150            PUNOPN
02EC 97 30              STAA    STEPC
02EE 4F         TRAIL   CLRA
02EF 8D E5              BSR     PUNOUT
02F1 7A 0030            DEC     STEPC
02F4 26 F8              BNE     TRAIL
02F6 20 B5              BRA     TR

     FF10       RDR     EQU     $FF10
     FF11       RDRC    EQU     $FF11
     FF12       PUN     EQU     $FF12
     FF13       PUNC    EQU     $FF13
```

```
                        * BUFFER ONE LINE FROM INPUT DEVICE
                        *
02F8 86 40      DELINE  LDAA    #'@
02FA BD 038F            JSR     OUTCH1  OUTPUT A '@' ON CTRL/X
02FD BD 03B9            JSR     CRLF     AND IGNORE WHOLE LINE.
0300 86 2A      ACC INPSTR  LDAA    #'*
0302 BD 038F            JSR     OUTCH1  OUTPUT A PROMPT.
0305 CE 0038            LDX     #FIRSTC INIT BUFFER
0308 6F 00              CLR     0,X
030A 7F 0034    NEXTCF  CLR     DELFLG-ACB
030D DF 32      ADX NEXTC   STX     CP
030F BD 039A            JSR     INCH 891
0312 81 18              CMPA    #$18    CTRL/X (CAN)?
0314 27 E2              BEQ     DELINE
0316 81 08              CMPA    #8      BACKSPACE?
0318 27 2F              BEQ     DEL
031A 81 7F              CMPA    #$7F    DELETE?
031C 27 2B              BEQ     DEL
031E A7 01              STAA    1,X     NO,STORE CHAR.
0320 7D 0034            TST     DELFLG  AT END OF DEL.SEQUENCE?
0323 27 04              BEQ     NORML
0325 86 5D              LDAA    #']     YES,CLOSE STRING OF
0327 8D 66              BSR     OUTCH1  DELETED CHARACTERS.
0329 A6 01      NORML   LDAA    1,X
032B 8D 62              BSR     OUTCH1  ECHO TYPED CHAR.
032D 81 0D              CMPA    #$D     CR?
032F 27 08              BEQ     CR
0331 8C 00FF            CPX     #LASTC  NO,BUFFER FULL?
0334 27 D4              BEQ     NEXTCF  YES
0336 08                 INX
0337 20 D1              BRA     NEXTCF
0339 CE 0038    CR      LDX     #FIRSTC RESET BUFFER POINTER
033C DF 32              STX     CP
033E 7F 0037            CLR     CCOUNT
0341 8D 4C              BSR     OUTCH1  REFLECT CR & LF.
0343 86 0A              LDAA    #$A
0345 8D 48              BSR     OUTCH1
0347 20 1B              BRA     GETC
0349 8C 0038    DEL     CPX     #FIRSTC
034C 27 BF              BEQ     NEXTC   BUFFER UNDERFLOW!
034E 09                 DEX
034F 81 08              CMPA    #8
0351 27 0D              BEQ     SHWDEL+2
0353 7D 0034            TST     DELFLG  TYPE DELETED CHARACTERS
0356 26 06              BNE     SHWDEL   ON 'DEL' ONLY.
0358 86 5B              LDAA    #'[
035A 97 34              STAA    DELFLG
035C 8D 31              BSR     OUTCH1
035E A6 01      SHWDEL  LDAA    1,X
0360 8D 2D              BSR     OUTCH1
0362 20 A9              BRA     NEXTC
```

```
                        *
                        * GET ONE CHAR. FROM INPUT BUFFER
                        *
0364 DE 32 B4A  GETC    LDX     CP 1ABC
0366 A6 00              LDAA    0,X     GET CHARACTER FROM INPUT BUFFER

0368 81 0D              CMPA    #$D     END OF LINE?
036A 27 94              BEQ     INPSTR  YES,GO READ NEXT LINE
036C A6 01              LDAA    1,X     NO,GET NEXT CHAR.
036E 08                 INX
036F DF 32              STX     CP 1ABC
0371 81 2C              CMPA    #',     MAKE A 'SPACE' OF THE
0373 27 08              BEQ     SP       FOLLOWING CHARACTERS:
0375 81 09              CMPA    #9      'KOMMA' 'TAB' CARRIAGE RET.'
0377 27 04              BEQ     SP
0379 81 0D              CMPA    #$D
037B 26 02              BNE     NOSP
037D 86 20      SP      LDAA    #$20
037F 39         NOSP    RTS
                        *
                        * OUTPUT ONE CHAR.
                        *
0380 7C 0037    OUTCH   INC     CCOUNT
0383 81 20              CMPA    #'      SPACE?
0385 26 08              BNE     OUTCH1
0387 96 37              LDAA    CCOUNT  YES,MORE THAN 55 CHAR. ON ALINE?
0389 81 37              CMPA    #55
038B 2E 2C              BGT     CRLF    YES,OUTPUT A CR & LF.
038D 86 20              LDAA    #'      NO
038F DF 2A      OUTCH1  STX     TEMPX
0391 DE 35              LDX     DEVICE
0393 EE 02              LDX     2,X
0395 AD 00              JSR     0,X
0397 DE 2A              LDX     TEMPX
0399 39                 RTS
                        *
                        * INPUT ONE CHARACTER
                        *
039A DF 2A      INCH    STX     TEMPX
039C DE 35              LDX     DEVICE
039E EE 00              LDX     0,X
03A0 AD 00              JSR     0,X
03A2 DE 2A              LDX     TEMPX
03A4 39                 RTS
```

```
                        *
                        * PRINT MESSAGE ON DEV #1
                        *
03A5 DF 2C      PMESSG  STX     MP
03A7 DE 0E              LDX     N
03A9 8D 4C              BSR     ASSIGN  ASSIGN DEV #1
03AB 8D 0C              BSR     CRLF
03AD DE 2C              LDX     MP
03AF A6 00      PMESG1  LDAA    0,X
03B1 8D CD              BSR     OUTCH   PRINT STRING
03B3 6D 00              TST     0,X
03B5 08                 INX
03B6 2A F7              BPL     PMESG1
03B8 39                 RTS
                        *
                        * NEW LINE
                        *
03B9 36         CRLF    PSHA
03BA 86 0D              LDAA    #$D     CARRIAGE RETURN
03BC 8D C2              BSR     OUTCH    & LINE FEED ON TTY.
03BE 86 0A              LDAA    #$A
03C0 8D BE              BSR     OUTCH
03C2 7F 0037            CLR     CCOUNT
03C5 32                 PULA
03C6 39                 RTS
```

```
                        *
                        * PUSH X ONTO STACK
                        *
03C7 DF 2A      PUSHX   STX     TEMPX
03C9 30                 TSX
03CA 9C 0A              CPX     LIMIT
03CC 2B 11              BMI     STKOVF  STACK OVERFLOW!
03CE E6 01              LDAB    1,X     SAVE RETURN ADDRESS
03D0 37                 PSHB
03D1 E6 00              LDAB    0,X
03D3 37                 PSHB
03D4 D6 2A              LDAB    TEMPX   PLACE X ONTO STACK
03D6 E7 00              STAB    0,X
03D8 D6 2B              LDAB    TEMPX+1
03DA E7 01              STAB    1,X
03DC DE 2A              LDX     TEMPX
03DE 39                 RTS
03DF CE 0D66    STKOVF  LDX     #SO.MS
03E2 7E 0235            JMP     SYSERR
                        *
                        * PULL X FROM STACK
                        *
03E5 30         PULLX   TSX
03E6 E6 02              LDAB    2,X     GET X FROM STACK
03E8 D7 2A              STAB    TEMPX
03EA E6 03              LDAB    3,X
03EC D7 2B              STAB    TEMPX+1
03EE 33                 PULB
03EF E7 02              STAB    2,X     REPLACE RETURN ADDRESS
03F1 33                 PULB
03F2 E7 03              STAB    3,X
03F4 DE 2A              LDX     TEMPX
03F6 39                 RTS
```

```
                    *
                    * ASSIGN DEV. # TO INPUT- AND OUTPUT HANDLER
                    * X MUST POINT TO AN NUMERIC ATOM.
                    * IF X IS NIL THEN ASSIGN DEV # 1
                    *
03F7 C6 01          ASSIGN  LDAB    #1
03F9 DF 28                  STX     SAVEX   SAVE ATOM POINTER
03FB 27 11                  BEQ     DEV1
03FD EE 00                  LDX     0,X
03FF 09                     DEX
0400 26 21                  BNE     ID.ER   NON NUMERIC DEV. #
0402 DE 28                  LDX     SAVEX
0404 EE 02                  LDX     2,X
0406 A6 00                  LDAA    0,X
0408 26 19                  BNE     ID.ER   DEV. # > 256!
040A E6 01                  LDAB    1,X
040C 27 15                  BEQ     ID.ER   DEV. # = 0!
040E CE 00FE        DEV1    LDX     #DEVTBL-8
0411 86 08          NXTDEV  LDAA    #8      OK SO FAR!
0413 08             X8      INX              NOW LOOKUP DEV.TABLE
0414 4A                     DECA
0415 26 FC                  BNE     X8
0417 A6 00                  LDAA    0,X
0419 27 08                  BEQ     ID.ER   ZERO ENDS THE THE TABLE
041B 5A                     DECB
041C 26 F3                  BNE     NXTDEV
041E DF 35                  STX     DEVICE  SET LINK ADDRESS
0420 DE 14                  LDX     ARG1
0422 39                     RTS

0423 CE 0D74        ID.ER   LDX     #ID.MS
0426 7E 044E                JMP     FATAL
                    *
                    * GET ONE CHAR. BUT SKIP SPACES.
                    *
0429 BD 0364        GETCS   JSR     GETC
042C 81 20                  CMPA    #' 
042E 27 F9                  BEQ     GETCS
0430 39                     RTS
                    *
                    * PUT BACK ONE CHAR. IN INPUT BUFFER
                    *
0431 DE 32          PUTBAK  LDX     CP
0433 09                     DEX
0434 DF 32                  STX     CP
0436 39                     RTS
```

```
                        *
                        *  (READ DEV)
                        * READ ONE S-EXPRESSION FROM 'DEV'
                        *
0437 8D BE      READ    BSR     ASSIGN BCE.
0439 8D EE      READE   BSR     GETCS  C4)
043B 81 28              CMPA    #'(
043D 27 15              BEQ     S.EXPR   EXPRESSION BEGINS WITH: (
043F 81 2E              CMPA    #'.
0441 27 08              BEQ     SN.ER
0443 81 29              CMPA    #')
0445 27 04              BEQ     SN.ER    . AND ) ARE ILLEGAL NOW
0447 8D E8              BSR     PUTBAK
0449 20 50              BRA     READ1E  ATOMIC EXPRESSION.

044B CE 0CCE    SN.ER   LDX     #SN.MS
044E BD 03A5    FATAL   JSR     PMESSG
0451 7E 019C            JMP     LISP

0454 BD 0429    S.EXPR  JSR     GETCS


0457 81 2E              CMPA    #'.
0459 27 F0              BEQ     SN.ER   DOT IS ILLEGAL NOW!

045B DE 0E      RDLIST  LDX     N       EXPRESSION IS A LIST STRUCTURE.
045D 81 29              CMPA    #')
045F 27 19              BEQ     S.END
0461 81 2E              CMPA    #'.
0463 27 16              BEQ     DOT     DOT NOTATION.
0465 8D CA              BSR     PUTBAK
0467 8D D0              BSR     READE   READ NEXT FORM
0469 BD 03C7            JSR     PUSHX
046C 8D BB              BSR     GETCS
046E 8D EB              BSR     RDLIST  AND GO IN RDLIST AGAIN.
0470 DF 16              STX     ARG2
0472 BD 03E5            JSR     PULLX
0475 DF 14              STX     ARG1
0477 BD 076A            JSR     CONS    NOW CONS ALL FORMS TO A LIST
047A 39         S.END   RTS

047B 8D BC      DOT     BSR     READE   READ LAST FORM
047D DF 16              STX     ARG2
047F 8D A8              BSR     GETCS
0481 81 29              CMPA    #')      WICH MUST BE CLOSED WITH: )
0483 26 C6              BNE     SN.ER
0485 DE 16              LDX     ARG2
0487 39                 RTS
```

```
                    *
                    *  (READCH DEV)
                    *  READ ONE CHARACTER FROM 'DEV'
                    *
0488 BD 03F7        READCH  JSR     ASSIGN
048B BD 0364                JSR     GETC    GET ONE CHAR.
048E DE 32                  LDX     CP
0490 DF 2E                  STX     PNAME
0492 7E 0547                JMP     AATOM    AND BUILD AN ATOM OF IT.

0495 7E 044B        SN.ER1  JMP     SN.ER
                    *
                    *  (READ1 DEV)
                    *  READ AN ATOM FROM 'DEV'.
                    *
0498 BD 03F7        READ1   JSR     ASSIGN
049B BD 0429        READ1E  JSR     GETCS
049E DE 32                  LDX     CP
04A0 DF 2E                  STX     PNAME   'PNAME' POINTS TO BEGIN OF ATOM
 04A2 81 27                 CMPA    #''
04A4 27 7F          BEQ     SQUOTE
04A6 81 22                  CMPA    #'"     ' AND " ARE SUPER-QUOTE CHAR.
04A8 27 7B                  BEQ     SQUOTE
04AA 09                     DEX
04AB DF 32                  STX     CP
04AD BD 0364        RDATOM  JSR     GETC    FIND END OF ATOM
04B0 81 28                  CMPA    #'(
04B2 27 0C                  BEQ     ENDATM
04B4 81 29                  CMPA    #')
04B6 27 08                  BEQ     ENDATM
04B8 81 20                  CMPA    #' 
04BA 27 04                  BEQ     ENDATM
04BC 81 2E                  CMPA    #'.
04BE 26 ED                  BNE     RDATOM
04C0 BD 0431        ENDATM  JSR     PUTBAK  IF FIRST CHAR. IS + OR -
04C3 DE 2E                  LDX     PNAME    OR 0-9 THEN ATOM IS NUMERIC.
04C5 09                     DEX
04C6 9C 32                  CPX     CP      ANY ATOM?
04C8 27 C8                  BEQ     SN.ER1  NO
04CA 08                     INX
04CB 5F                     CLRB
04CC A6 00                  LDAA    0,X
04CE 81 2B                  CMPA    #'+
04D0 27 0F                  BEQ     NATOM
04D2 16                     TAB
04D3 81 2D                  CMPA    #'-
04D5 27 0A                  BEQ     NATOM
04D7 5F             NEATM   CLRB
04D8 81 30                  CMPA    #'0
04DA 2D 63                  BLT     AATOM
04DC 81 39                  CMPA    #'9
04DE 2E 67                  BGT     AATOM
04E0 09                     DEX
```

```
                        *
                        * ATOM IS NUMERIC.
                        *
04E1 D7 31   COE   NATOM   STAB    SIGN
04E3 4F                   CLRA
04E4 5F                   CLRB
04E5 D7 24                STAB    NUM1
04E7 D7 25                STAB    NUM1+1
04E9 9C 32         NXTDIG CPX     CP        END OF ATOM?
04EB 27 27                BEQ     BLDNMB
04ED 08                   INX
04EE A6 00                LDAA    0,X       NO,GET NEXT DIGIT
04F0 81 30                CMPA    #'0        AND CHECK IF 0-9.
04F2 2D 2B                BLT     IN.ER
04F4 81 39                CMPA    #'9
04F6 2E 27                BGT     IN.ER
04F8 84 0F                ANDA    #$F
04FA 97 26                STAA    NUM2
04FC 96 24                LDAA    NUM1      NOW SHIFT DIGIT INTO NUMBER
04FE D6 25                LDAB    NUM1+1     JUST READ SO FAR.
0500 58                   ASLB
0501 49                   ROLA
0502 58                   ASLB
0503 49                   ROLA
0504 DB 25                ADDB    NUM1+1
0506 99 24                ADCA    NUM1
0508 58                   ASLB
0509 49                   ROLA
050A DB 26                ADDB    NUM2
050C 89 00                ADCA    #0
050E 97 24                STAA    NUM1
0510 D7 25                STAB    NUM1+1
0512 20 D5                BRA     NXTDIG
0514 7D 0031       BLDNMB TST     SIGN
0517 27 03                BEQ     POSN
0519 BD 0C34              JSR     NEG       NEG. NUMBER IS TWO'S COMPL.
051C 7E 0CAB       POSN   JMP     PUTIN

051F CE 0D13       IN.ER  LDX     #IN.MS
0522 7E 044E              JMP     FATAL
```

```
                        *
                        * ATOM IS SUPER QUOTED.
                        *
0525 97 31      SQUOTE  STAA    SIGN    SAVE QUOTE CHAR.
0527 08                 INX
0528 DF 2E              STX     PNAME
052A 09                 DEX
052B 08         BQ      INX
052C A6 00              LDAA    0,X     NOW READ ALL CHAR.
052E DF 32              STX     CP       BETWEEN THE QUOTES,
0530 81 0D              CMPA    #$D        A CR ALSO DELIMITS THE QUOTING
0532 27 13              BEQ     AATOM
0534 91 31              CMPA    SIGN
0536 26 F3              BNE     BQ
0538 86 20              LDAA    #'      DELETE LAST QUOTE CHAR.
053A A7 00              STAA    0,X      IN THE BUFFER.
053C 09                 DEX
053D DF 32              STX     CP
053F 08                 INX
0540 9C 2E              CPX     PNAME   IS THERE A QUOTED STRING?
0542 26 03              BNE     AATOM
0544 DE 0E              LDX     N       NO,RETURN NIL
0546 39                 RTS
```

```
                        *
                        *  ATOM IS ALPHA NUMERIC.
                        *
0547 CE 0DAC DAA AATOM  LDX     #OBLIST
054A EE 02              LDX     2,X
054C EE 02              LDX     2,X       GET VALUE OF OBLIST
054E DF 24              STX     NUM1
0550 EE 00              LDX     0,X
0552 DF 1E      NXTOBJ  STX     OBLSTP
0554 EE 00              LDX     0,X       GET ATOM FROM OBLIST
0556 EE 00              LDX     0,X        AND GET ITS PRINTNAME
0558 09                 DEX
0559 DF 20              STX     OP
055B DE 2E              LDX     PNAME
055D DF 22              STX     AP
055F DE 20      NXT2C   LDX     OP
0561 27 2E              BEQ     NOMTCH
0563 A6 00              LDAA    0,X
0565 E6 01              LDAB    1,X       GET 2 CHAR. FROM ATOM ON OBLIST
 0567 EE 02             LDX     2,X
0569 DF 20              STX     OP
056B DE 22              LDX     AP
056D A1 00              CMPA    0,X       AND COMPARE WITH 2 CHAR.
056F 26 20              BNE     NOMTCH     OF ATOM IN INPUT BUFFER
0571 9C 32              CPX     CP
0573 27 13              BEQ     MATCH
0575 E1 01              CMPB    1,X
0577 26 18              BNE     NOMTCH
0579 08                 INX
057A 08                 INX
057B DF 22              STX     AP
057D 09                 DEX
057E 9C 32              CPX     CP
0580 26 DD              BNE     NXT2C
0582 DE 20              LDX     OP
0584 26 0B              BNE     NOMTCH
0586 C6 80              LDAB    #F
0588 C1 80      MATCH   CMPB    #F
058A 26 05              BNE     NOMTCH
058C DE 1E              LDX     OBLSTP    ATOM IS ON THE OBLIST
058E EE 00              LDX     0,X        RETURN ITS POINTER AS VALUE.
0590 39                 RTS
```

```
0591 DE 1E      NOMTCH  LDX     OBLSTP   NO MATCH SO FAR,
0593 EE 02              LDX     2,X      TAKE NEXT ATOM FROM OBLIST.
0595 26 BB              BNE     NXTOBJ
                *
                * ATOM NOT ON OBLIST
                *
0597 DF 14              STX     ARG1
0599 DF 16              STX     ARG2
059B BD 076A            JSR     CONS     BUILD ATOM CELL
059E DF 18              STX     ARG3
05A0 DE 32              LDX     CP
05A2 08                 INX
05A3 DF 22              STX     AP
05A5 96 2F              LDAA    PNAME+1
05A7 98 23              EORA    AP+1
05A9 46                 RORA
05AA 24 0B              BCC     BLDATM   EVEN NUMBER OF CHAR.
05AC BD 076A            JSR     CONS
05AF DF 16              STX     ARG2     ODD NUMBER OF CHAR,
05B1 DE 22              LDX     AP        SO ADD A FILL CHAR.
05B3 C6 80              LDAB    #F
05B5 20 0A              BRA     ODD
05B7 BD 076A    BLDATM  JSR     CONS     NOW BUILD AN ATOM,
05BA DF 16              STX     ARG2      WITH 2 CHAR. PER CELL
05BC DE 22              LDX     AP
05BE 09                 DEX
05BF E6 00              LDAB    0,X
05C1 09         ODD     DEX
05C2 A6 00              LDAA    0,X
05C4 DF 22              STX     AP
05C6 DE 16              LDX     ARG2
05C8 A7 00              STAA    0,X
05CA E7 01              STAB    1,X
05CC DE 18              LDX     ARG3
05CE BD 077E            JSR     RPLACA   UPDATE ATOM-CELL
05D1 6C 01              INC     1,X       AND SET ATOM-MARK
05D3 DE 22              LDX     AP
05D5 9C 2E              CPX     PNAME
05D7 26 DE              BNE     BLDATM
05D9 DE 18              LDX     ARG3     AND ADD IT ON TOP OF THE OBLIST
 05DB DF 14             STX     ARG1
05DD DE 24              LDX     NUM1
05DF EE 00              LDX     0,X
05E1 DF 16              STX     ARG2
05E3 BD 076A            JSR     CONS
05E6 DF 16              STX     ARG2
05E8 DE 24              LDX     NUM1
05EA BD 077E            JSR     RPLACA   UPDATE OBLIST!
05ED DE 18              LDX     ARG3     VALUE IS THE NEW ATOM
05EF 39                 RTS
```

```
                    *
                    *   (OPEN DEV FILENAME)
                    *
                    * OPEN A FILE ON 'DEV' WITH 'FILENAME'.
                    * ON NON FILE ORIENTED DEVICES,
                    * THIS CALL ONLY INITIALISES THE HANDLER
                    *
05F0 BD 03F7        OPEN    JSR     ASSIGN
05F3 DE 35                  LDX     DEVICE  GET OPEN-LINK FOR 'DEV'
05F5 EE 04                  LDX     4,X
05F7 6E 00                  JMP     0,X     AND GO TO REQUESTED HANDLER
                    *
                    *   (CLOSE DEV)
                    *
                    * CLOSE FILE ON 'DEV'
                    *
05F9 BD 03F7        CLOSE   JSR     ASSIGN
05FC DE 35                  LDX     DEVICE
05FE EE 06                  LDX     6,X
0600 6E 00                  JMP     0,X
```

```
                        *
                        * (PRINT X DEV)
                        *
0602 DE 16      PRINT   LDX     ARG2
0604 BD 03F7            JSR     ASSIGN
0607 4F         PRINTE  CLRA
0608 36                 PSHA            BOTTOM OF STACK IS NIL.
0609 36                 PSHA

060A DF 14      PRINT2  STX     ARG1
060C BD 06DC            JSR     ATOM    ATOMIC?
060F 27 32              BEQ     PRINT7  NO
0611 DE 14              LDX     ARG1
0613 8D 41              BSR     PRINTE  YES,PRINT ATOM
0615 BD 03E5    PRINT3  JSR     PULLX
0618 26 01              BNE     PRINT4
061A 39                 RTS

061B EE 02      PRINT4  LDX     2,X     GET CDR-PART
061D DF 14              STX     ARG1
061F 27 18              BEQ     PRINT5
0621 BD 06DC            JSR     ATOM    CDR PART ATOMIC?
0624 27 1A              BEQ     PRINT6  NO
0626 86 20              LDAA    #'
0628 BD 0380            JSR     OUTCH
062B 86 2E              LDAA    #'.
062D BD 0380            JSR     OUTCH   YES,PRINT A DOT
0630 86 20              LDAA    #'
0632 BD 0380            JSR     OUTCH
0635 DE 14              LDX     ARG1
0637 8D 1D              BSR     PRINTE  AND PRINT ATOM

0639 86 29      PRINT5  LDAA    #')
063B BD 0380            JSR     OUTCH
063E 20 D5              BRA     PRINT3

0640 86 20      PRINT6  LDAA    #'
0642 8C                 FCB     $8C     SKIP

0643 86 28      PRINT7  LDAA    #'(
0645 BD 0380            JSR     OUTCH
0648 DE 14              LDX     ARG1
064A BD 03C7            JSR     PUSHX
064D EE 00              LDX     0,X
064F 20 B9              BRA     PRINT2
```

```
                    *  (PRIN1 X DEV)
                    *
                    * PRINT 'PRINTNAME' OF ATOM X
                    * OR NUMERIC VALUE IN CASE OF A NUMBER
                    *
0651 DE 16          PRIN1   LDX     ARG2
0653 BD 03F7                JSR     ASSIGN
0656 BD 029D F8F    PRINIE  JSR     ABORT
0659 DE 14                  LDX     ARG1
065B A6 01                  LDAA    1,X
065D 46                     RORA
065E 24 6F                  BCC     NA.ER       NOT ATOMIC
0660 EE 00                  LDX     0,X         GET LINK TO PRINT NAME
0662 09                     DEX                 CLEAR ATOM MARK
0663 27 11                  BEQ     PRNMB       NUMERIC!
0665 A6 00          FAC PRCHAR  LDAA    0,X     PRINT CHAR.STRING UNTIL
0667 BD 0380                JSR     OUTCH        FILLCHAR. OR NIL
066A A6 01                  LDAA    1,X
066C 2B 03                  BMI     DONE
066E BD 0380                JSR     OUTCH
0671 EE 02          DONE    LDX     2,X
0673 26 F0                  BNE     PRCHAR
0675 39                     RTS                 VALUE OF PRIN1 IS NIL
```

```
                    *
                    * ATOM IS NUMERIC
                    *
0676 DE 14      PRNMB   LDX     ARG1
0678 EE 02              LDX     2,X     GET NUMBER
067A E6 01              LDAB E  1,X
067C A6 00              LDAA D  0,X
067E 2A 0C              BPL     POS
0680 86 2D              LDAA    #'-     NEGATIVE NUMBER,
0682 BD 0380            JSR     OUTCH   PRINT MINUS SIGN
0685 A6 00              LDAA    0,X
0687 E6 01              LDAB    1,X
0689 BD 0C34            JSR     NEG     AND NEGATE NUMBER
068C CE 06C5    POS     LDX     #DECTBL
068F 7F 0022            CLR     AP
0692 7F 0031    DEC1    CLR     SIGN    BINARY TO DEC CONVERSION.
0695 7C 0031    DEC2    INC     SIGN     WITH LEADING ZERO SUPPRESSION.
0698 E0 01              SUBB    1,X
069A A2 00              SBCA    0,X
069C 2A F7              BPL     DEC2
069E EB 01              ADDB    1,X
06A0 A9 00              ADCA    0,X
06A2 36                 PSHA
06A3 37                 PSHB
06A4 96 31              LDAA    SIGN
06A6 4A                 DECA
06A7 26 05              BNE     DEC3
06A9 7D 0022            TST     AP
06AC 27 07              BEQ     DEC4    LEADING ZERO
06AE 8B 30      DEC3    ADDA    #'0     MAKE ASCII
06B0 97 22              STAA    AP
06B2 BD 0380            JSR     OUTCH   AND PRINT
06B5 08         DEC4    INX
06B6 08                 INX
06B7 33                 PULB
06B8 32                 PULA
06B9 6D 01              TST     1,X     AT END OF TABEL?
06BB 26 D5              BNE     DEC1
06BD 17                 TBA
06BE 8B 30              ADDA    #'0     YES,PRINT LAST DIGIT.
06C0 BD 0380            JSR     OUTCH
06C3 20 1C              BRA     FALSE

06C5 27         DECTBL  FDB     10000,1000,100,10,0

06CF CE 2CDA    NA.ER   LDX     #NA.MS
06D2 BD 03A5    PNTARG  JSR     PMESSG
06D5 DE 14              LDX     ARG1
06D7 BD 0607            JSR     PRINTE  PRINT ARGUMENT
06DA 20 13              -BRA    ~~TERPRI~~
                                LISP
```

```
                        *
                        *  (ATOM X)
                        *
                        * IF X IS ATOMIC THEN TRUE ELSE NIL
                        *
06DC A6 01      ATOM    LDAA    1,X
06DE 46                 RORA
06DF 25 0A              BCS     TRUE    CELL IS ATOMIC
06E1 DE 0E      FALSE   LDX     N
06E3 39                 RTS
                        *
                        *  (NUMBER X)
                        *
                        * IF X IS A NUMBER, VALUE IS TRUE; ELSE NIL.
                        *
06E4 EE 00      NUMBER  LDX     0,X
06E6 09                 DEX
                        *
                        *  (NULL X)
                        *
                        * IF X IS NIL RETURN 'T';ELSE NIL.
                        *
06E7 9C 0E      NULL    CPX     N
06E9 26 F6              BNE     FALSE
06EB CE 0DCC    TRUE    LDX     #T
06EE 39                 RTS
                        *
                        * (TERPRI DEV)
                        *
                        * TERMINATES PRINTLINE (CR & LF)
                        * VALUE IS NIL
                        *
06EF BD 03F7    TERPRI  JSR     ASSIGN
06F2 BD 03B9            JSR     CRLF
06F5 20 EA              BRA     FALSE
```

```
                    *
                    * (TEREAD)
                    *
                    * RESET  INPUT BUFFER
                    *
06F7 CE 0038        TEREAD  LDX     #FIRSTC
06FA DF 32                  STX     CP
06FC 86 0D                  LDAA    #$D
06FE A7 00                  STAA    0,X
0700 20 DF                  BRA     FALSE
                    *
                    *  (EQ X Y)
                    *
                    * IF X IS EQUAL TO Y THE VALUE IS 'T',OTHERWISE NIL
                    *
0702 9C 16          EQ      CPX     ARG2
0704 27 E5                  BEQ     TRUE
0706 EE 00                  LDX     0,X
0708 09                     DEX
0709 26 D6                  BNE     FALSE
070B DE 16                  LDX     ARG2
070D EE 00                  LDX     0,X
070F 09                     DEX
0710 26 CF                  BNE     FALSE
0712 DE 14                  LDX     ARG1
0714 BD 0BD4                JSR     GET2N   IF BOTH ARGUMENTS ARE NUMERIC
0717 E0 01                  SUBB    1,X      THEN COMPARE THERE VALUES
0719 26 C6                  BNE     FALSE
071B A2 00                  SBCA    0,X
071D 26 C2                  BNE     FALSE
071F 20 CA                  BRA     TRUE
```

DE = Numi
HC = Numi

```
                        *
                        *  (QUOTE X)
                        *
                        * PREVENT X FROM EVALUATION.
                        * VALUE IS X.
                        *
0721 DE 14      QUOTE   LDX     ARG1
0723 27 02              BEQ     NO.ARG
0725 EE 00              LDX     0,X
0727 39         NO.ARG  RTS
                        *
                        *  (GREATERP X Y)
                        *
                        * IF X >= Y THEN TRUE ELSE FALSE
                        *
0728 BD 0BD4    GREATR  JSR     GET2N   GET 2 VALUES
072B A1 00              CMPA    0,X
072D 2D B2              BLT     FALSE
072F 2E BA              BGT     TRUE
0731 E1 01              CMPB    1,X
0733 22 B6              BHI     TRUE
0735 20 AA              BRA     FALSE
                        *
                        *  (FUNCTION X)
                        *
                        * RETURNS THE LIST:
                        *   (FUNARG X ALIST)
                        *
0737 DF 18      FUNCTI  STX     ARG3
0739 DE 16              LDX     ARG2
073B DF 14              STX     ARG1
073D DE 0E              LDX     N
073F DF 16              STX     ARG2
0741 8D 27              BSR     CONS    (CONS ALIST NIL)
0743 DF 16              STX     ARG2
0745 DE 18              LDX     ARG3
0747 EE 00              LDX     0,X
0749 DF 14              STX     ARG1
074B 8D 1D              BSR     CONS    (CONS X (CONS ALIST NIL)
074D DF 16              STX     ARG2
074F CE 0E88            LDX     #FUNARG
0752 DF 14              STX     ARG1
0754 20 14              BRA     CONS    (CONS FUNARG(CONS X(CONS ALIST N
                        *
                        *  (ALIST)
                        *
                        * RETURNS THE CURRENT ASSOCIATION LIST AS VALUE.
                        *
0756 DE 16      ALIST   LDX     ARG2
0758 39                 RTS
```

```
                    *
                    *  (CAR X)
                    *
0759 A6 01          CAR     LDAA    1,X
075B 44                     LSRA
075C 25 03                  BCS     AA.ER   CAR OF AN ATOM IS ILLEGAL.
075E EE 00                  LDX     0,X
0760 39                     RTS
0761 CE 0D07        AA.ER   LDX     #AA.MS
0764 7E 06D2                JMP     PNTARG
                    *
                    *  (CDR X)
                    *
0767 EE 02          CDR     LDX     2,X
0769 39                     RTS
                    *
                    *  (CONS X Y)
                    *
                    *  VALUE IS A LIST OF WHICH THE CAR-PART IS X
                    * AND THE CDR-PART IS Y
                    *
876A BD 01C1        CONS    JSR     CELL    GET A NEW CELL
876D 96 14                  LDAA    ARG1
876F D6 15                  LDAB    ARG1+1
0771 A7 00                  STAA    0,X     ARG1 TO CAR-PART
0773 E7 01                  STAB    1,X
                    *
                    *  (RPLACD X Y)
                    *
                    * REPLACE CDR-LINK OF X BY Y
                    *
0775 96 16          RPLACD  LDAA    ARG2
0777 D6 17                  LDAB    ARG2+1
0779 A7 02                  STAA    2,X
077B E7 03                  STAB    3,X
077D 39                     RTS
                    *
                    * (RPLACA X Y)
                    *
                    * REPLACE CAR-LINK OF X BY Y
                    *
077E 96 16          RPLACA  LDAA    ARG2
0780 D6 17                  LDAB    ARG2+1
0782 A7 00                  STAA    0,X
0784 E7 01                  STAB    1,X
0786 39                     RTS
```

```
                        *
                        *  (SETQ X Y)
                        *
                        * X GETS AS VALUE, THE VALUE OF Y
                        * VALUE OF SETQ IS THIS VALUE!
                        *
0787 BD 03C7    SETQ    JSR     PUSHX
078A 27 3E              BEQ     ARG?
078C EE 02              LDX     2,X
078E 27 3A              BEQ     ARG?
0790 EE 00              LDX     0,X     GET Y.
0792 DF 14              STX     ARG1
0794 DE 16              LDX     ARG2
0796 BD 03C7            JSR     PUSHX   PUSH A-LIST
0799 DE 14              LDX     ARG1
079B BD 08B9            JSR     EVAL    EVALUATE Y
079E DF 18              STX     ARG3
07A0 BD 03E5            JSR     PULLX   PULL UP A-LIST.
07A3 DF 16              STX     ARG2
07A5 BD 03E5            JSR     PULLX   PULL UP ARG-LIST.
07A8 EE 00              LDX     0,X     GET X
07AA DF 14              STX     ARG1
07AC BD 0870            JSR     SASSOC  ON A-LIST?
07AF 27 0E              BEQ     SETAPV  NO.
07B1 DF 14              STX     ARG1    YES,REPLACE
07B3 DE 18              LDX     ARG3     ASSOCIATED VALUE BY Y.
07B5 DF 16              STX     ARG2
07B7 DE 14              LDX     ARG1
07B9 BD 0775            JSR     RPLACD
07BC EE 02              LDX     2,X     RETURN WITH Y.
07BE 39                 RTS
07BF DE 18      SETAPV  LDX     ARG3    NOT ON A-LIST; SO PUT THE
07C1 DF 16              STX     ARG2     VALUE,UNDER AN APVAL IND.,
07C3 CE 0E04            LDX     #APVAL     ON ATOM' PROP-LIST.
07C6 DF 18              STX     ARG3    THIS IS DONE BY PUTPROP.
07C8 20 0E              BRA     PUTPRP

07CA CE 0D46    ARG?    LDX     #TLA.MS
07CD 7E 044D            JMP     FATAL
```

```
                        *
                        *  (PUTPROP ATOM PROPERTY INDICATOR)
                        *
                        * PUT 'PROPERTY' ON PROPERTY-LIST OF 'ATOM'
                        * TOGETHER WITH 'INDICATOR'.
                        *
07D0 DE 14      GETIND  LDX     ARG1    GET NEXT IND. ON
07D2 EE 02              LDX     2,X      PROPERTY LIST
07D4 EE 02              LDX     2,X
07D6 DF 14              STX     ARG1
07D8 DE 14      PUTPRP  LDX     ARG1
07DA EE 02              LDX     2,X
07DC 27 12              BEQ     NO.PRP  PROPERTY-LIST EXHAUSTED
07DE EE 00              LDX     0,X     GET INDICATOR.
07E0 9C 18              CPX     ARG3    IS IT THE REQUESTED IND.?
07E2 26 EC              BNE     GETIND  NO!
07E4 DE 14              LDX     ARG1    YES,REPLACE PROP. ASSOCIATED
07E6 EE 02              LDX     2,X      WITH THIS INDICATOR.
07E8 EE 02              LDX     2,X
07EA BD 077E            JSR     RPLACA
07ED EE 00              LDX     0,X     AND RETURN WITH PROPERTY.
07EF 39                 RTS

07F0 DE 14      NO.PRP  LDX     ARG1    NO SUCH INDICATOR
07F2 BD 03C7            JSR     PUSHX    ON PROPERTY-LIST
07F5 DE 16              LDX     ARG2
07F7 DF 14              STX     ARG1
07F9 DE 0E              LDX     N       NOW APPEND THE NEW PROP.
07FB DF 16              STX     ARG2     TOGETHER WITH THE IND.
07FD BD 076A            JSR     CONS      TO THE PROPERTY-LIST.
0800 DF 16              STX     ARG2
0802 DE 18              LDX     ARG3
0804 DF 14              STX     ARG1
0806 BD 076A            JSR     CONS
0809 DF 16              STX     ARG2
080B BD 03E5            JSR     PULLX
080E BD 0775            JSR     RPLACD
0811 EE 02              LDX     2,X
0813 EE 02              LDX     2,X
0815 EE 00              LDX     0,X     RETURN WITH PROP.
0817 39                 RTS
```

```
                        *
                        *  (COND (X Y)(P Q).....)
                        *
                        * IF X THEN Y ELSE IF P THEN Q ......  ELSE NIL
                        *
0818 BD 03C7    COND1   JSR     PUSHX   SAVE RUNNING POINTER ON
081B EE 00              LDX     0,X      LIST OF CONDITIONALS.
081D EE 00              LDX     0,X     GET CONDITION
081F DF 14              STX     ARG1
0821 DE 16              LDX     ARG2
0823 BD 03C7            JSR     PUSHX   SAVE CURRENT A-LIST.
0826 DE 14              LDX     ARG1
0828 BD 08B9            JSR     EVAL    AND EVALUATE
082B DF 14              STX     ARG1
082D 27 15              BEQ     NXTCND  FALSE! SO NEXT CONDITION.
082F BD 03E5            JSR     PULLX   TRUE!
0832 DF 16              STX     ARG2     RETREVE A-LIST AND RUNNING POI
0834 BD 03E5            JSR     PULLX
0837 EE 00              LDX     0,X
0839 EE 02              LDX     2,X     AND  GET ASSOCIATED EXPRESSION
083B 27 16              BEQ     RETCND   IF THERE IS ANY;
083D EE 00              LDX     0,X
083F DF 14              STX     ARG1      AND EVALUATE.
0841 7E 08B9            JMP     EVAL    THE VALUE OF 'COND' IS THAT RES
0844 BD 03E5    NXTCND  JSR     PULLX
0847 DF 16              STX     ARG2
0849 BD 03E5            JSR     PULLX
084C EE 02              LDX     2,X     GET NEXT CONDITION
084E DF 14      COND    STX     ARG1
0850 26 C6              BNE     COND1
0852 39                 RTS             RUNNING OUT OF LIST;RESULT IS N

0853 DE 14      RETCND  LDX     ARG1    NO ASSOCIATED EXPRESSION,
0855 39                 RTS              SO RETURN VALUE OF CONDITION.
```

```
                        *
                        *  (GET ATOM IND)
                        *
                        * SEARCH ON ATOMS PROPERTY LIST FOR
                        * AN INDICATOR 'IND'.
                        * WHEN FOUND RETURN THE VALUE,
                        * ASSOCIATED WITH THIS INDICATOR;
                        * OTHERWISE RETURN NIL.
                        *                          NC = Not Found
0856 EE 02      GET     LDX     2,X     (CDR ATOM)   C = Found
0858 DF 14              STX     ARG1
085A 26 01              BNE     GET2
085C 39                 RTS             EMPTY PROPERTY,RETURN NIL.
085D EE 00      GET2    LDX     0,X     GET INDICATOR
085F 9C 16              CPX     ARG2    MATCH?
0861 27 06              BEQ     GET3
0863 DE 14              LDX     ARG1    NO,NEXT INDICATOR!
0865 EE 02              LDX     2,X
0867 20 ED              BRA     GET
0869 DE 14      GET3    LDX     ARG1    YES,GET VALUE!
086B EE 02              LDX     2,X
086D EE 00              LDX     0,X
086F 39                 RTS
                        *
                        *  (SASSOC VAR ALIST)
                        *
                        * SEARCH FOR A VARABLE 'VAR' ON THE
                        * ASSOCIATION LIST 'ALIST'.
                        * WHEN FOUND,RETURN THE VARIABLE-VALUE PAIR;
                        * OTHERWISE RETURN NIL.
                        *
0870 DE 16      SASSOC  LDX     ARG2    GET ALIST
0872 20 0C              BRA     ASSOC1
0874 EE 00      LOOKUP  LDX     0,X     (CAAR ALIST)
0876 EE 00              LDX     0,X
0878 9C 14              CPX     ARG1    MATCH?
087A 27 09              BEQ     ONALST
087C DE 28              LDX     SAVEX   NO,GET NEXT PAIR!
087E EE 02              LDX     2,X
0880 DF 28      ASSOC1  STX     SAVEX   EMPTY?
0882 26 F0              BNE     LOOKUP  NO
0884 39                 RTS             YES,RETURN NIL
0885 DE 28      ONALST  LDX     SAVEX   FOUND,RETURN VARIABLE-
0887 EE 00              LDX     0,X      VALUE PAIR.
0889 39                 RTS
```

```
                        *
                        * ILLEGAL FUNCTION NAME.
                        *
088A CE 0CF8    FN.ER   LDX     #FN.MS
088D 7E 0BA3            JMP     FATAL1
                        *
                        * NO IND. ON FUNCTION NAME PROPERTY.
                        *
0890 DE 14      NO.IND  LDX     ARG1
0892 EE 02              LDX     2,X
0894 BD 03C7            JSR     PUSHX
0897 DE 14              LDX     ARG1
0899 EE 00              LDX     0,X
089B DF 14              STX     ARG1
089D 8D D1              BSR     SASSOC  LOOKUP THE A-LIST
089F 27 E9              BEQ     FN.ER    FOR THIS FUNCTION NAME
08A1 EE 02              LDX     2,X     AND TRY AGAIN.
08A3 DF 14              STX     ARG1
08A5 DE 16              LDX     ARG2
08A7 DF 18              STX     ARG3
08A9 BD 03E5            JSR     PULLX
08AC DF 16              STX     ARG2
08AE BD 076A            JSR     CONS
08B1 DF 14              STX     ARG1
08B3 DE 18              LDX     ARG3
08B5 DF 16              STX     ARG2
08B7 DE 14              LDX     ARG1
```

```
                        *
                        *   (EVAL FORM ALIST)
                        *
                        * EVALUATE FORM
                        *
08B9 DF 14      EVAL    STX     ARG1
08BB BD 029D            JSR     ABORT
08BE DE 14              LDX     ARG1
08C0 BD 06E4            JSR     NUMBER  A NUMBER IS ITSELF AS VALUE
08C3 27 03              BEQ     EVAL1
08C5 DE 14              LDX     ARG1
08C7 39                 RTS
08C8 DE 14      EVAL1   LDX     ARG1
08CA BD 06DC            JSR     ATOM    ATOMIC FORM?
08CD 27 12              BEQ     SPEC    NO,SPECIAL FORM
08CF BD 0870            JSR     SASSOC  YES,ON A-LIST?
08D2 27 03              BEQ     EVAL3
08D4 EE 02              LDX     2,X     YES,GET ITS ASSOCIATED VAL.
08D6 39         EVALEX  RTS              AND RETURN WITH IT
08D7 CE 0E04    EVAL3   LDX     #APVAL  NO,LOOKUP ATOMS PROPERTY-LIST
08DA DF 16              STX     ARG2     FOR AN APVAL-INDICATOR
08DC DE 14              LDX     ARG1      AND RETURN WITH THAT VALUE.
08DE 7E 0856            JMP     GET

08E1 DE 14      SPEC    LDX     ARG1
08E3 EE 00              LDX     0,X
08E5 BD 06DC            JSR     ATOM    ATOMIC FUNCTION NAME?
08E8 27 2F              BEQ     EVAL4   NO
08EA DE 14              LDX     ARG1
08EC EE 00              LDX     0,X
08EE BD 06E4            JSR     NUMBER  YES,IS IT A NUMBER?
08F1 26 97              BNE     FN.ER   YES,ILLEGAL FUNCTION.
08F3 DE 14              LDX     ARG1
08F5 EE 00              LDX     0,X
08F7 EE 02      EVAL6   LDX     2,X     NO,GET PROPERTY
08F9 DF 28              STX     SAVEX
08FB 27 93              BEQ     NO.IND  NIL!
08FD EE 00              LDX     0,X     GET INDICATOR
08FF 8C 0E6C            CPX     #FEXPR
0902 27 19              BEQ     EXFEX   FEXPR
0904 8C 0E54            CPX     #EXPR
0907 27 42              BEQ     EXEX    EXPR
0909 8C 0E38            CPX     #FSUBR
090C 27 62              BEQ     EXFSBR  FSUBR
090E 8C 0E20            CPX     #SUBR
0911 27 6D              BEQ     EXSBR   SUBR
0913 DE 28              LDX     SAVEX   NO SUCH INDICATOR FOUND YET,
0915 EE 02              LDX     2,X      GET NEXT IND. FROM PROPERTY!
0917 20 DE              BRA     EVAL6

0919 DE 14      EVAL4   LDX     ARG1    GO EVALUATE ARG. LIST
091B 20 32              BRA     EVAL9    AND HAND OVER TO APPLY
```

```
091D DE 28      EXFEX   LDX     SAVEX
091F BD 03C7            JSR     PUSHX
0922 DE 14              LDX     ARG1
0924 BD 03C7            JSR     PUSHX
0927 DE 16              LDX     ARG2    BUILD A LIST OF TWO ELEMENTS
0929 DF 18              STX     ARG3     THE FIRST IS THE ARG-LIST.
092B DF 14              STX     ARG1      THE SECOND THE A-LIST.
092D DE 0E              LDX     N
092F DF 16              STX     ARG2
0931 BD 076A            JSR     CONS
0934 DF 16              STX     ARG2
0936 BD 03E5            JSR     PULLX
0939 EE 02              LDX     2,X
093B DF 14              STX     ARG1
093D BD 076A            JSR     CONS
0940 DF 16              STX     ARG2
0942 BD 03E5            JSR     PULLX
0945 EE 02              LDX     2,X
0947 EE 00              LDX     0,X     GET FUNCTION; AND GO TO APPLY
0949 20 21              BRA     GO.PLY

094B DE 28      EXEX    LDX     SAVEX
094D EE 02              LDX     2,X
094F EE 00      EVAL9   LDX     0,X
0951 BD 03C7            JSR     PUSHX   PUSH FUNCTION
0954 DE 16              LDX     ARG2
0956 BD 03C7            JSR     PUSHX   PUSH CURRENT A-LIST
0959 DE 14              LDX     ARG1
095B EE 02              LDX     2,X     GET ARGUMENT LIST.
095D DF 14              STX     ARG1
095F BD 0ADC            JSR     EVLIS   AND EVALUATE!
0962 DF 16              STX     ARG2
0964 BD 03E5            JSR     PULLX   PULL UP A-LIST
0967 DF 18              STX     ARG3
0969 BD 03E5            JSR     PULLX   PULL UP FUNCTION
096C DF 14      GO.PLY  STX     ARG1    AND GO INTO APPLY
096E 20 5A              BRA     APPLY
```

```
0970 DE 28      EXFSBR  LDX     SAVEX
0972 EE 02              LDX     2,X     GET MACHINE ENTRY ADDRESS.
0974 EE 00              LDX     0,X
0976 BD 03C7            JSR     PUSHX   PUSH FOR TRICKY JUMP
0979 DE 14              LDX     ARG1    GET ARGUMENT1 FOR FUNTION
097B EE 02              LDX     2,X
097D DF 14              STX     ARG1    ARG2 = A-LIST
097F 39                 RTS             GOTO MACHINE ROUTINE

0980 DE 28      EXSBR   LDX     SAVEX
0982 EE 02              LDX     2,X     GET MACHINE ENTRY ADDRESS.
0984 EE 00              LDX     0,X
0986 BD 03C7            JSR     PUSHX   AND PUSH FOR TRICKY JUMP
0989 DE 14              LDX     ARG1
098B EE 02              LDX     2,X
098D DF 14              STX     ARG1
098F BD 0ADC            JSR     EVLIS   EVALUATE ARGUMENT.
0992 DF 12      ARGLST  STX     CURCEL  SAVE RESULT
0994 86 06              LDAA    #6
0996 CE 0014            LDX     #ARG1
0999 DF 28              STX     SAVEX
099B 6F 00      CLRARG  CLR     0,X     'NIL' ALL ARGUMENTS
099D 08                 INX
099E 4A                 DECA
099F 26 FA              BNE     CLRARG
09A1 DE 12      GETARG  LDX     CURCEL
09A3 27 22              BEQ     GSUBR   END OF ARG. LIST
09A5 A6 00              LDAA    0,X     GET ARG. FROM THIS LIST.
09A7 E6 01              LDAB    1,X
09A9 EE 02              LDX     2,X
09AB DF 12              STX     CURCEL
09AD DE 28              LDX     SAVEX
09AF 8C 001A            CPX     #ARG3+2
09B2 27 0A              BEQ     TMA.ER
09B4 A7 00              STAA    0,X
09B6 E7 01              STAB    1,X
09B8 08                 INX
09B9 08                 INX
09BA DF 28              STX     SAVEX
09BC 20 E3              BRA     GETARG
09BE CE 0CEA    TMA.ER  LDX     #TMA.MS TOO MANY ARGUMENTS!
09C1 BD 03A5            JSR     PMESSG
09C4 BD 03B9            JSR     CRLF    JMP LISP
09C7 DE 14      GSUBR   LDX     ARG1    X=ARG1 ON ENTRY
09C9 39         EX      RTS             AND GOTO MACHINE SUBR.
```

```
                        *
                        *  (APPLY FN ARGS ALIST)
                        *
                        * APPLY THE ARGUMENT TO 'FN'
                        *
09CA 27 FD      APPLY   BEQ     EX      NIL AS FUNCTION RETURNS NIL!
09CC BD 06DC            JSR     ATOM    ATOMIC FUNCTION-NAME?
09CF 27 59              BEQ     APPLY1  NO
09D1 DE 14              LDX     ARG1
09D3 BD 06E4            JSR     NUMBER  YES,IS IT A NUMBER?
09D6 26 37              BNE     FN.ER.  YES,ILLEGAL FUNCTION.
09D8 DE 14              LDX     ARG1    NO,GET ITS PROPERTY.
09DA EE 02      APPLY3  LDX     2,X
09DC 27 14              BEQ     APPLY2  ILLEGAL FUNCTION NAME
09DE DF 28              STX     SAVEX
09E0 EE 00              LDX     0,X     GET INDICATOR!
09E2 8C 0E20            CPX     #SUBR
09E5 27 2B              BEQ     EXSBR1  SUBR.
09E7 8C 0E54            CPX     #EXPR
09EA 27 34              BEQ     EXEX1   EXPR.
09EC DE 28              LDX     SAVEX
09EE EE 02              LDX     2,X     NONE OF THESE,GET NEXT IND.
09F0 20 E8              BRA     APPLY3
09F2 DE 16      APPLY2  LDX     ARG2    NOW TRY TO FIND
09F4 BD 03C7            JSR     PUSHX    FUNCTION-NAME ON A-LIST
09F7 DE 18              LDX     ARG3
09F9 DF 16              STX     ARG2
09FB DE 14              LDX     ARG1
09FD BD 0870            JSR     SASSOC
0A00 27 0D              BEQ     FN.ER.  NOT THERE,SO ERROR
0A02 EE 02              LDX     2,X     FOUND!,GET ASSOCIATED NAME
0A04 DF 14              STX     ARG1
0A06 BD 03E5            JSR     PULLX   AND TRY AGAIN
0A09 DF 16              STX     ARG2
0A0B DE 14              LDX     ARG1
0A0D 20 BB              BRA     APPLY
0A0F 7E 088A    FN.ER.  JMP     FN.ER
0A12 DE 28      EXSBR1  LDX     SAVEX
0A14 EE 02              LDX     2,X     GET MACHINE ENTRY-ADDRESS
0A16 EE 00              LDX     0,X
0A18 BD 03C7            JSR     PUSHX   AND PUSH FOR TRICKY JUMP!
0A1B DE 16              LDX     ARG2    GET ARGUMENT LIST
0A1D 7E 0992            JMP     ARGLST  AND GO EVALUATE FIRST

0A20 DE 28      EXEX1   LDX     SAVEX   EXPR INDICATOR
0A22 EE 02              LDX     2,X     GET FUNCTION LIST
0A24 EE 00              LDX     0,X
0A26 DF 14              STX     ARG1
0A28 20 A0              BRA     APPLY   AND GO INTO APPLY AGAIN
```

```
0A2A DE 14      APPLY1  LDX     ARG1
0A2C EE 00              LDX     0,X     FUNCTION IS A LIST.
0A2E 8C 0DE8            CPX     #LAMBDA
0A31 27 3C              BEQ     EVLAMB  LAMBDA FORM!
0A33 8C 0E88            CPX     #FUNARG
0A36 27 22              BEQ     EVLFNA  FUNARG FORM!
0A38 DE 16              LDX     ARG2
0A3A BD 03C7            JSR     PUSHX   NONE OF THESE;
0A3D DE 18              LDX     ARG3     SO EVALUATE FUNCTION LIST FIRST
 0A3F DF 16             STX     ARG2
0A41 BD 03C7            JSR     PUSHX
0A44 DE 14              LDX     ARG1
0A46 BD 08B9            JSR     EVAL
0A49 DF 14              STX     ARG1
0A4B BD 03E5            JSR     PULLX
0A4E DF 18              STX     ARG3
0A50 BD 03E5            JSR     PULLX
0A53 DF 16              STX     ARG2
0A55 DE 14              LDX     ARG1    AND THEN GO INTO APPLY AGAIN
0A57 7E 09CA            JMP     APPLY

0A5A DE 14      EVLFNA  LDX     ARG1
0A5C EE 02              LDX     2,X     GET OLD A-LIST
0A5E EE 02              LDX     2,X
0A60 EE 00              LDX     0,X
0A62 DF 18              STX     ARG3
0A64 DE 14              LDX     ARG1
0A66 EE 02              LDX     2,X     GET FUNCTION
0A68 EE 00              LDX     0,X
0A6A DF 14              STX     ARG1
0A6C 7E 09CA            JMP     APPLY   AND GO INTO APPLY
```

```
0A6F DE 14      EVLAMB  LDX     ARG1
0A71 BD 03C7            JSR     PUSHX   POINTER TO LAMBDA-FORM
0A74 EE 02              LDX     2,X
0A76 EE 00              LDX     0,X
0A78 DF 28              STX     SAVEX   POINTER TO LAMBDA-VARIABLES.
0A7A DE 16              LDX     ARG2    ACTUAL PARAMETERS.
0A7C 27 2A              BEQ     VAREXH
0A7E BD 03C7    PAIRLS  JSR     PUSHX   NOW ASSOCIATE LAMBDA-VAR.
0A81 EE 00              LDX     0,X      WITH ACTUALS.
0A83 DF 16              STX     ARG2
0A85 DE 28              LDX     SAVEX
0A87 27 35              BEQ     TA.ER
0A89 EE 00              LDX     0,X
0A8B DF 14              STX     ARG1    EXTEND A-LIST WITH A
0A8D BD 076A            JSR     CONS     VARIABLE-VALUE PAIR.
0A90 DF 14              STX     ARG1
0A92 DE 18              LDX     ARG3
0A94 DF 16              STX     ARG2
0A96 BD 076A            JSR     CONS
0A99 DF 18              STX     ARG3    UPDATE A-LIST POINTER.
0A9B DE 28              LDX     SAVEX
0A9D EE 02              LDX     2,X     NEXT LAMBDA  VARIABLE
0A9F DF 28              STX     SAVEX
0AA1 BD 03E5            JSR     PULLX
0AA4 EE 02              LDX     2,X     NEXT ACTUAL PARAMETER.
0AA6 26 D6              BNE     PAIRLS
0AA8 DE 28      VAREXH  LDX     SAVEX   BOTH LISTS EXHAUSTED?
0AAA 26 19              BNE     TLA.ER  NO,TOO LITTLE ARGS.
0AAC DE 18              LDX     ARG3
0AAE DF 16              STX     ARG2    GET UPDATED A-LIST
0AB0 BD 03E5            JSR     PULLX   GET FUNCTION TO BE EVALUATED
0AB3 EE 02              LDX     2,X      (2ND FORM ON LAMBDA LIST)
0AB5 EE 02              LDX     2,X
0AB7 EE 00              LDX     0,X
0AB9 DF 14              STX     ARG1    AND LET EVAL DO THE WORK
0ABB 7E 08B9            JMP     EVAL

0ABE CE 0CEA    TA.ER   LDX     #TMA.MS
0AC1 31                 INS
0AC2 31                 INS
0AC3 20 03              BRA     RECOV1
0AC5 CE 0D46    TLA.ER  LDX     #TLA.MS
0AC8 BD 03A5    RECOV1  JSR     PMESSG
0ACB CE 0D56            LDX     #FOR.MS
0ACE BD 03AF            JSR     PMESG1
0AD1 BD 03E5            JSR     PULLX
0AD4 DF 14              STX     ARG1
0AD6 BD 0607            JSR     PRINTE  PRINT FUNCTION CALL
0AD9 7E 019C            JMP     LISP
```

```
                        *
                        *  (EVLIS  LIST ALIST)
                        *
                        * EVALUATE ALL ELEMENTS OF 'LIST'
                        * THE VALUE IS A LIST OF RESULTS
                        * OF ALL THESE EVALUATIONS.
                        *
0ADC DE 14      EVLIS   LDX     ARG1
0ADE 27 32              BEQ     EVLISX  EMPTY SOURCE-LIST RETURNS NIL.
0AE0 BD 03C7            JSR     PUSHX
0AE3 EE 00              LDX     0,X     GET A FORM FROM SOURCE-LIST
0AE5 DF 14              STX     ARG1
0AE7 DE 16              LDX     ARG2
0AE9 BD 03C7            JSR     PUSHX
0AEC DE 14              LDX     ARG1
0AEE BD 08B9            JSR     EVAL    AND EVALUATE THAT FORM
0AF1 DF 18              STX     ARG3
0AF3 BD 03E5            JSR     PULLX
0AF6 DF 16              STX     ARG2
0AF8 BD 03E5            JSR     PULLX
0AFB EE 02              LDX     2,X     GET CDR OF SOURCE-LIST
0AFD DF 14              STX     ARG1
0AFF DE 18              LDX     ARG3
0B01 BD 03C7            JSR     PUSHX
0B04 DE 14              LDX     ARG1
0B06 8D D4              BSR     EVLIS   AND GO INTO EVLIS AGAIN!
0B08 DF 16              STX     ARG2
0B0A BD 03E5            JSR     PULLX
0B0D DF 14              STX     ARG1
0B0F BD 076A            JSR     CONS    CONS THE RESULT TO
0B12 39         EVLISX  RTS               A RESULT-LIST.
```

```
                        *
                        * (PROG (VAR1 VAR2....) LABEL (STATEMENT).......)
                        *
0B13 DE 1A      PROG    LDX     PROGB
0B15 BD 03C7            JSR     PUSHX
0B18 DE 1C              LDX     RUNP
0B1A BD 03C7            JSR     PUSHX
0B1D DE 14              LDX     ARG1
0B1F 27 56              BEQ     RETNIL
0B21 EE 02              LDX     2,X
0B23 27 52              BEQ     RETNIL  NO PROG-LIST
0B25 DF 1A              STX     PROGB
0B27 DE 16              LDX     ARG2
0B29 DF 18              STX     ARG3    SAVE A-LIST
0B2B DE 14              LDX     ARG1
0B2D EE 00              LDX     0,X     GET LIST OF PROG VARIABLES.
0B2F 27 20              BEQ     PROG1
0B31 BD 03C7    LOCVAR  JSR     PUSHX
0B34 EE 00              LDX     0,X     APPEND PROG VARIABLES
0B36 DF 14              STX     ARG1    TO THE CURRENT A-LIST
0B38 DE 0E              LDX     N       THE ASSOCIATED VALUE
0B3A DF 16              STX     ARG2
0B3C BD 076A            JSR     CONS     IS INITIALLY NIL.
0B3F DF 14              STX     ARG1
0B41 DE 18              LDX     ARG3
0B43 DF 16              STX     ARG2
0B45 BD 076A            JSR     CONS
0B48 DF 18              STX     ARG3    UPDATE A-LIST POINTER
0B4A BD 03E5            JSR     PULLX
0B4D EE 02              LDX     2,X
0B4F 26 E0              BNE     LOCVAR  NEXT VARIABLE
0B51 DE 18      PROG1   LDX     ARG3
0B53 DF 16              STX     ARG2    UPDATED A-LIST
0B55 DE 1A              LDX     PROGB   BEGIN OF PROG.
0B57 DF 1C      EXPROG  STX     RUNP
0B59 EE 00              LDX     0,X     GET STATEMENT FROM PROG-LIST
0B5B DF 14              STX     ARG1
0B5D BD 06DC            JSR     ATOM    ATOMIC?
0B60 26 0F              BNE     NXTST   YES,LABEL;SO SKIP.
0B62 DE 16              LDX     ARG2
0B64 BD 03C7            JSR     PUSHX
0B67 DE 14              LDX     ARG1
0B69 BD 08B9            JSR     EVAL    NO,EVALUATE STATEMENT
0B6C BD 03E5    RETPRG  JSR     PULLX
0B6F DF 16              STX     ARG2
0B71 DE 1C      NXTST   LDX     RUNP
0B73 EE 02              LDX     2,X     NEXT STATEMENT
0B75 26 E0              BNE     EXPROG
0B77 BD 03E5    RETNIL  JSR     PULLX   RUNNING OUT OF PROGRAM!
0B7A DF 1C              STX     RUNP
0B7C BD 03E5            JSR     PULLX
0B7F DF 1A              STX     PROGB
0B81 DE 0E              LDX     N        RETURN WITH VALUE NIL
0B83 39                 RTS
```

```
                        *
                        * (GO X)
                        *
                        * GO TO LABEL X (LITERAL)
                        *
0B84 DE 14      GO      LDX     ARG1
0B86 27 18              BEQ     GO.ER
0B88 EE 00              LDX     0,X
0B8A DF 14              STX     ARG1
0B8C DE 1A              LDX     PROGB   IN A PROG.?
0B8E 27 10              BEQ     GO.ER   NO,ERROR
0B90 DF 16      FNDLBL  STX     ARG2    SEARCH FOR A LABEL
0B92 EE 00              LDX     0,X
0B94 9C 14              CPX     ARG1    MATCH?
0B96 27 16              BEQ     LABEL   YES
0B98 DE 16              LDX     ARG2
0B9A EE 02              LDX     2,X     NO,NEXT ELEMENT ON PROG-LIST
0B9C DF 16              STX     ARG2
0B9E 26 F0              BNE     FNDLBL
0BA0 CE 0D21    GO.ER   LDX     #GO.MS
0BA3 BD 03A5    FATAL1  JSR     PMESSG
0BA6 DE 14              LDX     ARG1
0BA8 BD 2607            JSR     PRINTE
0BAB 7E 019C            JMP     LISP

0BAE 96 16      LABEL   LDAA    ARG2    LABEL FOUND,
0BB0 D6 17              LDAB    ARG2+1   MODIFY RUNNING POINTER
0BB2 97 1C              STAA    RUNP
0BB4 D7 1D              STAB    RUNP+1
0BB6 39                 RTS
                        *
                        *  (RETURN X)
                        *
                        * RETURN FROM PROG WITH X AS VALUE
                        *
0BB7 DE 1A      RETURN  LDX     PROGB   IN A PROG.?
0BB9 27 16              BEQ     RETVAL  NO,JUST RETURN VALUE
0BBB 30         CLRSTK  TSX
0BBC 31                 INS             YES,CLEAN UP STACK
0BBD 31                 INS              USED BY PROG.
0BBE EE 00              LDX     0,X
0BC0 8C 0B6C            CPX     #RETPRG
0BC3 26 F6              BNE     CLRSTK
0BC5 31                 INS
0BC6 31                 INS
0BC7 BD 03E5            JSR     PULLX
0BCA DF 1C              STX     RUNP
0BCC BD 03E5            JSR     PULLX
0BCF DF 1A              STX     PROGB
0BD1 DE 14      RETVAL  LDX     ARG1    RETURN EVALUATED ARG OF RETURN
0BD3 39                 RTS
```

```
                        *
                        * GET TWO NUMERIC VALUES
                        * ON EXIT A&B CONTAINS THE NUMERIC VALUE OF ARG1 DE
                        * AND X POINT TO THE NUMERIC VALUE OF ARG2. HL
                        *                                          E=MSB
                                                                   D=LSB
0BD4 EE 00      GET2N   LDX     0,X
0BD6 09                 DEX
0BD7 26 18              BNE     NN.ER   ARG1 NOT A NUMBER          DE=NUM1
0BD9 DE 16              LDX     ARG2                               HL=NUM2
0BDB EE 00              LDX     0,X
0BDD 09                 DEX
0BDE 26 0D              BNE     NN.ER2  ARG2 NOT ANUMBER
0BE0 DE 14              LDX     ARG1
0BE2 EE 02              LDX     2,X
0BE4 A6 00              LDAA    0,X     GET NUMERIC VALUE OF ARG1
0BE6 E6 01              LDAB    1,X
0BE8 DE 16              LDX     ARG2
0BEA EE 02              LDX     2,X     POINT TO NUMERIC VALUE OF ARG2
0BEC 39                 RTS
0BED DE 16      NN.ER2  LDX     ARG2
0BEF DF 14              STX     ARG1
0BF1 CE 0D2D    NN.ER   LDX     #NN.MS
0BF4 7E 0BA3            JMP     FATAL1
                        *
                        * (PLUS X Y)
                        *
                        * RESULT IS THE SUM OF X AND Y
                        *
0BF7 BD 0BD4    PLUS    JSR     GET2N
0BFA EB 01              ADDB    1,X     16-BIT ADD
0BFC A9 00              ADCA    0,X
0BFE 29 0B              BVS     OV.ER   OVERFLOW
0C00 7E 0CAB    STRES   JMP     PUTIN   STORE RESULT
                        *
                        *  (MINUS X Y)
                        *
                        * RESULT IS X - Y
                        *
0C03 8D CF      MINUS   BSR     GET2N
0C05 E0 01              SUBB    1,X
0C07 A2 00              SBCA    0,X     16-BIT SUBSTRACT
0C09 28 F5              BVC     STRES
0C0B CE 0D3E    OV.ER   LDX     #OV.MS
0C0E 36                 PSHA
0C0F 37                 PSHB
0C10 BD 03A5            JSR     PMESSG
0C13 33                 PULB
0C14 32                 PULA
0C15 7E 0CAB            JMP     PUTIN
```

```
                    * GET TWO NUMERIC VALUES
                    * AND MAKE ABSOLUTE.
                    * NUM1=ABS(ARG1)
                    * NUM2=ABS(ARG2)
                    *
0C18 8D BA          G2ABS   BSR     GET2N
0C1A 36                     PSHA
0C1B A8 00                  EORA    0,X
0C1D 97 31                  STAA    SIGN        SAVE RESULT SIGN
0C1F 32                     PULA
0C20 8D 0F                  BSR     ABS
0C22 97 24                  STAA    NUM1
0C24 D7 25                  STAB    NUM1+1
0C26 A6 00                  LDAA    0,X
0C28 E6 01                  LDAB    1,X
0C2A 8D 05                  BSR     ABS
0C2C 97 26                  STAA    NUM2
0C2E D7 27                  STAB    NUM2+1
0C30 39                     RTS

0C31 4D             ABS     TSTA
0C32 2A 05                  BPL     ABS1
0C34 43             NEG     COMA                ABSOLUTE VAL OF A & B
0C35 50                     NEGB
0C36 26 01                  BNE     ABS1
0C38 4C                     INCA
0C39 39             ABS1    RTS
                    *
                    *  (TIMES X Y)
                    *
                    * RESULT IS THE PRODUCT OF X AND Y
                    *
0C3A 8D DC          TIMES   BSR     G2ABS
0C3C CE 0011                LDX     #17         STEP COUNT
0C3F 4F                     CLRA
0C40 5F                     CLRB
0C41 20 08                  BRA     MUL3
0C43 24 04          MUL1    BCC     MUL2
0C45 DB 25                  ADDB    NUM1+1
0C47 99 24                  ADCA    NUM1
0C49 46             MUL2    RORA
0C4A 56                     RORB
0C4B 76 0026        MUL3    ROR     NUM2
0C4E 76 0027                ROR     NUM2+1
0C51 09                     DEX
0C52 26 EF                  BNE     MUL1
0C54 4D                     TSTA
0C55 26 B4                  BNE     OV.ER
0C57 5D                     TSTB
0C58 26 B1                  BNE     OV.ER       TEST FOR OVERFLOW
0C5A 96 26                  LDAA    NUM2
0C5C 2B AD                  BMI     OV.ER
0C5E D6 27                  LDAB    NUM2+1
0C60 20 43                  BRA     RSTRS
```

```
                        *
                        * (QUOTIENT X Y)
                        *
                        * RESULT IS X/Y
                        *
0C62 8D B4      DIV     BSR     G2ABS
0C64 DE 26              LDX     NUM2
0C66 27 A3              BEQ     OV.ER   DIVISION BY ZERO!
0C68 7F 0030            CLR     STEPC
0C6B 7C 0030            INC     STEPC
0C6E 7C 0030    DIVSCL  INC     STEPC
0C71 58                 ASLB            PRESCALE NUM2
0C72 49                 ROLA
0C73 2A F9              BPL     DIVSCL
0C75 97 26              STAA    NUM2
0C77 D7 27              STAB    NUM2+1
0C79 96 24              LDAA    NUM1
0C7B D6 25              LDAB    NUM1+1
0C7D 7F 0024            CLR     NUM1
0C80 7F 0025            CLR     NUM1+1
0C83 D0 27      DIV1    SUBB    NUM2+1
0C85 92 26              SBCA    NUM2
0C87 24 06              BCC     DIV2
0C89 DB 27              ADDB    NUM2+1
0C8B 99 26              ADCA    NUM2
0C8D 0C                 CLC
0C8E 9C                 FCB     $9C     SKIP
0C8F 0D         DIV2    SEC
0C90 79 0025            ROL     NUM1+1  SHIFT INTO RESULT
0C93 79 0024            ROL     NUM1
0C96 74 0026            LSR     NUM2
0C99 76 0027            ROR     NUM2+1
0C9C 7A 0030            DEC     STEPC
0C9F 26 E2              BNE     DIV1
0CA1 96 24              LDAA    NUM1
0CA3 D6 25              LDAB    NUM1+1
0CA5 7D 0031    RSTRS   TST     SIGN    RESTORE SIGN
0CA8 BD 0C32            JSR     ABS+1
```

```
                        *
                        * CREATE A NUMERIC ATOM WHOSE VALUE
                        * IS IN ACCUMULATOR A & B (A IS MSB)  IN HL
                        *
0CAB 97 24      PUTIN   STAA    NUM1    CREATE A NUMERIC ATOM
0CAD D7 25              STAB    NUM1+1   FOR THE RESULT OF AN
0CAF DE 0E              LDX     N
0CB1 DF 14              STX     ARG1
0CB3 DF 16              STX     ARG2
0CB5 BD 076A            JSR     CONS
0CB8 DF 16              STX     ARG2
0CBA BD 076A            JSR     CONS
0CBD DF 14              STX     ARG1
0CBF 6C 01              INC     1,X
0CC1 DE 16              LDX     ARG2
0CC3 96 24              LDAA    NUM1
0CC5 D6 25              LDAB    NUM1+1
0CC7 A7 00              STAA    0,X
0CC9 E7 01              STAB    1,X
0CCB DE 14              LDX     ARG1
0CCD 39                 RTS
```

AB ≡ DE

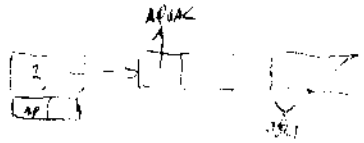
X = HL

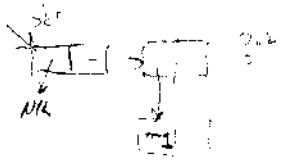
```
0CCE 53          SN.MS    FCS      /SYNTAX ERROR/
0CDA 4E          NA.MS    FCS      /NON ATOMIC ARG: /
0CEA 54          TMA.MS   FCS      /TOO MANY ARGS /
0CF8 49          FN.MS    FCS      /ILL. FUNCTION: /
0D07 41          AA.MS    FCS      /ATOMIC ARG: /
0D13 49          IN.MS    FCS      /ILLEGAL NUMBER/
0D21 49          GO.MS    FCS      /ILLEGAL GO: /
0D2D 4E          NN.MS    FCS      /NON NUMERIC ARG: /
0D3E 4F          OV.MS    FCS      /OVERFLOW/
0D46 54          TLA.MS   FCS      /TOO LITTLE ARGS /
0D56 46          FOR.MS   FCS      /FOR: /
0D5B 4D          FL.MS    FCS      /MEMORY FULL/
0D66 53          SO.MS    FCS      /STACK OVERFLOW/
0D74 49          ID.MS    FCS      /ILLEGAL DEVICE/
0D82 20          HED      FCS      /  LISP 1.5  V3A/
```

```
                    *
                    *  THE OBJECT LIST (OBLIST)
                    *
     0D94           LOC     EQU     *+3
     0D94                   ORG     LOC/4*4 EVEN WORD ADDRESSING

0D94 4E             NAMNIL  FCC     /NI/    PRINT-NAME OF ATOM 'NIL'
0D96 0D                     FDB     *+2
0D98 4C                     FCB     'L,$80
0D9A 00                     FDB     NIL
0D9C 0E             PRPNIL  FDB     APVAL,*+4       PROPERTY OF ATOM 'NIL'
0DA0 00                     FDB     NIL,*+4
0DA4 0E                     FDB     FSUBR,*+4
0DA8 06                     FDB     FALSE,NIL

0DAC 0D             OBLIST  FDB     *+12+M,*+4
0DB0 0E                     FDB     APVAL,*+4
0DB4 0D                     FDB     OBL1,NIL
0DB8 4F                     FCC     /OB/
0DBA 0D                     FDB     *+2
0DBC 4C                     FCC     /LI/
0DBE 0D                     FDB     *+2
0DC0 53                     FCC     /ST/
0DC2 00                     FDB     NIL
0DC4 00             OBL1    FDB     NIL,*+4
0DC8 0D                     FDB     *+4,OBL2
0DCC 0D             T       FDB     T.PNAM+M,*+4
0DD0 0E                     FDB     APVAL,*+4
0DD4 0D                     FDB     T,*+4
0DD8 0E                     FDB     SUBR,*+4
0DDC 09                     FDB     EX,NIL
0DE0 54             T.PNAM  FCB     'T,F
0DE2 00                     FDB     NIL
     0DE4           OBL2    EQU     *
0DE4 0D                     FDB     *+4,OBL3
0DE8 0D             LAMBDA  FDB     *+12+M,*+4
0DEC 0E                     FDB     APVAL,*+4
0DF0 0D                     FDB     *-8,NIL
0DF4 4C                     FCC     /LA/
0DF6 0D                     FDB     *+2
0DF8 4D                     FCC     /MB/
0DFA 0D                     FDB     *+2
0DFC 44                     FCC     /DA/
0DFE 00                     FDB     NIL
     0E00           OBL3    EQU     *
0E00 0E                     FDB     *+4,OBL4
0E04 0E             APVAL   FDB     *+12+M,*+4
0E08 0E                     FDB     APVAL,*+4
0E0C 0E                     FDB     *-8,NIL
0E10 41                     FCC     /AP/
0E12 0E                     FDB     *+2
0E14 56                     FCC     /VA/
0E16 0E                     FDB     *+2
0E18 4C                     FCB     'L,F
0E1A 00                     FDB     NIL
```





```
      0E1C        OBL4    EQU     *
0E1C  0E                  FDB     *+4,OBL5
0E20  0E          SUBR    FDB     *+12+M,*+4
0E24  0E                  FDB     APVAL,*+4
0E28  0E                  FDB     *-8,NIL
0E2C  53                  FCC     /SU/
0E2E  0E                  FDB     *+2
0E30  42                  FCC     /BR/
0E32  00                  FDB     NIL
      0E34        OBL5    EQU     *
0E34  0E                  FDB     *+4,OBL6
0E38  0E          FSUBR   FDB     *+12+M,*+4
0E3C  0E                  FDB     APVAL,*+4
0E40  0E                  FDB     *-8,NIL
0E44  46                  FCC     /FS/
0E46  0E                  FDB     *+2
0E48  55                  FCC     /UB/
0E4A  0E                  FDB     *+2
0E4C  52                  FCB     'R,F
0E4E  00                  FDB     NIL
      0E50        OBL6    EQU     *
0E50  0E                  FDB     *+4,OBL7
0E54  0E          EXPR    FDB     *+12+M,*+4
0E58  0E                  FDB     APVAL,*+4
0E5C  0E                  FDB     *-8,NIL
0E60  45                  FCC     /EX/
0E62  0E                  FDB     *+2
0E64  50                  FCC     /PR/
0E66  00                  FDB     NIL
      0E68        OBL7    EQU     *
0E68  0E                  FDB     *+4,OBL8
0E6C  0E          FEXPR   FDB     *+12+M,*+4
0E70  0E                  FDB     APVAL,*+4
0E74  0E                  FDB     *-8,NIL
0E78  46                  FCC     /FE/
0E7A  0E                  FDB     *+2
0E7C  58                  FCC     /XP/
0E7E  0E                  FDB     *+2
0E80  52                  FCB     'R,F
0E82  00                  FDB     NIL
      0E84        OBL8    EQU     *
0E84  0E                  FDB     *+4,OBL90
0E88  0E          FUNARG  FDB     *+12+M,*+4
0E8C  0E                  FDB     APVAL,*+4
0E90  0E                  FDB     *-8,NIL
0E94  46                  FCC     /FU/
0E96  0E                  FDB     *+2
0E98  4E                  FCC     /NA/
0E9A  0E                  FDB     *+2
0E9C  52                  FCC     /RG/
0E9E  00                  FDB     NIL
      0EA0        OBL90   EQU     *

0EA0  0E                  FDB     *+4,OBL9
0EA4  0E                  FDB     *+12+M,*+4
```

```
0EA8 0E                    FDB     SUBR,*+4
0EAC 06                    FDB     ATOM,NIL
0EB0 41                    FCC     /AT/
0EB2 0E                    FDB     *+2
0EB4 4F                    FCC     /OM/
0EB6 00                    FDB     NIL
     0EB8          OBL9    EQU     *
0EB8 0E                    FDB     *+4,OBL10
0EBC 0E                    FDB     *+12+M,*+4
0EC0 0E                    FDB     SUBR,*+4
0EC4 07                    FDB     CAR,NIL
0EC8 43                    FCC     /CA/
0ECA 0E                    FDB     *+2
0ECC 52                    FCB     'R,F
0ECE 00                    FDB     NIL
     0ED0          OBL10   EQU     *
0ED0 0E                    FDB     *+4,OBL11
0ED4 0E                    FDB     *+12+M,*+4
0ED8 0E                    FDB     SUBR,*+4
0EDC 07                    FDB     CDR,NIL
0EE0 43                    FCC     /CD/
0EE2 0E                    FDB     *+2
0EE4 52                    FCB     'R,F
0EE6 00                    FDB     NIL
     0EE8          OBL11   EQU     *
0EE8 0E                    FDB     *+4,OBL12
0EEC 0E                    FDB     *+12+M,*+4
0EF0 0E                    FDB     SUBR,*+4
0EF4 07                    FDB     CONS,NIL
0EF8 43                    FCC     /CO/
0EFA 0E                    FDB     *+2
0EFC 4E                    FCC     /NS/
0EFE 00                    FDB     NIL
     0F20          OBL12   EQU     *
0F00 0F                    FDB     *+4,OBL13
0F04 0F                    FDB     *+12+M,*+4
0F08 0E                    FDB     SUBR,*+4
0F0C 06                    FDB     NULL,NIL
0F10 4E                    FCC     /NU/
0F12 0F                    FDB     *+2
0F14 4C                    FCC     /LL/
0F16 00                    FDB     NIL
     0F18          OBL13   EQU     *
0F18 2F                    FDB     *+4,OBL14
0F1C 0F                    FDB     *+12+M,*+4
0F20 0E                    FDB     SUBR,*+4
0F24 07                    FDB     EQ,NIL
0F28 45                    FCC     /EQ/
0F2A 00                    FDB     NIL
     0F2C          OBL14   EQU     *
0F2C 0F                    FDB     *+4,OBL15
0F30 0F                    FDB     *+12+M,*+4
0F34 0E                    FDB     SUBR,*+4
0F38 06                    FDB     NUMBER,NIL
0F3C 4E                    FCC     /NU/
```

```
0F3E 0F                    FDB     *+2
0F40 4D                    FCC     /MB/
0F42 0F                    FDB     *+2
0F44 45                    FCC     /ER/
0F46 00                    FDB     NIL
     0F48          OBL15   EQU     *
0F48 0F                    FDB     *+4,OBL16
0F4C 0F                    FDB     *+12+M,*+4
0F50 0E                    FDB     SUBR,*+4
0F54 07                    FDB     GREATR,NIL
0F58 47                    FCC     /GR/
0F5A 0F                    FDB     *+2
0F5C 45                    FCC     /EA/
0F5E 0F                    FDB     *+2
0F60 54                    FCC     /TE/
0F62 0F                    FDB     *+2
0F64 52                    FCC     /RP/
0F66 00                    FDB     NIL
     0F68          OBL16   EQU     *
0F68 0F                    FDB     *+4,OBL17
0F6C 0F                    FDB     *+12+M,*+4
0F70 0E                    FDB     SUBR,*+4
0F74 04                    FDB     READ,NIL
0F78 52                    FCC     /RE/
0F7A 0F                    FDB     *+2
0F7C 41                    FCC     /AD/
0F7E 00                    FDB     NIL
     0F80          OBL17   EQU     *
0F80 0F                    FDB     *+4,OBL18
0F84 0F                    FDB     *+12+M,*+4
0F88 0E                    FDB     SUBR,*+4
0F8C 06                    FDB     PRINT,NIL
0F90 50                    FCC     /PR/
0F92 0F                    FDB     *+2
0F94 49                    FCC     /IN/
0F96 0F                    FDB     *+2
0F98 54                    FCB     'T,F
0F9A 00                    FDB     NIL
     0F9C          OBL18   EQU     *
0F9C 0F                    FDB     *+4,OBL19
0FA0 0F                    FDB     *+12+M,*+4
0FA4 0E                    FDB     SUBR,*+4
0FA8 06                    FDB     PRINI,NIL
0FAC 50                    FCC     /PR/
0FAE 0F                    FDB     *+2
0FB0 49                    FCC     /IN/
0FB2 0F                    FDB     *+2
0FB4 31                    FCB     '1,F
0FB6 00                    FDB     NIL
     0FB8          OBL19   EQU     *
0FB8 0F                    FDB     *+4,OBL20
0FBC 0F                    FDB     *+12+M,*+4
0FC0 0E                    FDB     SUBR,*+4
0FC4 06                    FDB     TERPRI,NIL
0FC8 54                    FCC     /TE/
```

```
0FCA 0F                        FDB     *+2
0FCC 52                        FCC     /RP/
0FCE 0F                        FDB     *+2
0FD0 52                        FCC     /RI/
0FD2 00                        FDB     NIL
     0FD4              OBL20   EQU     *
0FD4 0F                        FDB     *+4,OBL21
0FD8 0F                        FDB     *+12+M,*+4
0FDC 0E                        FDB     SUBR,*+4
0FE0 07                        FDB     RPLACA,NIL
0FE4 52                        FCC     /RP/
0FE6 0F                        FDB     *+2
0FE8 4C                        FCC     /LA/
0FEA 0F                        FDB     *+2
0FEC 43                        FCC     /CA/
0FEE 00                        FDB     NIL
     0FF0              OBL21   EQU     *
0FF0 0F                        FDB     *+4,OBL22
0FF4 10                        FDB     *+12+M,*+4
0FF8 0E                        FDB     SUBR,*+4
0FFC 07                        FDB     RPLACD,NIL
1000 52                        FCC     /RP/
1002 10                        FDB     *+2
1004 4C                        FCC     /LA/
1006 10                        FDB     *+2
1008 43                        FCC     /CD/
100A 00                        FDB     NIL
     100C              OBL22   EQU     *
100C 10                        FDB     *+4,OBL23
1010 10                        FDB     *+12+M,*+4
1014 0E                        FDB     SUBR,*+4
1018 08                        FDB     EVAL,NIL
101C 45                        FCC     /EV/
101E 10                        FDB     *+2
1020 41                        FCC     /AL/
1022 00                        FDB     NIL
     1024              OBL23   EQU     *
1024 10                        FDB     *+4,OBL24
1028 10                        FDB     *+12+M,*+4
102C 0E                        FDB     SUBR,*+4
1030 09                        FDB     APPLY,NIL
1034 41                        FCC     /AP/
1036 10                        FDB     *+2
1038 50                        FCC     /PL/
103A 10                        FDB     *+2
103C 59                        FCB     'Y,F
103E 00                        FDB     NIL
     1040              OBL24   EQU     *
1040 10                        FDB     *+4,OBL25
1044 10                        FDB     *+12+M,*+4
1048 0E                        FDB     SUBR,*+4
104C 08                        FDB     SASSOC,NIL
1050 53                        FCC     /SA/
1052 10                        FDB     *+2
1054 53                        FCC     /SS/
```

```
1056 10                    FDB     *+2
1058 4F                    FCC     /OC/
105A 00                    FDB     NIL
     105C          OBL25   EQU     *
105C 10                    FDB     *+4,OBL26
1060 10                    FDB     *+12+M,*+4
1064 0E                    FDB     SUBR,*+4
1068 08                    FDB     GET,NIL
106C 47                    FCC     /GE/
106E 10                    FDB     *+2
1070 54                    FCB     'T,F
1072 00                    FDB     NIL
     1074          OBL26   EQU     *
1074 10                    FDB     *+4,OBL27
1078 10                    FDB     *+12+M,*+4
107C 0E                    FDB     SUBR,*+4
1080 0B                    FDB     RETURN,NIL
1084 52                    FCC     /RE/
1086 10                    FDB     *+2
1088 54                    FCC     /TU/
108A 10                    FDB     *+2
108C 52                    FCC     /RN/
108E 00                    FDB     NIL
     1090          OBL27   EQU     *
1090 10                    FDB     *+4,OBL28
1094 10                    FDB     *+12+M,*+4
1098 0E                    FDB     SUBR,*+4
109C 06                    FDB     TEREAD,NIL
10A0 54                    FCC     /TE/
10A2 10                    FDB     *+2
10A4 52                    FCC     /RE/
10A6 10                    FDB     *+2
10A8 41                    FCC     /AD/
10AA 00                    FDB     NIL
     10AC          OBL28   EQU     *
10AC 10                    FDB     *+4,OBL29
10B0 10                    FDB     *+12+M,*+4
10B4 0E                    FDB     SUBR,*+4
10B8 0B                    FDB     PLUS,NIL
10BC 50                    FCC     /PL/
10BE 10                    FDB     *+2
10C0 55                    FCC     /US/
10C2 00                    FDB     NIL
     10C4          OBL29   EQU     *
10C4 10                    FDB     *+4,OBL30
10C8 10                    FDB     *+12+M,*+4
10CC 0E                    FDB     SUBR,*+4
10D0 0C                    FDB     MINUS,NIL
10D4 4D                    FCC     /MI/
10D6 10                    FDB     *+2
10D8 4E                    FCC     /NU/
10DA 10                    FDB     *+2
10DC 53                    FCB     'S,F
10DE 00                    FDB     NIL
     10E0          OBL30   EQU     *
```

```
10E0 10                     FDB     *+4,OBL31
10E4 10                     FDB     *+12+M,*+4
10E8 0E                     FDB     SUBR,*+4
10EC 0C                     FDB     TIMES,NIL
10F0 54                     FCC     /TI/
10F2 10                     FDB     *+2
10F4 4D                     FCC     /ME/
10F6 10                     FDB     *+2
10F8 53                     FCB     'S,F
10FA 00                     FDB     NIL
     10FC           OBL31   EQU     *
10FC 11                     FDB     *+4,OBL32
1100 11                     FDB     *+12+M,*+4
1104 0E                     FDB     SUBR,*+4
1108 0C                     FDB     DIV,NIL
110C 51                     FCC     /QU/
110E 11                     FDB     *+2
1110 4F                     FCC     /OT/
1112 11                     FDB     *+2
1114 49                     FCC     /IE/
1116 11                     FDB     *+2
1118 4E                     FCC     /NT/
111A 00                     FDB     NIL
     111C           OBL32   EQU     *
111C 11                     FDB     *+4,OBL33
1120 11                     FDB     *+12+M,*+4
1124 0E                     FDB     SUBR,*+4
1128 04                     FDB     READCH,NIL
112C 52                     FCC     /RE/
112E 11                     FDB     *+2
1130 41                     FCC     /AD/
1132 11                     FDB     *+2
1134 43                     FCC     /CH/
1136 00                     FDB     NIL
     1138           OBL33   EQU     *
1138 11                     FDB     *+4,OBL34
113C 11                     FDB     *+12+M,*+4
1140 0E                     FDB     SUBR,*+4
1144 04                     FDB     READ1,NIL
1148 52                     FCC     /RE/
114A 11                     FDB     *+2
114C 41                     FCC     /AD/
114E 11                     FDB     *+2
1150 31                     FCB     '1,F
1152 00                     FDB     NIL
     1154           OBL34   EQU     *
1154 11                     FDB     *+4,OBL35
1158 11                     FDB     *+12+M,*+4
115C 0E                     FDB     SUBR,*+4
1160 05                     FDB     OPEN,NIL
1164 4F                     FCC     /OP/
1166 11                     FDB     *+2
1168 45                     FCC     /EN/
116A 00                     FDB     NIL
     116C           OBL35   EQU     *
```

```
1168 11                     FDB     *+4,OBL36
116A 11                     FDB     *+12+M,*+4
116C 0E                     FDB     SUBR,*+4
116E 05                     FDB     CLOSE,NIL
1170 43                     FCC     /CL/
1172 11                     FDB     *+2
1174 4F                     FCC     /OS/
1176 11                     FDB     *+2
1178 45                     FCB     'E,F
117A 00                     FDB     NIL
     1188           OBL36   EQU     *
1188 11                     FDB     *+4,OBL37
118A 11                     FDB     *+12+M,*+4
118C 0E                     FDB     SUBR,*+4
118E 07                     FDB     PUTPRP,NIL
1190 50                     FCC     /PU/
1192 11                     FDB     *+2
1194 54                     FCC     /TP/
1196 11                     FDB     *+2
1198 52                     FCC     /RO/
119A 11                     FDB     *+2
119C 50                     FCB     'P,F
119E 00                     FDB     NIL
     11A8           OBL37   EQU     *

11A8 11                     FDB     *+4,OBL40
11AA 11                     FDB     *+12+M,*+4
11AC 0E                     FDB     FSUBR,*+4
11AE 07                     FDB     QUOTE,NIL
11B0 51                     FCC     /QU/
11B2 11                     FDB     *+2
11B4 4F                     FCC     /OT/
11B6 11                     FDB     *+2
11B8 45                     FCB     'E,F
11BA 00                     FDB     NIL
     11C4           OBL40   EQU     *
11C4 11                     FDB     *+4,OBL41
11C6 11                     FDB     *+12+M,*+4
11C8 0E                     FDB     FSUBR,*+4
11CA 08                     FDB     COND,NIL
11CC 43                     FCC     /CO/
11CE 11                     FDB     *+2
11D0 4E                     FCC     /ND/
11D2 00                     FDB     NIL
     11DC           OBL41   EQU     *
11DC 11                     FDB     *+4,OBL42
11DE 11                     FDB     *+12+M,*+4
11E0 0E                     FDB     FSUBR,*+4
11E2 0A                     FDB     EVLIS,NIL
11E4 4C                     FCC     /LI/
11E6 11                     FDB     *+2
11E8 53                     FCC     /ST/
11EA 00                     FDB     NIL
     11F4           OBL42   EQU     *
11F4 11                     FDB     *+4,OBL43
```

```
11F8 12                      FDB     *+12+M,*+4
11FC 0E                      FDB     FSUBR,*+4
1200 0B                      FDB     PROG,NIL
1204 50                      FCC     /PR/
1206 12                      FDB     *+2
1208 4F                      FCC     /OG/
120A 00                      FDB     NIL
     120C            OBL43   EQU     *
120C 12                      FDB     *+4,OBL44
1210 12                      FDB     *+12+M,*+4
1214 0E                      FDB     FSUBR,*+4
1218 07                      FDB     SETQ,NIL
121C 53                      FCC     /SE/
121E 12                      FDB     *+2
1220 54                      FCC     /TQ/
1222 00                      FDB     NIL
     1224            OBL44   EQU     *
1224 12                      FDB     *+4,OBL45
1228 12                      FDB     *+12+M,*+4
122C 0E                      FDB     FSUBR,*+4
1230 07                      FDB     ALIST,NIL
1234 41                      FCC     /AL/
1236 12                      FDB     *+2
1238 49                      FCC     /IS/
123A 12                      FDB     *+2
123C 54                      FCB     'T,F
123E 00                      FDB     NIL
     1240            OBL45   EQU     *
1240 12                      FDB     *+4,OBL46
1244 12                      FDB     *+12+M,*+4
1248 0E                      FDB     FSUBR,*+4
124C 07                      FDB     FUNCTI,NIL
1250 46                      FCC     /FU/
1252 12                      FDB     *+2
1254 4E                      FCC     /NC/
1256 12                      FDB     *+2
1258 54                      FCC     /TI/
125A 12                      FDB     *+2
125C 4F                      FCC     /ON/
125E 00                      FDB     NIL
     1260            OBL46   EQU     *
1260 12                      FDB     *+4,OBL47
1264 12                      FDB     *+12+M,*+4
1268 0E                      FDB     FSUBR,*+4
126C 0B                      FDB     GO,NIL
1270 47                      FCC     /GO/
1272 00                      FDB     NIL
     1274            OBL47   EQU     *
1274 0D                      FDB     OBLIST,NIL

     1278            LISPSP  EQU     *
                             END
00000  ERRORS
```

AA.ER	0761	AA.MS	0D07	AATOM	0547
ABORT	029D	ABS	0C31	ABS1	0C39
ACIACS	FF00	ACIADA	FF01	ALIST	0756
AP	0022	APPLY	09CA	APPLY1	0A2A
APPLY2	09F2	APPLY3	09DA	APVAL	0E04
ARG1	0014	ARG2	0016	ARG3	0018
ARG?	07CA	ARGLST	0992	ASSIGN	03F7
ASSOC1	0880	ATOM	06DC	BLDATM	05B7
BLDNMB	0514	BQ	052B	CAR	0759
CCOUNT	0037	CDR	0767	CELL	01C1
CLOSE	05F9	CLRARG	099B	CLRLOC	0187
CLRSTK	0BBB	COND	084E	COND1	0818
CONS	076A	CP	0032	CR	0339
CRLF	03B9	CURCEL	0012	DEC1	0692
DEC2	0695	DEC3	06AE	DEC4	06B5
DECTBL	06C5	DEL	0349	DELFLG	0034
DELINE	02F8	DEV1	040E	DEVICE	0035
DEVTBL	0106	DIV	0C62	DIV1	0C83
DIV2	0C8F	DIVSCL	0C6E	DONE	0671
DOT	047B	DUMMY	02C5	END	0151
ENDATM	04C0	EQ	0702	EVAL	08B9
EVAL1	08C8	EVAL3	08D7	EVAL4	0919
EVAL6	08F7	EVAL9	094F	EVALEX	08D6
EVLAMB	0A6F	EVLFNA	0A5A	EVLIS	0ADC
EVLISX	0B12	EX	09C9	EXEX	094B
EXEX1	0A20	EXFEX	091D	EXFSBR	0970
EXPR	0E54	EXPROG	0B57	EXSBR	0980
EXSBR1	0A12	F	0080	FALSE	06E1
FATAL	044E	FATAL1	0BA3	FEXPR	0E6C
FIRSTC	0038	FL.MS	0D5B	FN.ER	088A
FN.ER.	0A0F	FN.MS	0CF8	FNDLBL	0B90
FOR.MS	0D56	FREE	0010	FSUBR	0E38
FULL	0232	FUNARG	0E88	FUNCTI	0737
FWAM	0004	G2ABS	0C18	GCOL	01D2
GCOL1	01E8	GCOL2	01FD	GCOL3	0204
GET	0856	GET2	085D	GET2N	0BD4
GET3	0869	GETARG	09A1	GETC	0364
GETCS	0429	GETIND	07D0	GO	0B84
GO.ER	0BA0	GO.MS	0D21	GO.PLY	096C
GREATR	0728	GSUBR	09C7	HED	0D82
ID.ER	0423	ID.MS	0D74	IN.ER	051F
IN.MS	0D13	INCH	039A	INPSTR	0300
LABEL	0BAE	LAMBDA	0DE8	LASTC	00FF
LIMIT	000A	LISP	019C	LISPSP	1278
LOC	0D94	LOCVAR	0B31	LOOKUP	0874
LWAM	0006	M	0001	MARK1	0240
MARK2	0272	MARKEX	027E	MARKL	023B
MATCH	0588	MINUS	0C03	MP	002C
MRKATM	0266	MRKNAM	0262	MRKNUM	026B
MUL1	0C43	MUL2	0C49	MUL3	0C4B
N	000E	NA.ER	06CF	NA.MS	0CDA
NAMNIL	0D94	NATOM	04E1	NEG	0C34
NEXTC	030D	NEXTCF	030A	NIL	0000
NN.ER	0BF1	NN.ER2	0BED	NN.MS	0D2D
NO.ARG	0727	NO.IND	0892	NO.PRP	07F0

NOMTCH	0591	NORML	0329	NOSP	037F
NULL	06E7	NUM1	0024	NUM2	0026
NUMBER	06E4	NXT2C	055F	NXTCND	0844
NXTDEV	0411	NXTDIG	04E9	NXTOBJ	0552
NXTST	0B71	OBL1	0DC4	OBL10	0ED0
OBL11	0EE8	OBL12	0F00	OBL13	0F18
OBL14	0F2C	OBL15	0F48	OBL16	0F68
OBL17	0F80	OBL18	0F9C	OBL19	0FB8
OBL2	0DE4	OBL20	0FD4	OBL21	0FF0
OBL22	100C	OBL23	1024	OBL24	1040
OBL25	105C	OBL26	1074	OBL27	1090
OBL28	10AC	OBL29	10C4	OBL3	0E00
OBL30	10E0	OBL31	10FC	OBL32	111C
OBL33	1138	OBL34	1154	OBL35	116C
OBL36	1188	OBL37	11A8	OBL4	0E1C
OBL40	11C4	OBL41	11DC	OBL42	11F4
OBL43	122C	OBL44	1224	OBL45	1240
OBL46	1260	OBL47	1274	OBL5	0E34
OBL6	0E50	OBL7	0E68	OBL8	0E84
OBL9	0EB8	OBL90	0EA0	OBLIST	0DAC
OBLSTB	000C	OBLSTP	001E	ODD	05C1
ONALST	0885	OP	0020	OPEN	05F0
OUTCH	0380	OUTCH1	038F	OV.ER	0C0B
OV.MS	0D3E	PAIRLS	0A7E	PLUS	0BF7
PMESG1	03AF	PMESSG	03A5	PNAME	002E
PNTARG	06D2	POS	068C	POSN	051C
PRCHAR	0665	PRIN1	0651	PRINIE	0656
PRINT	0602	PRINT2	060A	PRINT3	0615
PRINT4	061B	PRINT5	0639	PRINT6	0640
PRINT7	0643	PRINTE	0607	PRNMB	0676
PROG	0B13	PROG1	0B51	PROGB	001A
PRPNIL	0D9C	PULLX	03E5	PUN	FF12
PUNC	FF13	PUNCLS	02EA	PUNOPN	22DF
PUNOUT	02D6	PUSHX	03C7	PUTIN	0CAB
PUTBAK	0431	PUTPRP	07D8	QUOTE	0721
RDATOM	04AD	RDLIST	045B	RDR	FF10
RDRC	FF11	RDRIN	02B0	RDROPN	02C6
READ	0437	READ1	0498	READ1E	049B
READCH	0488	READE	0439	RECOV1	0AC8
RETCND	0853	RETNIL	0B77	RETPRG	0B6C
RETURN	0BB7	RETVAL	0BD1	RPLACA	077E
RPLACD	0775	RSTART	0182	RSTR	0173
RSTRS	0CA5	RUNP	001C	S.END	047A
S.EXPR	0454	SASSOC	0870	SAVEX	0028
SEEK	0146	SETAPV	07BF	SETQ	0787
SHWDEL	035E	SIGN	0031	SN.ER	044B
SN.ER1	0495	SN.MS	0CCE	SO.MS	0D66
SP	037D	SPEC	08E1	SQUOTE	0525
STACK	0008	START	0130	STEPC	0030
STKOVF	03DF	STRES	0C00	SUBR	0E20
SWEEP	020A	SWPDON	022B	SYSERR	0235
T	0DCC	T.PNAM	0DE0	TA.ER	0ABE
TEMPX	002A	TEREAD	06F7	TERPRI	06EF
TIMES	0C3A	TLA.ER	0AC5	TLA.MS	0D46
TMA.ER	09BE	TMA.MS	0CEA	TR	02AD

TRAIL	02EE	TRUE	06EB	TTYIN	027F
TTYOPN	0296	TTYOUT	028B	VAREXH	0AA8
X8	0413				

```
$DUPLEX,0;
$CW,<[,]>;[DEF,FILL,  ][DEF,COND,<[#1,#1,[DEF,#1,<        FDB     *+2
        FCB     '#1,F
>][UPDATE,FILL,#1 ][DEF,[VAL,FILL],<        FDB     *+2
        FCC     /#1/
>][DEF,,]]>]
[DEF,L,<        FDB     *+4,OBL#1
        FDB     *+12+M,*+4
        FDB     #3,*+4
        FDB     #2,NIL
        FCC     /#4/
[COND,#5][COND,#6][COND,#7]         FDB     NIL
OBL#1   EQU     *>]
[DEF,S,<[L,#1,#2,SUBR,#3,#4,#5,#6]>]
[DEF,F,<[L,#1,#2,FSUBR,#3,#4,#5,#6]>]
[DEF,I,<        FDB     *+4,OBL#1
#2  FDB     *+12+M,*+4
        FDB     APVAL,*+4
        FDB     *-8,NIL
        FCC     /#3/
[COND,#4][COND,#5][COND,#6]         FDB     NIL
OBL#1   EQU     *>]
OBLIST  FDB     *+12+M,*+4
        FDB     APVAL,*+4
        FDB     OBL1,NIL
        FCC     /OB/
        FDB     *+2
        FCC     /LI/
        FDB     *+2
        FCC     /ST/
        FDB     NIL
OBL1    FDB     NIL,*+4
[I,2 ,T     ,T$]
[I,3 ,LAMBDA,LA,MB,DA]
[I,4 ,APVAL,AP,VA,L]
[I,5 ,SUBR ,SU,BR]
[I,6 ,FSUBR,FS,UB,R]
[I,7 ,EXPR ,EX,PR]
[I,8 ,FEXPR,FE,XP,R]
[I,90,FUNARG,FU,NA,RG]
        SPC 1
[S,9 ,ATOM,AT,OM]
[S,10,CAR,CA,R]
[S,11,CDR,CD,R]
[S,12,CONS,CO,NS]
[S,13,NULL,NU,LL]
[S,14,EQ,EQ]
[S,15,NUMBER,NU,MB,ER]
[S,16,GREATR,GR,EA,TE,RP]
[S,17,READ,RE,AD]
[S,18,PRINT,PR,IN,T]
[S,19,PRIN1,PR,IN,1]
[S,20,TERPRI,TE,RP,RI]
[S,21,RPLACA,RP,LA,CA]
[S,22,RPLACD,RP,LA,CD]
[S,23,EVAL,EV,AL]
[S,24,APPLY,AP,PL,Y]
[S,25,SASSOC,SA,SS,OC]
[S,26,GET,GE,T]
[S,27,RETURN,RE,TU,RN]
[S,28,TEREAD,TE,RE,AD]
[S,29,PLUS,PL,US]
[S,30,MINUS,MI,NU,S]
```