```
// JOB                                            LISP 1.6   G STEELE   2/14/72
// ASM
*MACLIB LMACS
*OVERFLOW SECTORS 0,8,0
*PRINT SYMBOL TABLE
                          ABS
                          ORG     /01FE
                          LIST
                    NIL   EQU     0
                    NO    EQU     0
                    YES   EQU     1
                    **************************************************
                    *     EQUIVALENCES FOR RESIDENT MONITOR          *
                    **************************************************
                    $PRET EQU     /28       PRE-OPERATIVE ERROR TRAP
                    $ULET EQU     /2D       TABLE OF LET DISK ADRS
                    $IOCT EQU     /32       I/O PENDING INTERRUPT COUNT
                    $EXIT EQU     /38       BRANCH ADR FOR CALL EXIT
                    $DUMP EQU     /3F       BRANCH LOC FOR CORE DUMP
                    $KCSW EQU     /7C       KEYBOARD (// TYP) SWITCH
                    $FPAD EQU     /95       TABLE OF WS DISK ADRS
                    $XR3X EQU     /E4       XR3 SETTING FOR LIBF VECTOR
                    $DBSY EQU     /EE       DISK I/O BUSY SWITCH
                    DISKZ EQU     /F2       DISK I/O SUBROUTINE
                    $ZEND EQU     /01E0
                          LIST
```

```
      HDNG   01 SPECIAL INDEX REGISTER 3 AREA
**************************************************
*     SPECIAL AREA - ADDRESSABLE THROUGH XR3     *
**************************************************
X     EQU    *+128     XR3 NORMALLY HAS THIS VALUE
**************************************************
*     TRANSFER VECTOR FOR SUBROUTINES            *
**************************************************
PUSHJ DC     *-*       PUSH JUMP
      BSC  L PSHJ1
POPJ  DC     *-*       POP JUMP
      BSC  L POPJ1
PUSHA DC     *-*       PUSH ACC
      BSC  L PSHA1
POPA  DC     *-*       POP ACC
      BSC  L POPA1
PUSHS DC     *-*       SPEC PDL PUSH
      BSC  L PSHS1
POPS  DC     *-*       SPEC PDL POP
      BSC  L POPS1
POPN  DC     *-*       SPEC PDL POP N ENTRIES
      BSC  L POPN1
XCAR  DC     *-*       CAR XSUBR
      BSC  L XCAR1
XCDR  DC     *-*       CDR XSUBR
      BSC  L XCDR1
XATOM DC     *-*       ATOM XSUBR
      BSC  L XATM1
XNULL DC     *-*       NULL XSUBR
      BSC  L XNUL1
XNMBP DC     *-*       NUMBERP XSUBR
      BSC  L XNMP1
ERROR DC     *-*       ERROR SUBROUTINE
      BSC  L ERR01
XCONS DC     *-*       CONS XSUBR
      BSC  L XCNS1
MKFXN DC     *-*       MAKE FIXED NUMBER
      BSC  L MKFX1
XNCHK DC     *-*       NUMBER CHECK
      BSC  L XNCH1
XSTRP DC     *-*       STRINGP XSUBR
      BSC  L XSTR1
XSCHK DC     *-*       STRING CHECK
      BSC  L XSCH1
      EJCT
**************************************************
*     COMMONLY USEFUL VARIABLES                  *
**************************************************
@TMPA DC      *-*       'WASTEBASKETS' (USED BY
@TMPB DC      *-*        VARIOUS SUBROUTINES FOR
@TMPC DC      *-*        TEMPORARY STORAGE)
@TRUE DC      #T        ADR OF ATOM T
@SPDL DC      S@SPD     SPEC PDL POINTER
@TMLS DC      $TMLS     LIST OF GC-PROTECTED ADRS
@ARG1 DC      *-*       *    ARGUMENTS
@ARG2 DC      *-*       **   FOR
@ARG3 DC      *-*       ***  MACHINE
@ARG4 DC      *-*       **** LANGUAGE
@ARG5 DC      *-*       **** FUNCTIONS
@ARG6 DC      *-*       ***  ARE
@ARG7 DC      *-*       **   PLACED
@ARG8 DC      *-*       *    HERE
@FSTL DC      NIL       FREE STORAGE LIST
@FXSL DC      NIL       FIXED-POINT FREE LIST
      BSS  E  8
@SYSO DC      @OSTD     STD OUTPUT(SET TO 1 IF TYP)
@SYSI DC      @ISTD     STD INPUT (SET TO 6 IF TYP)
@SYSP DC      @OSTD     STANDARD PRINTER
@SYSR DC      @ISTD     STANDARD READER
      LIST
      HDNG    02 PUSHDOWN LIST SUBROUTINES
**************************************************
*     PUSH ACC ONTO REG PDL                      *
**************************************************
PSHA1 MDX   1 -1        MOVE POINTER
      STO   3 @TMPA-X
      LD   L  1         CHECK FOR OVERFLOW
      S     3 @SPDL-X
      BSC  L  PSHA2,+
      LD    3 @TMPA-X
      STO   1 0         PUT WORD ON STACK
      BSC  I  PUSHA
PSHA2 BSI   3 ERROR-X   HANDLE PDL OVERFLOW
PSHA3 DC      1+@MAJR   (ALTERED TO 11+@FATL FOR GC
**************************************************
*     POP ACC FROM REG PDL                       *
**************************************************
POPA1 LD   L  1         CHECK FOR UNDERFLOW
      S       POPA9
      BSC  L  POPA2,-
      LD    1 0         GET WORD
      MDX   1 1         MOVE POINTER
      BSC  I  POPA
POPA2 BSI   3 ERROR-X   HANDLE PDL UNDERFLOW
      DC      2+@DUMP
**************************************************
POPA9 DC      S@RPD
**************************************************
*     PUSH JUMP                                  *
**************************************************
PSHJ1 STO   3 @TMPB-X   SAVE ACC
      LD    3 PUSHJ-X
      A       PSHJ9
      BSI   3 PUSHA-X   PUSH RETURN ADR
      LD   I  PUSHJ     GET JUMP ADR
      STO     PSHJ2+1
      LD    3 @TMPB-X   RESTORE ACC
PSHJ2 BSC  L  *-*       JUMP
**************************************************
PSHJ9 DC      /8001     BIT 0 FLAG + 1
**************************************************
*     POP JUMP                                   *
**************************************************
POPJ1 STO   3 @TMPA-X   SAVE ACC
      BSI   3 POPA-X    GET RETURN ADR
      STO     POPJ2+1
      LD    3 @TMPA-X   RESTORE ACC
POPJ2 BSC  L  *-*       POP JUMP
**************************************************
*     PUSH ONTO SPEC PDL                         *
**************************************************
PSHS1 LD    3 @SPDL-X   CHECK FOR OVERFLOW
      A       PSHS9
      S    L  1
      BSC  L  PSHS3,-
      LD   I  PUSHS     GET ADR OF LOC TO SAVE
      OR      PSHS8     SET BIT 0
      MDX  L  @SPDL,1
      STO  I  @SPDL     PUT ON STACK
      STO     PSHS2+1
PSHS2 LD   L  *-*       GET CONTENTS OF LOC
      MDX  L  @SPDL,1
      STO  I  @SPDL     PUT ON STACK
      MDX  L  PUSHS,1
      BSC  I  PUSHS
PSHS3 BSI   3 ERROR-X   HANDLE PDL OVERFLOW
      DC      3+@MAJR
**************************************************
PSHS8 DC      /8000
PSHS9 DC      2
**************************************************
*     POP FROM SPEC PDL                          *
```

```
**************************************************
POPS1 LD    3 @SPDL-X   MOVE POINTER
      S       PSHS9
      STO   3 @SPDL-X
      STO     POPS9
      S       POPS8     CHECK FOR UNDERFLOW
      BSC   L POPS3,+Z
      MDX   L POPS9,1
      LD    I POPS9     GET ADR OF LOC TO RESTORE
      STO     POPS2+1
      MDX   L POPS9,1
      LD    I POPS9     GET CONTENTS
POPS2 STO   L *-*        AND PUT IN LOC
      BSC   I POPS
POPS3 BSI   3 ERROR-X   HANDLE PDL UNDERFLOW
      DC      4+@DUMP
**************************************************
POPS8 DC      S@SPD
POPS9 DC      *-*
**************************************************
*     POP N ENTRIES FROM SPEC PDL                *
**************************************************
POPN1 STO   3 @TMPA-X   SAVE ACC
      BSI   3 POPA-X    GET COUNT (NEGATIVE)
      BSC   L POPN4,+-  RETURN IF NONE TO POP
      STO     POPN9
POPN2 BSI   3 POPS-X    POP AN ENTRY
      MDX   L POPN9,1
      MDX     POPN2
POPN4 LD    3 @TMPA-X   RESTORE ACC
      BSC   I POPN
**************************************************
POPN9 DC      *-*
**************************************************
      LIST
      HDNG    03 SHORT XSUBRS AND BASIC FUNCTIONS
**************************************************
*     CAR XSUBR                                  *
**************************************************
XCAR1 BSC     +-        USE NIL IF ZERO
      LD      XCAR9
      A       XCAR8     ADD 1
      STO     XCAR2+1
XCAR2 LD   L  *-*       GET CAR
      AND     XCAR7     AND OUT BIT 0
      BSC  I  XCAR
**************************************************
XCAR7 DC      /7FFF
XCAR8 DC      1
XCAR9 DC      #NIL      ADR OF ATOM 'NIL'
**************************************************
*     CDR XSUBR                                  *
**************************************************
XCDR1 BSC     +-        USE NIL IF ZERO
      LD      XCAR9
      STO     XCDR2+1
XCDR2 LD   L  *-*       GET CDR
      BSC  I  XCDR
**************************************************
*     ATOM XSUBR                                 *
**************************************************
XATM1 STO     XATM2+1
      S       XATM9
      BSC  L  XATM3,+Z
      S       XATM8
      BSC  L  XATM3,-   ATOM IF OUTSIDE FREE STG
      MDX  L  XATM2+1,1
      NOP
XATM2 LD   L  *-*
      BSC  L  XATM3,+Z  ATOM IF BIT 0 OF CAR SET
      SRA     16
      BSC     +-Z
XATM3 LD    3 @TRUE-X
      BSC  I  XATOM
**************************************************
XATM8 DC      E@FST-S@FST
XATM9 DC      S@FST
**************************************************
*     NULL XSUBR                                 *
**************************************************
XNUL1 BSC  L  XNUL2,Z
      LD    3 @TRUE-X   RETURN 'T' IF NIL
      BSC     +-Z
XNUL2 SRA     16        RETURN 'NIL' IF TRUE
      BSC  I  XNULL
**************************************************
*     NUMBERP XSUBR                              *
**************************************************
XNMP1 BSC  L  XNMP3,+-  NOT A NUMBER IF NIL
      S       XNMP9
      BSC  L  XNMP5,+Z
      S       XNMP8
      BSC  L  XNMP5,-   NUMBER IF OUTSIDE FREE STG
XNMP3 SRA     16
      BSC     +-Z
XNMP5 LD    3 @TRUE-X
      BSC  I  XNMBP
**************************************************
XNMP8 DC      E@FST-S@FST
XNMP9 DC      S@FST
**************************************************
*     CHECK ARGUMENT FOR NUMBER - ERROR IF NOT   *
**************************************************
XNCH1 STO     XNCH6
      BSC  L  XNCH4,+-  NOT A NUMBER IF NIL
      S       XNCH9
      BSC  L  XNCH2,+Z  BRANCH IF NUMBER
      S       XNCH8
      BSC  L  XNCH4,+Z  BRANCH UNLESS NUMBER
XNCH2 LD   I  XNCH6     GET NUMBER IN ACC
      MDX  L  XNCHK,1
      BSC  I  XNCHK     RETURN
XNCH4 LD   I  XNCHK
      STO     XNCH5
      BSI   3 ERROR-X   ERROR - NOT A NUMBER
      DC      38+@MAJR
XNCH5 DC      *-*
XNCH6 DC      *-*
*************************************************
XNCH8 DC      E@FST-S@FST
XNCH9 DC      S@FST
*************************************************
*     STRINGP XSUBR                             *
*************************************************
XSTR1 STO     XSTR3+1
      S       XSTR9
      BSC  L  XSTR4,+Z  BRANCH IF NUMBER OR NIL
      S       XSTR8
      BSC  L  XSTR4,-   BRANCH IF NUMBER
      A       XSTR7
      STO     XSTR2+1
XSTR2 LD   L  *-*
      BSC  L  XSTR4,-   BRANCH UNLESS ATOM
XSTR3 LD   L  *-*
      EOR     XSTR6
      BSC  L  XSTR4,Z   BRANCH UNLESS STRING
      LD    3 @TRUE-X   RETURN T IF STRING
      BSC     +-Z
XSTR4 SRA     16        ELSE RETURN NIL
      BSC  I  XSTRP
*************************************************
XSTR6 DC      @STR
XSTR7 DC      1+E@FST
XSTR8 DC      E@FST-S@FST
XSTR9 DC      S@FST
*************************************************
*     CHECK ARGUMENT FOR STRING - ERROR IF NOT  *
*************************************************
XSCH1 STO     XSCH9     SAVE ARG
      BSI   3 XSTRP-X
      BSC  L  XSCH4,+-  BRANCH IF NOT STRING
      LD      XSCH9
      BSI   3 XCAR-X    ELSE RETURN CHAR LIST
      MDX  L  XSCHK,1
      BSC  I  XSCHK
XSCH4 LD   I  XSCHK     PRINT ERROR MESSAGE
      STO     XSCH6
      BSI   3 ERROR-X
      DC      47+@MAJR
XSCH6 DC      *-*
*************************************************
XSCH9 DC      *-*
*************************************************
*     CAR FUNCTION                              *
*************************************************
      DC      @LAM+1    (LAMBDA (X) ...
CAR   LD    3 @ARG1-X
      BSI   3 XCAR-X
      BSI   3 POPJ-X
*************************************************
*     QUOTE FUNCTION                            *
*************************************************
      DC      @NLAM+1   (NLAMBDA (X) ...
QUOTE LD    3 @ARG1-X
      BSI   3 POPJ-X
*************************************************
*     CDR FUNCTION                              *
*************************************************
      DC      @LAM+1    (LAMBDA (X) ...
CDR   LD    3 @ARG1-X
      BSI   3 XCDR-X
      BSI   3 POPJ-X
*************************************************
*     NULL PREDICATE                            *
*************************************************
      DC      @LAM+1    (LAMBDA (X) ...
NULL  LD    3 @ARG1-X
      BSI   3 XNULL-X
      BSI   3 POPJ-X
*************************************************
*     NOT PREDICATE                             *
*************************************************
NOT   EQU     NULL
*************************************************
*     ATOM PREDICATE                            *
*************************************************
      DC      @LAM+1    (LAMBDA (X) ...
ATOM  LD    3 @ARG1-X
      BSI   3 XATOM-X
      BSI   3 POPJ-X
*************************************************
*     NUMBERP PREDICATE                         *
*************************************************
      DC      @LAM+1    (LAMBDA (X) ...
NMBRP LD    3 @ARG1-X
      BSI   3 XNMBP-X
      BSI   3 POPJ-X
*************************************************
*     STRINGP FUNCTION                          *
*************************************************
      DC      @LAM+1
STRP  LD    3 @ARG1-X
      BSI   3 XSTRP-X
      BSI   3 POPJ-X
      LIST
      HDNG    04 EQ, *EQUALP, EQUAL
*************************************************
*     EQ PREDICATE                              *
*************************************************
      DC      @LAM+2    (LAMBDA (X Y) ...
EQ    LD    3 @ARG1-X   COMPARE TWO ARG ADRS
      EOR   3 @ARG2-X
      BSC  L  EQ2,Z
      LD    3 @TRUE-X
      BSI   3 POPJ-X
EQ2   SRA     16
      BSI   3 POPJ-X
*************************************************
      LIST
      HDNG    05 EBCDIC CHARACTER CODE TABLE
***** WARNING -- PNAME SCREWS IF ARG IS NIL
****************************************************
*     SYMBOLS FOR CHARACTER TABLE                  *
****************************************************
@QT   EQU     /80       QUOTE ON PRIN1 OUTPUT
@HNM1 EQU     /20       MAY BE 1ST CHAR OF HEX NUM
@DNM1 EQU     /08       MAY BE 1ST CHAR OF DEC NUM
@HNUM EQU     /04       MAY BE HEX DIGIT
@DNUM EQU     /01       MAY BE DEC DIGIT
****************************************************
*     EBCDIC CHARACTER CODE AND FLAG TABLE         *
****************************************************
EBCTB EQU     *
@     DC      256*. +@QT     BLANK
@CR   DC      /1500     CARRIAGE RETURN
@CENT DC      256*.\    CENTS SIGN
@PER  DC      256*..+@QT
@LESS DC      256*.<    LESS THAN SIGN
@LPAR DC      256*.(+@QT
@PLUS DC      256*.++@DNM1
@OR   DC      256*.|    LOGICAL OR
@AMPR DC      256*.&+@QT     AMPERSAND
@EXCL DC      256*.!    EXCLAMATION POINT
@$    DC      256*.$
@STAR DC      256*.*
@RPAR DC      256*.)+@QT
@SCLN DC      256*.;    SEMICOLON
@NOT  DC      256*.^    LOGICAL NOT
@DASH DC      256*.-+@DNM1
@SLSH DC      256*./+@HNM1
@COMA DC      256*.,+@QT
@PCNT DC      256*.%    PERCENT SIGN
@UNDR DC      256*._    UNDERSCORE
@GRTR DC      256*.>    GREATER THAN SIGN
@QUES DC      256*.?    QUESTION MARK
***** WARNING -- PDL OVERFLOW HANDLING ROTS
@COLN DC      256*.:    COLON
@NMBR DC      256*.#    NUMBER SIGN
@AT   DC      256*.@    AT SIGN
@QUOT DC      256*.'+@QT     QUOTE MARK
@EQAL DC      256*.=    EQUAL SIGN
@DBQT DC      256*."    DOUBLE QUOTE
@A    DC      256*.A+@HNUM
@B    DC      256*.B+@HNUM
@C    DC      256*.C+@HNUM
@D    DC      256*.D+@HNUM
@E    DC      256*.E+@HNUM
@F    DC      256*.F+@HNUM
@G    DC      256*.G
@H    DC      256*.H
@I    DC      256*.I
@J    DC      256*.J
@K    DC      256*.K
@L    DC      256*.L
@M    DC      256*.M
@N    DC      256*.N
@O    DC      256*.O
@P    DC      256*.P
@Q    DC      256*.Q
@R    DC      256*.R
@S    DC      256*.S
@T    DC      256*.T
@U    DC      256*.U
@V    DC      256*.V
@W    DC      256*.W
@X    DC      256*.X
@Y    DC      256*.Y
@Z    DC      256*.Z
@0    DC      256*.0+@DNM1+@DNUM+@HNUM
@1    DC      256*.1+@DNM1+@DNUM+@HNUM
@2    DC      256*.2+@DNM1+@DNUM+@HNUM
```

```
@3    DC      256*.3+@DNM1+@DNUM+@HNUM
@4    DC      256*.4+@DNM1+@DNUM+@HNUM
@5    DC      256*.5+@DNM1+@DNUM+@HNUM
@6    DC      256*.6+@DNM1+@DNUM+@HNUM
@7    DC      256*.7+@DNM1+@DNUM+@HNUM
@8    DC      256*.8+@DNM1+@DNUM+@HNUM
@9    DC      256*.9+@DNM1+@DNUM+@HNUM
L@EBC EQU     *-EBCTB   LENGTH OF EBCDIC TABLE
      LIST
      HDNG    06 1403/CPR PRINTER CODES TABLE
**************************************************
*     1403 PRINTER/CONSOLE PRINTER CODE TABLE    *
**************************************************
PRTTB DC      /7F21     BLANK
      DC      /7F81     CARRIAGE RETURN
      DC      /7F02     CENTS SIGN
      DC      /6E00     .
      DC      /7FDE     LESS THAN SIGN
      DC      /57FE     (
      DC      /6DDA     +
      DC      /7FC6     LOGICAL OR
      DC      /1544     AMPERSAND
      DC      /7F42     EXCLAMATION POINT
      DC      /6240     $
      DC      /23D6     *
      DC      /2FF6     )
      DC      /7FD2     SEMICOLON
      DC      /7FF2     LOGICAL NOT
      DC      /6184     -
      DC      /4CBC     /
      DC      /1680     COMMA
      DC      /7F06     PERCENT SIGN
      DC      /7FBE     UNDERSCORE
      DC      /7F46     GREATER THAN SIGN
      DC      /7F86     QUESTION MARK
      DC      /7F82     COLON
      DC      /7FC0     NUMBER SIGN
      DC      /7F04     AT SIGN
      DC      /0BE6     '
      DC      /4AC2     =
      DC      /7FE2     DOUBLE QUOTE
      DC      /643C     A
      DC      /2518     B
      DC      /261C     C
      DC      /6730     D
      DC      /6834     E
      DC      /2910     F
      DC      /2A14     G
      DC      /6B24     H
      DC      /2C20     I
      DC      /587C     J
      DC      /1958     K
      DC      /1A5C     L
      DC      /5B70     M
      DC      /1C74     N
      DC      /5D50     O
      DC      /5E54     P
      DC      /1F64     Q
      DC      /2060     R
      DC      /0D98     S
      DC      /0E9C     T
      DC      /4FB0     U
      DC      /10B4     V
      DC      /5190     W
      DC      /5294     X
      DC      /13A4     Y
      DC      /54A0     Z
      DC      /49C4     0
      DC      /40FC     1
      DC      /01D8     2
      DC      /02DC     3
      DC      /43F0     4
      DC      /04F4     5
      DC      /45D0     6
      DC      /46D4     7
      DC      /07E4     8
      DC      /08E0     9
      AIF     (L@EBC EQ *-PRTTB),.OKAY
ERR** .....   LENGTHS OF EBCTB AND PRTTB DIFFERENT
.OKAY ANOP
      LIST
      HDNG    07 CARD CODE CHARACTER TABLE
*************************************************
*     CARD CODE TABLE                           *
*************************************************
CRDTB DC      /0000      BLANK
      DC      /0001      CR (CANNOT BE READ IN)
      DC      /8820      CENTS SIGN
      DC      /8420      .
      DC      /8220      LESS THAN SIGN
      DC      /8120      (
      DC      /80A0      +
      DC      /8060      LOGICAL OR
      DC      /8000      AMPERSAND
      DC      /4820      EXCLAMATION POINT
      DC      /4420      $
      DC      /4220      *
      DC      /4120      )
      DC      /40A0      SEMICOLON
      DC      /4060      LOGICAL NOT
      DC      /4000      -
      DC      /3000      /
      DC      /2420      ,
      DC      /2220      PERCENT SIGN
      DC      /2120      UNDERSCORE
      DC      /20A0      GREATER THAN SIGN
      DC      /2060      QUESTION MARK
      DC      /0820      COLON
      DC      /0420      NUMBER SIGN
      DC      /0220      AT SIGN
      DC      /0120      '
      DC      /00A0      =
      DC      /0060      DOUBLE QUOTE
      DC      /9000      A
      DC      /8800      B
      DC      /8400      C
      DC      /8200      D
      DC      /8100      E
      DC      /8080      F
      DC      /8040      G
      DC      /8020      H
      DC      /8010      I
      DC      /5000      J
      DC      /4800      K
      DC      /4400      L
      DC      /4200      M
      DC      /4100      N
      DC      /4080      O
      DC      /4040      P
      DC      /4020      Q
      DC      /4010      R
      DC      /2800      S
      DC      /2400      T
      DC      /2200      U
      DC      /2100      V
      DC      /2080      W
      DC      /2040      X
      DC      /2020      Y
      DC      /2010      Z
      DC      /2000      0
      DC      /1000      1
      DC      /0800      2
      DC      /0400      3
      DC      /0200      4
      DC      /0100      5
      DC      /0080      6
      DC      /0040      7
      DC      /0020      8
      DC      /0010      9
      AIF     (L@EBC EQ *-CRDTB),.OKAY
ERR** .....   LENGTHS OF EBCTB AND CRDTB DIFFERENT
.OKAY ANOP
      LIST
      HDNG    00 OUTPT - OUTPUT ONE CHARACTER
**************************************************
*     OUTPUT ONE CHARACTER                       *
**************************************************
OUTPT DC       *-*
      STX    1 OUT40+1   SAVE XR1
      STX    2 OUT45+1   SAVE XR2
      LDX   I3 $XR3X     SET XR3 FOR LIBFS
      LDX   L1 @OSTD     PUT DEVICE NUMBER IN XR1
OUTDV EQU      *-1       DEV NUMBER MAY BE CHANGED
      S        OUT99     ACC HAS AN ADR WITHIN EBCTB
      STO      OUT30+1
      S        OUT98
      BSC   L  OUT20,+-  BRANCH IF CR
      LD    L1 OUTCH
OUT10 S        OUT97
      STO   L1 OUTCH     DECR CHAR COUNT
      BSC   L  OUT30,-   BRANCH UNLESS END OF LINE
      LDX    2 @CR-EBCTB
      BSI   I1 OUTSB     OUTPUT CARRIAGE RETURN
      LD    L1 OUTLN     RESET CHAR COUNT
      MDX      OUT10
OUT20 LD    L1 OUTLN     RESET CHAR COUNT
      STO   L1 OUTCH
OUT30 LDX   L2 *-*       PUT OFFSET BACK IN XR2
      BSI   I1 OUTSB     OUTPUT CHAR
OUT40 LDX   L1 *-*       RESTORE XR1
OUT45 LDX   L2 *-*       RESTORE XR2
      LDX   L3 X         RESTORE XR3
      BSC   I  OUTPT
**************************************************
OUT97 DC       1
OUT98 DC       @CR-EBCTB
OUT99 DC       EBCTB
**************************************************
*     SET UP OUTPT FOR PROPER OUTPUT DEVICE      *
**************************************************
STOUT DC       *-*
      LD    I  STOUT     GET NAME OF CALLING FN
      MDX   L  STOUT,1
      STO      STOU2
      LD     3 @ARG1-X   GET ARG 1
      BSI    3 XNCHK-X   CHECK IT
STOU2 DC       *-*
      BSI    3 PUSHJ-X   CHECK IF DEFINED
      DC       ODEVP
      BSC   L  STOU5,+-  ERROR IF NOT
      LD    I  @ARG1
      STO   L  OUTDV     SET DEVICE NUMBER
      MDX      STOU7
STOU5 LD     3 @ARG1-X   PRINT ERROR MESSAGE
      STO      STOU6
      BSI    3 ERROR-X   NOTE - ERROR WILL SET DEV
      DC       10+@MINR   NUMBER TO SYS PRINTER
STOU6 DC       *-*       FAULTY DEVICE NUMBER
STOU7 BSC   I  STOUT
**************************************************
*     SET UP OUTPT FOR SYSTEM OUTPUT DEVICE      *
**************************************************
SYSOU DC       *-*
SYSO2 LD    L  #SYSO     CHECK SYSOUT
      STO    3 @ARG1-X
      BSI    3 XNMBP-X
      BSC   L  SYS05,+-  BAD IF NOT NUMBER
      BSI    3 PUSHJ-X
      DC       ODEVP
      BSC   L  SYS05,+-  BRANCH IF BAD
      LD    I  #SYSO     USE SYSOUT
      BSC      +-Z
SYS05 LD     3 @SYSO-X   USE STANDARD OUTPUT
      STO   L  OUTDV
      BSC   I  SYSOU
```

```
***************************************************
*     CHRCT FUNCTION                              *
***************************************************
      DC        @LAM+1     (LAMBDA (X) ...
CHRCT BSI   L   STOUT      CHECK DEVICE NUMBER
      DC        #CHRC
      LD    I   @ARG1
      A         CHRC9
      STO       CHRC1+1
CHRC1 LD    L   *-*        GET CHRCT FOR GIVEN DEVICE
      BSI    3  MKFXN-X
      BSI    3  POPJ-X
***************************************************
CHRC9 DC        OUTCH
***************************************************
*     LINEL FUNCTION                              *
***************************************************
      DC        @LAM+1+@LIST  (LAMBDA (X . Y) ...
LINEL BSI   L   STOUT      CHECK DEVICE NUMBER
      DC        #LINE
      STX    2  LINE7+1    SAVE XR2
      LD    I   @ARG1      PUT DEVICE
      STO   L   2           NUMBER IN XR2
      LD     3  @ARG2-X
      BSC   L   LINE5,+-   BRANCH IF NO SECOND ARG
      BSI    3  XCAR-X
      BSI    3  XCDR-X     GET NUMBER
      BSC       +
      LD        LINE9      USE 72 IF NON-POSITIVE
      S     L2  OUTMX
      BSC       -Z
      SRA       16
      A     L2  OUTMX      USE MAXIMUM IF TOO LARGE
      RTE       16
      LD    L2  OUTLN      GET OLD LINEL
      RTE       16
      STO   L2  OUTLN      PUT NEW LINEL
      RTE       16
      MDX       LINE6
LINE5 LD    L2  OUTLN      RETURN OLD LINEL
LINE6 BSI    3  MKFXN-X
LINE7 LDX   L2  *-*
      BSI    3  POPJ-X
***************************************************
LINE9 DC        72
***************************************************
      LIST
      HDNG    09 TABLES AND SYMBOLS FOR I/O DEVICES
************************************************
*     TABLES FOR I/O DEVICES AND SUBRS         *
************************************************
*     ADRS OF OUTPUT DEVICE HANDLERS           *
************************************************
OUTSB DC      *-*       0 - USED BY PRIN1STR, ETC.
      DC      OCPRT     1 - TYPEWRITER
      DC      O1442     2 - 1442 CARD READ/PUNCH
      DC      O1132     3 - 1132 PRINTER
      DC      O1055     4 - PAPER TAPE PUNCH
      DC      O1403     5 - 1403 PRINTER
      DC      0         - - - - - - -
      DC      O1627     7 - 1627 PLOTTER
      DC      0         - - - - - - -
      DC      ODISK     9 - DISK FILE OUTPUT
L@OSB EQU     *-OUTSB-1 NUMBER OF OUTPUT DEVICES
************************************************
*     LINE LENGTHS FOR OUTPUT DEVICES          *
************************************************
OUTLN DC      32767     PRIN1STR, ETC.
      DC      100       TYPEWRITER
      DC      72        1442 CARD READ/PUNCH
      DC      120       1132 PRINTER
      DC      32767     PAPER TAPE PUNCH
      DC      120       1403 PRINTER
      DC      0         - - - - - - -
      DC      32767     1627 PLOTTER
      DC      0         - - - - - - -
      DC      72        DISK FILE OUTPUT
************************************************
*     CURRENT CHRCT VALUES FOR OUTPUT DEVICES  *
************************************************
OUTCH DC      *-*       PRIN1STR, ETC.
      DC      0         TYPEWRITER
      DC      0         1442 CARD READ/PUNCH
      DC      0         1132 PRINTER
      DC      0         PAPER TAPE PUNCH
      DC      0         1403 PRINTER
      DC      0         - - - - - - -
      DC      0         1627 PLOTTER
      DC      0         - - - - - - -
      DC      0         DISK FILE OUTPUT
************************************************
*     MAXIMUM LINE LENGTHS FOR OUTPUT DEVICES  *
************************************************
OUTMX DC      *-*       PRIN1STR, ETC.
      DC      100       TYPEWRITER
      DC      80        1442 CARD READ/PUNCH
      DC      120       1132 PRINTER
      DC      32767     PAPER TAPE PUNCH
      DC      120       1403 PRINTER
      DC      0         - - - - - - -
      DC      32767     1627 PLOTTER
      DC      0         - - - - - - -
      DC      72        DISK FILE OUTPUT
************************************************
*     PAGESKIP SUBROUTINES FOR OUTPUT DEVICES  *
************************************************
OPSKP DC      0         PRIN1STR, ETC.
      DC      PCPRT     TYPEWRITER
      DC      P1442     1442 CARD READ/PUNCH
      DC      P1132     1132 PRINTER
      DC      P1055     PAPER TAPE PUNCH
      DC      P1403     1403 PRINTER
      DC      0         - - - - - - -
      DC      P1627     1627 PLOTTER
      DC      0         - - - - - - -
      DC      PDISK     DISK FILE OUTPUT
************************************************
*     ADDRESSES OF INPUT DEVICE HANDLERS       *
************************************************
```

```
INSUB DC      RDS50    0 - READSTR FUNCTION
      DC      0        - - - - - - -
      DC      I1442    2 - 1442 CARD READ/PUNCH
      DC      0        - - - - - - -
      DC      I1134    4 - PAPER TAPE READER
      DC      0        - - - - - - -
      DC      IKBRD    6 - KEYBOARD
      DC      0        - - - - - - -
      DC      I2501    8 - 2501 CARD READER
      DC      IDISK    9 - DISK FILE INPUT
L@ISB EQU     *-INSUB-1 NUMBER OF INPUT DEVICES
*************************************************
*     INPUT PEEK CHARACTERS                     *
*************************************************
INPKC DC      0        READSTR FUNCTION
      DC      0        - - - - - - -
      DC      0        1442 CARD READ/PUNCH
      DC      0        - - - - - - -
      DC      0        PAPER TAPE READER
      DC      0        - - - - - - -
      DC      0        KEYBOARD
      DC      0        - - - - - - -
      DC      0        2501 CARD READER
      DC      0        DISK FILE INPUT
*************************************************
*     FLUSH SUBROUTINES FOR READ ERRORS         *
*************************************************
IFLSH DC      RDS70    READSTR FUNCTION
      DC      0        - - - - - - -
      DC      F1442    1442 CARD READ/PUNCH
      DC      0        - - - - - - -
      DC      F1134    PAPER TAPE READER
      DC      0        - - - - - - -
      DC      FKBRD    KEYBOARD
      DC      0        - - - - - - -
      DC      F2501    2501 CARD READER
      DC      FDISK    DISK FILE INPUT
*************************************************
*     EQUIVALENCES FOR DEVICE DESIGNATION       *
*************************************************
@CPRT EQU     YES      TYPEWRITER
@READ EQU     YES      1442 CARD READER
@PNCH EQU     YES      1442 CARD PUNCH
@1132 EQU     YES      1132 PRINTER
@1134 EQU     NO       1134 PAPER TAPE READER
@1055 EQU     NO       1055 PAPER TAPE PUNCH
@1403 EQU     NO       1403 PRINTER
@KBRD EQU     YES      KEYBOARD
@1627 EQU     NO       1627 PLOTTER
@2501 EQU     NO       2501 CARD READER
@IDSK EQU     YES      DISK FILE INPUT
@ODSK EQU     NO       DISK FILE OUTPUT
*************************************************
*     STANDARD DEVICES FOR INPUT/OUTPUT         *
*************************************************
      AIF     (@CPRT EQ YES),.DONE
      AIF     (@KBRD EQ NO),.DONE
@CPRT SET     YES
NOTE* AGO         KEYBOARD SPECIFIED WITHOUT TYPE-
      AGO         WRITER - TYPEWRITER IS ASSUMED
      AGO         (NECESSARY FOR KEYBOARD ECHO)
      ANOP
.DONE ANOP
      AIF     (@1403 EQ NO),.1132
@OSTD EQU     5
      AGO     .DONE
.1132 AIF     (@1132 EQ NO),.CPRT
@OSTD EQU     3
      AGO     .DONE
.CPRT AIF     (@CPRT EQ YES),.OKAY
NOTE* AGO         NO PRINTER SPECIFIED -
      AGO         TYPEWRITER ASSUMED FOR OUTPUT
      ANOP
@CPRT SET     YES
.OKAY ANOP
@OSTD EQU     1
.DONE ANOP
      AIF     (@2501 EQ NO),.1442
@ISTD EQU     8
      AGO     .DONE
.1442 AIF     (@READ EQ NO),.1134
@ISTD EQU     2
      AGO     .DONE
.1134 AIF     (@1134 EQ NO),.KBRD
@ISTD EQU     4
      AGO     .DONE
.KBRD AIF     (@KBRD EQ YES),.OKAY
NOTE* AGO         NO INPUT DEVICE SPECIFIED -
      AGO         KEYBOARD ASSUMED FOR INPUT
      ANOP
@KBRD SET     YES
.OKAY ANOP
@ISTD EQU     6
.DONE ANOP
@PSTD EQU     0
      AIF     (@PNCH EQ NO),.1055
@PSTD SET     2
      AGO     .DONE
.1055 AIF     (@1055 EQ NO),.DONE
@PSTD SET     4
.DONE ANOP
      LIST
      HDNG    10 XTYO, TYO, TERPRI
*************************************************
*     OUTDEVP/INDEVP FUNCTIONS                  *
*************************************************
      DC       @LAM+1    (LAMBDA (X) ...
ODEVP LD       XDVP9     SET UP FOR OUTDEVP
      STO      XDVP7
      LDD      XDVP6
      MDX      XDVP1
*************************************************
      DC       @LAM+1    (LAMBDA (X) ...
IDEVP LD       XDVP8     SET UP FOR INDEVP
      STO      XDVP7
      LDD      XDVP5
XDVP1 A     I  @ARG1
      STO      XDVP3+1
      RTE      16
      STO      XDVP2
      LD     3 @ARG1-X   GET ARG 1
      BSI    3 XNCHK-X   CHECK IT
XDVP2 DC       *-*
      BSC   L  XDVP4,+   NOT DEFINED IF NON-POSITIVE
      S        XDVP7
      BSC   L  XDVP4,-Z  NOT DEFINED IF TOO LARGE
XDVP3 LD    L  *-*
      BSC   L  XDVP4,+-  NOT DEFINED IF NO HANDLER
      LD     3 @TRUE-X
      BSI    3 POPJ-X    RETURN T
XDVP4 SRA      16
      BSI    3 POPJ-X    RETURN NIL
*************************************************
      BSS   E  0
XDVP5 DC       INSUB
      DC       #IDVP
XDVP6 DC       OUTSB
      DC       #ODVP
XDVP7 DC       *-*
XDVP8 DC       L@ISB
XDVP9 DC       L@OSB
*************************************************
*     TYO SUBROUTINE                            *
*************************************************
XTYO  DC       *-*
      STO      XTYO8     SAVE CHAR
      STX    2 XTYO4+1   SAVE XR2
      LDX    2 -L@EBC
XTYO1 LD    L2 EBCTB+L@EBC    SEARCH EBCDIC TABLE
      SRA      8
      EOR      XTYO8     COMPARE TO NUMERIC ARG
      BSC   L  XTYO2,+-
      MDX    2 1
      MDX      XTYO1
      LD       XTYO9     USE BLANK IF NOT FOUND
      MDX      XTYO3
XTYO2 LD    L  2         FORM ADR OF CHAR
      A        XTYO1+1
XTYO3 BSI   L  OUTPT     PRINT CHAR
XTYO4 LDX   L2 *-*       RESTORE XR2
      BSC   I  XTYO
*************************************************
XTYO8 DC       *-*
XTYO9 DC       EBCTB
*************************************************
*     TYO FUNCTION                              *
*************************************************
      DC       @LAM+2    (LAMBDA (X Y) ...
TYO   BSI   L  STOUT     SET OUTPUT DEVICE
      DC       #TYO
      LD    I  @ARG2     GET EBCDIC TO OUTPUT
      BSI      XTYO
      LD     3 @ARG2-X   RETURN SECOND ARG
      BSI    3 POPJ-X
```

```
***************************************************
      LIST
      HDNG    PRINT/PRIN1/PRINC FUNCTIONS
*************************************************
*     PRINT S-EXPRESSION                        *
*************************************************
*     ARG IS IN ACC - OUTDV AND AMPSW MUST      *
*     ALREADY BE APPROPRIATELY SET.             *
*************************************************
PREXP BSI   3 PUSHA-X   SAVE ARG ON STACK
      STX   2 PR040+1   SAVE XR2
      BSC  L  PR040,+-  BRANCH IF NIL
      BSI   3 XATOM-X
      BSC  L  PR600,+-  BRANCH UNLESS ATOM
      LD    1 0         GET ARG
      S       PR901
      BSC  L  PR200,+Z  BRANCH IF NUMBER
      S       PR902
      BSC  L  PR200,-   BRANCH IF NUMBER
      LD   I1 0
      EOR     PR903
      BSC  L  PR100,+-  BRANCH IF STRING
PR010 MDX  L  AMPSW,0   SKIP IF AMPSW ZERO (PRIN1)
      MDX     PR040     BRANCH IF NOT (PRINC)
      LDX   2 7         SET BITS IN CHAR TYPE SW
      STX   2 PR904      (ONLY TWO BITS NOW US&D)
      LDX   2 3         SET XR2 TO SHIFT FIRST CHAR
      LD    1 0         GET PNAME OF ATOM
      BSI   3 XCAR-X
PR020 BSC  L  PR030,+-  BRANCH IF DONE
      STO     PR905
      BSI   3 XCAR-X
      BSI   3 XCDR-X    GET CHAR BIT INDICATORS
      SRA   2           SHIFT IF FIRST CHAR
      AND     PR904     AND INDICATORS OVER TYPE SW
      BSC  L  PR040,+-  BRANCH IF NONE LEFT
      STO     PR904     ELSE SAVE
      LDX   2 0         SET XR2 FOR NEXT CHARS
      LD   I  PR905     GET NEXT CHAR
      MDX     PR020
PR030 LD      PR905     CHECK LAST CHAR FOR
      BSI   3 XCAR-X     -, +, OR / (IF SO,
      S       PR906      IT IS ONLY CHAR)
      BSC     Z
      S       PR907
      BSC     Z
      S       PR908
      BSC  L  PR040,+-  IF SO, BRANCH
      LD      PR909     ATOM LOOKS LIKE A NUMBER,
      BSI  L  OUTPT      PRECEDE WITH AMPERSAND
PR040 LDX  L2 *-*       RESTORE XR2
      LD    1 0         GET ARG
      BSI   3 XCAR-X    GET PNAME OF ATOM
PR050 BSC  L  PR230,+-  BRANCH IF DONE
      STO     PR904     SAVE PNAME LIST
      BSI   3 XCAR-X    GET CHAR
      MDX  L  AMPSW,0   SKIP IF AMPSW ZERO (PRIN1)
      MDX     PR070
      STO     PR905     SAVE CHAR
      LD   I  PR905
      SLA     8         BIT 8 SET MEANS QUOTE IT
      BSC  L  PR060,-   BRANCH UNLESS SET
      LD      PR909
      BSI  L  OUTPT     PRINT AMPERSAND
PR060 LD      PR905     GET CHAR
PR070 BSI  L  OUTPT     OUTPUT CHAR
      LD   I  PR904     CHAIN DOWN PNAME
      MDX     PR050
*************************************************
PR901 DC      S@FST
PR902 DC      E@FST-S@FST
PR903 DC      @STR
PR904 DC      *-*
PR905 DC      *-*
```

```
PR906 DC       @DASH
PR907 DC       @PLUS-@DASH
PR908 DC       @SLSH-@PLUS
PR909 DC .     @AMPR
***************************************************
PR100 MDX  L  AMPSW,0   SKIP IF AMPSW ZERO (PRIN1)
      MDX     PR110
      LD      PR920     PRINT LEADING COMMA
      BSI  L  OUTPT
PR110 LD    1 0         GET ARG
      BSI   3 XCAR-X    GET PNAME OF STRING
PR120 BSC  L  PR140,+-  BRANCH IF DONE
      STO     PR904
      BSI   3 XCAR-X    GET CHAR
      S       PR920
      BSC  L  PR130,Z   BRANCH UNLESS COMMA
      MDX  L  AMPSW,0
      MDX     PR130
      LD      PR920
      BSI  L  OUTPT     PRIN1 DOUBLES THE COMMA
      SRA     16        THIS GETS THE SECOND COMMA
PR130 A       PR920
      BSI  L  OUTPT     OUTPUT CHAR
      LD   I  PR904     CHAIN DOWN PNAME LIST
      MDX     PR120
PR140 MDX  L  AMPSW,0   SKIP IF PRIN1
      MDX     PR230
      LD      PR920
      BSI  L  OUTPT     PRINT TRAILING COMMA
      MDX     PR230
***************************************************
PR920 DC      @COMA
***************************************************
PR200 MDX  L  #HEX,0    OUTPUT NUMBER - TEST HEX
      MDX     PR300     BRANCH FOR HEX NUMBER
      SRA     16        OUTPUT DECIMAL NUMBER
      STO     PR904     CLEAR DIGIT COUNTER
      LD   I1 0         GET NUMBER
      BSC  L  PR210,-   BRANCH UNLESS NEGATIVE
      LD      PR906
      BSI  L  OUTPT     PRINT MINUS SIGN
      SRA     16
      S    I1 0         NOW USE ABSOLUTE VALUE
PR210 RTE     16        PUT NUMBER IN EXT
      SRA     16        CLEAR ACC
      D       PR930     GET REMAINDER MOD 10
      RTE     16
      BSI   3 PUSHA-X   PUSH REMAINDER
      MDX  L  PR904,1   INCR DIGIT COUNT
      RTE     16
      BSC  L  PR210,Z   MORE IF QUOTIENT NOT ZERO
PR220 BSI   3 POPA-X    POP DIGIT OFF STACK
      A       PR931     CONVERT TO CHAR TABLE ADR
      BSI  L  OUTPT     PRINT CHAR
      MDX  L  PR904,-1  COUNT DIGITS
      MDX     PR220
PR230 BSI   3 POPA-X    POP ARG OFF STACK
      BSI   3 POPJ-X
***************************************************
PR930 DC      10
PR931 DC      @0
***************************************************
PR300 LD      PR940     HEX NUMBER - PRINT SLASH
      BSI  L  OUTPT
      LD   I1 0         GET NUMBER
      STO     PR905     SAVE IT
      LDX   2 4         COUNT HEX DIGITS
PR310 SLT     16        CLEAR EXT
      LD      PR905
      RTE     28        GET ONE HEX DIGIT IN EXT
      STO     PR905     SAVE REST
      RTE     16
      S       PR930     CONVERT TO CHAR TABLE ADR
      BSC     -         SKIP IF 0-9
      A       PR941     ELSE A-F
      A       PR942
      BSI  L  OUTPT     PRINT HEX DIGIT
      MDX   2 -1
      MDX     PR310
      LDX  I2 PR040+1   RESTORE XR2
      MDX     PR230
**************************************************
PR940 DC      @SLSH
PR941 DC      @A-@0-10
PR942 DC      @0+10
**************************************************
PR600 LD      PR980
      BSI  L  OUTPT     PRINT '('
      LD    1 0
PR610 BSI   3 XCAR-X    GET CAR OF ARG
      BSI   3 PUSHJ-X   PRINT IT
      DC      PREXP
      LD   I1 0         GET CDR OF ARG
      BSC  L  PR650,+-  BRANCH IF NIL
      BSI   3 XATOM-X
      BSC  L  PR640,Z   BRANCH IF ATOM
      LD      PR981
      BSI  L  OUTPT     PRINT BLANK
      LD   I1 0         GET CDR OF ARG
      STO   1 0         SAVE IT
      MDX     PR610
PR640 LD      PR981     PRINT ' . '
      BSI  L  OUTPT
      LD      PR982
      BSI  L  OUTPT
      LD      PR981
      BSI  L  OUTPT
      LD   I1 0         PRINT ATOM
      BSI   3 PUSHJ-X
      DC      PREXP
PR650 LD      PR983     PRINT ')'
      BSI  L  OUTPT
      MDX     PR230
**************************************************
PR980 DC      @LPAR
PR981 DC      @
PR982 DC      @PER
PR983 DC      @RPAR
**************************************************
*     PRIN1/PRINC/PRINT FUNCTIONS                *
**************************************************
      DC      @LAM+1+@LIST
PRINT BSI  L  STOUT     SET OUTPUT DEVICE
      DC      #PRNT
      LD      PRI90
      BSI  L  OUTPT     PRINT CR
      LDS     1         SET FOR CR'S AFTER ITEMS
      MDX     PRI10
**************************************************
      DC      @LAM+1+@LIST
PRINC BSI  L  STOUT     SET OUTPUT DEVICE
      DC      #PRNC
      STX     AMPSW     SET FOR NO AMPERSANDS
      LDS     0
      MDX     PRI20
**************************************************
      DC      @LAM+1+@LIST
PRIN1 BSI  L  STOUT     SET OUTPUT DEVICE
      DC      #PRN1
      LDS     0
PRI10 SRA     16
      STO     AMPSW     SET FOR AMPERSANDS
PRI20 STS     PRI40     SAVE CR INDICATOR
      LD    3 @ARG2-X   GET LIST OF ARGS
```

```
PRI30 BSC  L  PRI60,+-  BRANCH IF DONE
      STO   3 @ARG2-X   SAVE LIST
      BSI   3 XCAR-X    GET FIRST ITEM ON LIST
      BSI   3 PUSHJ-X   PRINT IT
      DC       PREXP
PRI40 LDS      *-*
      LD       PRI90    IF PRINT, OUTPUT A CR
      BSI  L  OUTPT,0    AFTER EACH ITEM
      LD   I  @ARG2     CHAIN DOWN LIST OF ARGS
      MDX      PRI30
PRI60 LD    3 @ARG2-X
      BSC      Z        IF NO ARGS, RETURN NIL
      BSI   3 XCAR-X    ELSE RETURN LAST ARG
      BSI   3 POPJ-X
**************************************************
PRI90 DC       @CR
**************************************************
AMPSW DC       *-*      0 = PREXP USES AMPERSANDS
**************************************************
*     TYPEWRITER HANDLER                         *
**************************************************
      AIF      (@CPRT EQ YES),.YES
      AIF      (@KBRD EQ YES),.YES
OCPRT EQU      0
PCPRT EQU      0
      AGO      .NO
.YES  ANOP
OCPRT DC       *-*
      LD   L2 PRTTB     GET TYPEWRITER CODE CHAR
      SLA      8
      LIBF     KBCP0    PRINT IT
      BSC  I  OCPRT
**************************************************
PCPRT DC       *-*      TYPEWRITER PAGESKIP
      LDX   2 5
PCPR4 LD       PCPR9    OUTPUT 5 CARRIAGE RETURNS
      BSI  L  OUTPT
      MDX   2 -1
      MDX      PCPR4
      BSC  I  PCPRT
**************************************************
PCPR9 DC       @CR
**************************************************
.NO   ANOP
**************************************************
I1134 EQU      0
F1134 EQU      0
O1055 EQU      0
P1055 EQU      0
O1403 EQU      0
P1403 EQU      0
O1627 EQU      0
P1627 EQU      0
I2501 EQU      0
F2501 EQU      0
ODISK EQU      0
PDISK EQU      0
      LIST
      HDNG    13 ERROR HANDLING SUBROUTINE
***************************************************
*     SYMBOLS FOR ERROR TYPES                     *
***************************************************
@INFO EQU     /0000     INFORMATIONAL MESSAGE ONLY
@MINR EQU     /1000     MINOR ERROR - DEFAULT USED
@MAJR EQU     /2000     MAJOR ERROR - RECOVERABLE
@FATL EQU     /3000     FATAL ERROR - CALL EXIT
@DUMP EQU     /4000     SYSTEM ERROR - CORE DUMP
***************************************************
*     ERROR SUBROUTINE                            *
***************************************************
ERR01 STX  L2 ERR50+1   SAVE XR2
      BSI   3 PUSHS-X   SAVE OLD VALUE OF HEX
      DC       #HEX
      LD    3 @TRUE-X
      STO  L  #HEX      SET FOR HEX OUTPUT
      BSI  L  SYSOU     SET SYSTEM OUTPUT DEVICE
      LD   I  ERROR     GET ERROR NUMBER AND TYPE
      STO     ERR83     SAVE
      MDX  L  ERROR,1
      SRT     12
      STO  L  ERR45+1   SAVE TYPE
      S       ERR82
      BSC  L  ERR07,+Z  BRANCH IF NOT @FATL/@DUMP
      SRA     16        MAKE SURE THAT THIS MESSAGE
      STO     ERR99      GETS PRINTED
ERR07 SRA     16
      SLT     9         COMPUTE SECTOR ADR
      A       ERR98+1
      STO  L  DSKBF+1
      SRA     16
      STO     ERR94     SET SW FOR GETTING CHARS
      STO     ERR99     CLEAR ERRGAG SWITCH
      STO  L  AMPSW
      SLT     3         COMPUTE LOCATION OF MESSAGE
      M       ERR97      WITHIN BUFFER
      SLT     16
      A       ERR96
      STO     ERR20+1
      LD      ERR93
      SRT     16
      BSI  L  DISKZ     READ SECTOR OF ERROR FILE
      LD      ERR92     PRINT CR,'***** '
      BSI  L  OUTPT
      LDX   2 5
ERR10 LD      ERR91
      BSI  L  OUTPT
      MDX   2 -1
      MDX     ERR10
      LD      ERR90
      BSI  L  OUTPT
ERR15 MDX  L  $DBSY,0   WAIT FOR DISK READ
      MDX     ERR15
      LDX   2 -40
ERR20 LD   L2 *-*       GET TWO CHARS
      MDX  L  ERR94,0   SKIP FOR LEFT CHAR
      SLA     8
      SRA     8
      STO     ERR89     SAVE CHAR
      S       ERR88
      BSC  L  ERR25,+-  BRANCH IF CENTS SIGN
      S       ERR87
      BSC  L  ERR30,+-  BRANCH IF NUMBER SIGN
      S       ERR86
      BSC  L  ERR35,+-  BRANCH IF AT SIGN
      S       ERR85
      BSC  L  ERR43,+-  BRANCH IF SEMICOLON
      S       ERR84
      BSC  L  ERR31,+-  BRANCH IF PERCENT
      MDX  L  ERR99,0
      MDX     ERR40
      LD        ERR89
      BSI  L  XTYO      OUTPUT CHAR
      MDX       ERR40
ERR25 MDX  L  ERR99,0
      MDX       ERR40
      LD        ERR92     PRINT CARRIAGE RETURN
      BSI  L  OUTPT
      MDX       ERR40
ERR30 LD    3 @TRUE-X   SET FOR HEX OUTPUT
      BSC       +-Z
ERR31 SRA       16        SET FOR DEC
      STO  L  #HEX
      LD   L  ERROR
      MDX  L  ERROR,1
      BSI   3 PUSHJ-X   PRINT NUMBER
      DC        PREXP
      LD    3 @TRUE-X   SET HEX BACK TO T
      STO  L  #HEX
      MDX,      ERR40
***************************************************
ERR82 DC        @FATL
ERR83 DC        *-*
ERR84 DC        .%-.;     (PERCENT)-(SEMICOLON)
ERR85 DC        .;-.@     (SEMICOLON)-(AT SIGN)
ERR86 DC        .@-.#     (AT SIGN)-(NUMBER SIGN)
ERR87 DC        .#-.\     (NUMBER SIGN)-(CENTS SIGN)
ERR88 DC        .\        (CENTS SIGN)
ERR89 DC        *-*
ERR90 DC        @
ERR91 DC        @STAR
ERR92 DC        @CR
ERR93 DC        DSKBF
ERR94 DC        *-*
ERR95 DC        1
ERR96 DC        DSKBF+42
ERR97 DC        40
ERR98 DSA       LERRS     FILE OF ERROR MESSAGES
ERR99 DC        *-*
***************************************************
ERR35 LD   I  ERROR     PRINT AN ARBITRARY LIST
      MDX  L  ERROR,1
      MDX  L  ERR99,0
      MDX       ERR40
      BSI   3 PUSHJ-X
      DC        PREXP
ERR40 LD        ERR94     FLIP CHAR SWITCH
      EOR       ERR95
      STO       ERR94
      BSC  L  ERR20,Z
      MDX   2 1
      MDX       ERR20
ERR43 MDX  L  ERR99,0
      MDX       ERR45
      LD        ERR92     PRINT CR
      BSI  L  OUTPT
      BSI   3 POPS-X    POP OLD VALUE FOR HEX
ERR45 LDX  L2 *-*       PUT ERROR TYPE IN XR2
      BSC  I2 ERR47     BRANCH ON ERROR TYPE
***************************************************
ERR47 DC        ERR50     NORMAL RETURN
      DC        ERR50     NORMAL RETURN
      DC        ERR55     RECOVER ROUTINE
      DC        $EXIT     CALL EXIT
      DC        $DUMP+1   CORE DUMP
***************************************************
ERR50 LDX  L2 *-*       RESTORE XR2
      BSC  I  ERROR
***************************************************
ERR55 MDX  L  REDSW,0   CHECK IF THIS IS READ ERROR
      MDX       ERR58     BRANCH IF NOT
      STX  L  REDSW     RESET READ SWITCH
ERR56 LDS       0         WAS IT DUE TO /*/*/
```

```
      BSC  L  ERR58,0   BRANCH IF SO
      LDX  I1 INDEV
      BSI  I1 IFLSH     FLUSH INPUT DEVICE
ERR58 LDS     0         RESET SWITCH
      STS     ERR56
      MDX  L  ERP99,0   SKIP IF NO OUTSIDE ERRSET
      MDX     ERR60
      LD      XERR9     RESTORE SPEC PDL
      S     3 @SPDL-X
      SRT     1
      BSI   3 PUSHA-X
      BSI   3 POPN-X
      BSC  L  LSPER     BRANCH TO TOP LEVEL
ERR60 BSI     ERPOP     POP GARBAGE OFF PDL'S
      LD      ERR83     GET ERROR NUMBER
      SLA     8         LOSE TOP EIGHT BITS
      SRA     8
      BSI   3 MKFXN-X   RETURN ERROR NUMBER
      BSI   3 POPJ-X     FOR ERRSET
**************************************************
*     ERR FUNCTION                               *
**************************************************
      DC      @LAM+@LIST
XERR  LD    3 @ARG1-X   GET ARG LIST
      BSC     Z         SKIP IF NONE (NIL)
      BSI   3 XCAR-X    ELSE GET IT
      STO     XERR1     SAVE
      MDX  L  ERP99,0   SKIP IF NO OUTSIDE ERRSET
      MDX     XERR3
      LD      XERR9     RESTORE SPEC PDL
      S     3 @SPDL-X
      SRT     1
      BSI   3 PUSHA-X
      BSI   3 POPN-X
      BSI   3 ERROR-X   PRINT MESSAGE
      DC      41+@INFO
XERR1 DC      *-*
XERR3 BSI     ERPOP     POP GARBAGE OFF PDL'S
      BSC  L  LSPER     BRANCH TO TOP LEVEL
      LD      XERR1     RETURN GIVEN RESULT
      BSI   3 POPJ-X
**************************************************
XERR9 DC      S@SPD
**************************************************
*     POP UNWANTED JUNK OFF STACKS ON ERROR      *
**************************************************
ERPOP DC      *-*
      LD      ERP99     CHECK SPEC PDL
      S     3 @SPDL-X
      BSC  L  ERP10,+   BRANCH UNLESS OVERPOPPED
      BSI   3 ERROR-X
      DC      40+@DUMP
ERP10 STX   1 ERP98     SAVE REG PDL POINTER
      SRT     1         POP GARBAGE OFF SPEC PDL
      BSC  L  ERP30,-
      STO     ERP97
ERP20 BSI   3 POPS-X
      MDX  L  ERP97,1
      MDX     ERP20
ERP30 LD   L  1         CHECK REG PDL
      S       ERP98
      BSC  I  ERPOP,-   RETURN UNLESS OVERPOPPED
      BSI   3 ERROR-X
      DC      39+@DUMP
**************************************************
ERP97 DC      *-*
ERP98 DC      *-*
ERP99 DC      0
**************************************************
*     ERRSET FUNCTION                            *
**************************************************
      DC      @NLAM+1+@LIST
```

```
ERSET LD     3 @SPDL-X   SAVE CURRENT SPEC PDL LEVEL
      STO      ERP98
      BSI    3 PUSHS-X   PUSH LAST LEVEL
      DC       ERP99
      LD       ERP98     PUT THIS LEVEL IN SWITCH
      STO      ERP99
      BSI    3 PUSHS-X   PUSH ERR GAG SWITCH
      DC       ERR99
      BSI    3 PUSHS-X   PUSH REG PDL LEVEL
      DC       1          (XR1 IS REG PDL POINTER)
      LD     3 @ARG2-X   USE ARG 2, IF ANY, TO
      BSC      Z          SET ERR GAG SWITCH -
      BSI    3 XCAR-X     NIL = PRINT ERRORS
      STO   L  ERR99      NON-NIL = GAG
      BSI    3 PUSHJ-X   EVAL FIRST ARGUMENT
      DC       EVAL
      RTE      16        SAVE IN EXT
      BSI    3 POPS-X    RESTORE REG PDL
      BSI    3 POPS-X    RESTORE GAG SW
      BSI    3 POPS-X    RESTORE SPEC PDL LEVEL
      SRA      16
      BSI    3 XCONS-X   RETURN LIST OF EVALED ARG
      BSI    3 POPJ-X
**************************************************
      BSS   E  0
DSKBF DC       320       WORD COUNT
      DC       *-*       SECTOR ADDRESS
      BSS      320       ERROR MESSAGE DISK BUFFER
**************************************************
      LIST
      HDNG    14 GARBAGE COLLECTOR (RECLAIMER)
****************************************************
*     GARBAGE COLLECTOR                            *
****************************************************
*     COLLECTION PROCEEDS IN 7 STEPS -             *
*             1. CLEAR BIT TABLES                  *
*             2. MARK OBLIST STRUCTURE             *
*             3. MARK ATOMS ON OBLIST (NON-TWA)    *
*             4. MARK TEMLIST AND ITEMS THEREON    *
*             5. MARK FROM BOTH PDL'S              *
*             6. REMOVE TRULY WORTHLESS ATOMS      *
*             7. CONSTRUCT FREE LISTS              *
****************************************************
      DC      @NLAM
GC    STX   2 GC903     SAVE XR2
      LDX   2 0         INDICATE TYPE
      MDX     GC010
****************************************************
GCFST DC      *-*
      STX   2 GC903     SAVE XR2
      LDX   2 1         INDICATE TYPE
      MDX  L  #GCGA,0   IS GCGAG NIL
      MDX     GC010
      BSI   3 ERROR-X   IF SO, PRINT MESSAGE
      DC      8+@INFO
      MDX     GC010
****************************************************
GCFXS DC      *-*
      STX   2 GC903     SAVE XR2
      LDX   2 2         INDICATE TYPE
      MDX  L  #GCGA,0   IS GCGAG NIL
      MDX     GC010
      BSI   3 ERROR-X   IF SO, PRINT MESSAGE
      DC      9+@INFO
****************************************************
GC010 STX  L2 GC720+1   SAVE TYPE INDICATOR
      LD      GC909     ALTER ERROR NUMBER IN
      STO  L  PSHA3      CASE OF PDL OVERFLOW
************** 1. CLEAR BIT TABLES *****************
      SRA     16
      LDX  L2 L@FXB     CLEAR FXS BIT TABLE
GC020 STO  L2 S@FXB-1
      MDX   2 -1
      MDX     GC020
************** 2. MARK OBLIST STRUCTURE ************
      LD   L  #OBLS     GET ADR OF OBLIST
      STO     GC085     SAVE IN CASE OF ERROR
GC050 AND     GC902
      BSC  L  GC100,+-  BRANCH IF NIL
      STO     GC906
      LD   I  GC906     MARK NODE
      OR      GC907
      STO  I  GC906
      LD      GC906     MARK ATOM TEMPORARILY
      BSI   3 XCAR-X
      STO     GC065+1
GC065 LD   L  *-*
      S       GC901
      BSC     Z         DON'T MARK IF POTENTIAL TWA
      OR      GC907
      A       GC901
      STO  I  GC065+1
      MDX  L  GC065+1,1
      NOP
      LD   I  GC065+1
      BSC  L  GC080,-   ERROR IF NOT AN ATOM
      LD   I  GC906     (CDR GC906)
      MDX     GC050
GC080 LDX  L2 L@FST     CLEAR ALL MARK BITS
GC082 LD   L2 S@FST-2    IN FREE STORAGE
      AND     GC902
      STO  L2 S@FST-2
      MDX   2 -2
      MDX     GC082
      BSI   3 ERROR-X   PRINT ERROR MESSAGE
      DC      16+@FATL
GC085 DC      *-*
************** 3. MARK ATOMS ON OBLIST (NON-TWA) ***
GC100 LD   L  #OBLS     GET ADR OF OBLIST
GC110 AND     GC902
      BSC  L  GC200,+-  BRANCH IF NIL
      STO     GC906
      BSI   3 XCAR-X    GET ADR OF AN ATOM
      STO     GC905
      LD   I  GC905
      BSC  L  GC130,-   BRANCH IF POTENTIAL TWA
      AND     GC902
      STO  I  GC905     UNDO TEMP MARK
      LD      GC905
      BSI   3 PUSHJ-X   MARK THE ATOM
      DC      GCMRK
GC130 LD   I  GC906     (CDR GC906)
      MDX     GC110
************** 4. MARK TEMLIST AND ITEMS THEREON ***
GC200 LD    3 @TMLS-X   GET ADR OF TEMLIST
GC210 AND     GC902
      BSC  L  GC300,+-  BRANCH IF NIL
      STO     GC908
      LD   I  GC908     MARK NODE OF TEMLIST
      OR      GC907
      STO  I  GC908
      LD      GC908
      BSI   3 XCAR-X    TAKE CAR
      STO     GC220+1   GET CONTENTS OF LOC
GC220 LD   L  *-*        POINTED TO BY CAR
      BSI.  3 PUSHJ-X   MARK IT
      DC      GCMRK
      LD   I  GC908     (CDR GC908)
      MDX     GC210
***************************************************
GC901 DC      @UNDF
GC902 DC      /7FFF
GC903 DC      *-*       XR2 SAVED HERE
GC905 DC      *-*
GC906 DC      *-*
GC907 DC      /8000
GC908 DC      *-*
GC909 DC      11+@FATL  ERR NO. FOR GC PDL OVERFLOW
***************************************************
************** 5. MARK FROM BOTH PDL'S *************
GC300 LD   L  1         MARK FROM REG PDL...
GC310 STO     GC908
      S    L  GC919
      BSC  L  GC350,-   BRANCH IF DONE
      LD   I  GC908     GET WORD FROM PDL
      BSC  L  GC320,+   BRANCH IF BIT 0 SET
      BSI   3 PUSHJ-X   MARK FROM ADR FROM PDL
      DC      GCMRK
GC320 LD      GC908     MOVE DOWN PDL
      A       GC918
      MDX     GC310
GC350 LD    3 @SPDL-X   MARK SPEC PDL...
GC360 STO     GC908
      S       GC917
      BSC  L  GC400,+   BRANCH IF DONE
      LD   I  GC908
      BSC  L  GC370,+   BRANCH IF BIT 0 SET
      BSI   3 PUSHJ-X   MARK FROM ADR FROM PDL
      DC      GCMRK
GC370 LD      GC908     MOVE DOWN PDL
      S       GC918
      MDX     GC360
************** 6. REMOVE TRULY WORTHLESS ATOMS *****
GC400 LD      GC403+1
```

```
      STO     GC906
GC403 LD   L  #OBLS     GET ADR OF OBLIST
GC420 AND     GC916
      BSC  L  GC500,+-  BRANCH IF NIL
      STO     GC905
      BSI   3 XCAR-X
      BSI   3 XCDR-X
      BSC  L  GC430,+Z  BRANCH IF MARKED
      LD   I  GC905     REMOVE ATOM FROM OBLIST
      STO  I  GC906
      AND     GC916     UNMARK OBLIST NODE WHICH
      STO  I  GC905      WAS DISCONNECTED
      MDX     GC420
GC430 LD      GC905
      STO     GC906
      LD   I  GC905     (CDR GC905)
      MDX     GC420
****************************************************
GC912 DC      1+@MAJR   NORMAL PDL OVERFLOW ERR NO.
GC913 SLA     0         USED TO CONSTRUCT SHIFT
GC914 DC      L@FXS
GC915 DC      E@FXS-1
GC916 DC      /7FFF
GC917 DC      S@SPD
GC918 DC      1
GC919 DC      S@RPD
****************************************************
************* 7. CONSTRUCT FREE LISTS *************
GC500 SLT     32        CLEAR ACC AND EXT
      STO   3 @FSTL-X   CLEAR FREE LIST ADRS
      STO   3 @FXSL-X
      STO     GC710     CLEAR LIST COUNTERS
      STO     GC711
      LDX  L2 L@FST     COLLECT FREE STORAGE...
GC510 LD   L2 S@FST-2   GET A NODE
      BSC  L  GC520,+Z  BRANCH IF MARKED
      LD    3 @FSTL-X   ELSE ADD TO FREE LIST
      STD  L2 S@FST-2   NOTE - EXT IS STILL ZERO
      LD   L  2
      A       GC510+1
      STO   3 @FSTL-X
      MDX  L  GC710,2   INCR COUNTER
      MDX     GC530
GC520 AND     GC916     CLEAR BIT 0 IF MARKED
      STO  L2 S@FST-2
GC530 MDX   2 -2
      MDX     GC510
      LD      GC915     COLLECT FIXED-POINT LIST
      STO     GC908
      LD      GC914
      STO     GC906
GC540 LD      GC906     GET COUNTER
      S       GC918
      RTE     20
      SRA     12
      A       GC913     CONSTRUCT SHIFT TO GET BIT
      STO     GC550
      RTE     12
      SRA     4
      STO  L  2
      LD   L2 S@FXB     GET WORD OF BIT TABLE
GC550 SLA     *-*       GET BIT
      BSC  L  GC560,+Z  BRANCH IF MARKED
      LD    3 @FXSL-X   ADD TO FREE LIST
      STO  I  GC908
      LD      GC908
      STO   3 @FXSL-X
      MDX  L  GC711,1   INCR COUNTER
GC560 MDX  L  GC908,-1
      MDX  L  GC906,-1
      MDX     GC540
************* END OF GARBAGE COLLECTION ***********
```

```
GC700 LD      GC912     RESTORE NORMAL PDL
      STO  L  PSHA3      OVERFLOW ERROR CODE
GC705 MDX  L  #GCGA,0   SKIP IF GCGAG NIL
      MDX     GC720
      BSI   3 ERROR-X   PRINT MESSAGE
      DC      7+@INFO
GC710 DC      *-*
GC711 DC      *-*
GC720 LDX  L2 *-*       PUT TYPE CODE IN XR2
      BSC  I2 *         BRANCH ON TYPE
      DC      GC725     0 - INVOKED BY USER (GC)
      DC      GC730     1 - INVOKED BY CONS
      DC      GC750     2 - INVOKED BY MKFXN
**************************************************
GC725 LDX  I2 GC903     RESTORE XR2
      SRA     16        RETURN NIL AS VALUE
      BSI   3 POPJ-X    RETURN TO USER
**************************************************
GC730 LD    3 @FSTL-X   WAS ANY FST AREA COLLECTED
      BSC  L  GC740,Z   BRANCH IF SO
      BSI   3 ERROR-X   ELSE FATAL ERROR
      DC      5+@FATL
GC740 LDX  I2 GC903     RESTORE XR2
      BSC  I  GCFST
**************************************************
GC750 LD    3 @FXSL-X   WAS ANY FXS AREA COLLECTED
      BSC  L  GC760,Z   BRANCH IF SO
      BSI   3 ERROR-X   ELSE FATAL ERROR
      DC      6+@FATL
GC760 LDX  I2 GC903     RESTORE XR2
      BSC  I  GCFXS
**************************************************
      LIST
      HDNG    15 MARK TREE FOR GARBAGE COLLECTOR
**************************************************
*     GCMRK - MARK TREE FOR GARBAGE COLLECTOR    *
**************************************************
GCMRK AND      GCM92      CLEAR BIT 0 OF ARG
      STO      GCM99       AND SAVE IT
      S        GCM98
      BSC  L   GCM10,+Z
      S        GCM97
      BSC  L   GCM30,+Z   BRANCH IF IN FST
GCM10 LD       GCM99
      S        GCM96
      BSC  L   GCM20,+Z
      S        GCM95
      BSC  L   GCM60,+Z   BRANCH IF IN FXS
GCM20 BSI    3 POPJ-X     RETURN
GCM30 LD   I   GCM99      GET FIRST WORD OF NODE
      BSC      +Z
      BSI    3 POPJ-X     RETURN IF ALREADY MARKED
      OR       GCM94      ELSE MARK IT
      STO  I   GCM99
      LD       GCM99      IS IT AN ATOM (BIT 0
      A        GCM93       OF CAR SET)
      STO      GCM35+1
GCM35 LD   L   *-*
      BSC  L   GCM50,-    BRANCH IF NOT
      AND      GCM92
      BSC  L   GCM40,Z    BRANCH UNLESS PNAME NULL
      LD   I   GCM99      IS IT A (NULL) STRING
      S        GCM89       (IT IS NOW MARKED)
      BSC      +-          NO, SKIP FOR ERROR
      BSI    3 POPJ-X      YES, RETURN
      BSI    3 ERROR-X
      DC       15+@DUMP
GCM40 STO      GCM45+1    MARK PNAME
GCM45 LD   L   *-*
      OR       GCM94
      STO  I   GCM45+1
      AND      GCM92
      BSC  L   GCM40,Z
      LD   I   GCM99      GO MARK PROPERTY LIST
      MDX      GCMRK
GCM50 LD       GCM99      SAVE ARG ON PDL
      BSI    3 PUSHA-X
      BSI    3 XCAR-X     MARK CAR OF NODE
      BSI    3 PUSHJ-X
      DC       GCMRK
      BSI    3 POPA-X     POP ARG
      STO      GCM55+1    TAKE CDR (THIS IS A
GCM55 LD   L   *-*         FAST WAY)
      MDX      GCMRK      GO MARK CDR
GCM60 LD       GCM99      MARK WORD IN FXS
      S        GCM96
      RTE      20
      SRA      12
      OR       GCM91      CONSTRUCT SHIFT
      STO      GCM63
      RTE      12
      SRA      4
      A        GCM90      CONSTRUCT ADDRESS WITHIN
      STO      GCM65+1    BIT TABLE
      LD       GCM94
GCM63 SRA      *-*        PUT BIT IN PROPER POSITION
GCM65 OR   L   *-*        OR INTO BIT TABLE
      STO  I   GCM65+1
      MDX      GCM20
**************************************************
GCM89 DC       @STR+@ATOM
GCM90 DC       S@FXB
GCM91 SRA      0          USED TO CONSTRUCT SHIFT
GCM92 DC       /7FFF
GCM93 DC       1
```

```
GCM94 DC      /8000
GCM95 DC      E@FXS-S@FXS
GCM96 DC      S@FXS
GCM97 DC      E@FST-S@FST
GCM98 DC      S@FST
GCM99 DC      *-*
**************************************************
      LIST
      HDNG    16 XCONS, CONS, MKFXN
**************************************************
*     CONS XSUBR
**************************************************
XCNS1 STD     XCNS9     CDR IN ACC, CAR IN EXT
      LD    3 @FSTL-X
      BSI  L  GCFST,+-  GC IF NO FST LEFT
      LD    3 @FSTL-X   GET ADR OF FREE CELL
      STO     XCNS2+1
      LD   I  @FSTL     (SETQ @FSTL (CDR @FSTL))
      STO   3 @FSTL-X
      LDD     XCNS9     PUT GOODIES IN FREE CELL
XCNS2 STD  L  *-*
      SLT     32
      STD     XCNS9
      LD      XCNS2+1   RETURN ADR OF CELL
      BSC  I  XCONS
**************************************************
      BSS  E  0
XCNS9 DC      NIL       PROTECTED BY TEMLIST
      DC      NIL
**************************************************
*     CONS FUNCTION                              *
**************************************************
      DC      @LAM+2     (LAMBDA (X Y) ...
CONS  LD    3 @ARG1-X   PUT FIRST ARG IN EXT,
      RTE     16
      LD    3 @ARG2-X    SECOND IN ACC
      BSI   3 XCONS-X
      BSI   3 POPJ-X
**************************************************
*     MKFXN SUBROUTINE                           *
**************************************************
MKFX1 STO     MKFX9     SAVE NUMBER
      LD    3 @FXSL-X
      BSI  L  GCFXS,+-  GC IF NO FXS LEFT
      LD    3 @FXSL-X
      STO     MKFX2+1
      LD   I  @FXSL
      STO   3 @FXSL-X
      LD      MKFX9
MKFX2 STO  L  *-*
      LD      MKFX2+1   RETURN ADR OF NUMBER
      BSC  I  MKFXN
**************************************************
MKFX9 DC      *-*
**************************************************
      LIST
      HDNG    17 INPUT, TYI, READCH
*************************************************
*     SET UP INPUT FOR PROPER INPUT DEVICE      *
*************************************************
SETIN DC       *-*
      LD   I  SETIN     SET NAME OF CALLING FN
      MDX  L  SETIN,1
      STO     SETI2
      LD    3 @ARG1-X   GET ARG 1
SETI2 DC       *-*
      BSI   3 XNCHK-X   CHECK IT
      BSI   3 PUSHJ-X   CHECK IF DEFINED
      DC       IDEVP
      BSC  L  SETI5,+-  ERROR IF NOT
      LD   I  @ARG1
      STO  L  INDEV     SET DEVICE NUMBER
      BSC  I  SETIN
SETI5 LD    3 @ARG1-X   PRINT ERROR MESSAGE
      STO     SETI6
      BSI   3 ERROR-X
      DC       13+@MAJR
SETI6 DC       *-*       FAULTY DEVICE NUMBER
*************************************************
*     SET UP INPUT FOR SYSTEM INPUT DEVICE      *
*************************************************
SYSIN DC       *-*
      STO   3 @ARG1-X
SYSI2 LD   L  #SYSI     CHECK SYSIN
      BSI   3 XNMBP-X
      BSC  L  SYSI5,+-  BAD IF NOT A NUMBER
      LD   I  #SYSI     USE SYSIN
      BSC  L  SYSI5,+-  BRANCH IF BAD
      DC       IDEVP
      BSI   3 PUSHJ-X
      BSC  I  SYSIN
      STO  L  INDEV
SYSI5 LD    3 @SYSI-X   USE STANDARD INPUT
      BSC      +-Z
*************************************************
*     PEEK - PEEK AT AN INPUT CHARACTER         *
*************************************************
PEEK  DC       *-*
      STX   1 INPT5+1   SAVE XR1
      STX   2 PEEK6+1   SAVE XR2
      LDX  I3 $XR3X     SET XR3 FOR LIBFS
      LDX  I1 INDEV     PUT DEVICE NUMBER IN XR1
      LD   L1 INPKC     IS NEXT CHAR IN SAVE AREA
      BSI  I1 INSUB,+-  IF NOT, GET ONE
      STO  L1 INPKC     SAVE IT
      LDX  I1 INPT5+1   RESTORE XR1
PEEK6 LDX  L2 *-*       RESTORE XR2
      LDX  L3 X         RESTORE XR3
      BSC  I  PEEK
*************************************************
*     INPUT - INPUT A CHARACTER                 *
*************************************************
INPUT DC       *-*
      STX   1 INPT5+1   SAVE XR1
      STX   2 INPT6+1   SAVE XR2
      LDX  I3 $XR3X     SET XR3 FOR LIBFS
      LDX  L1 *-*       PUT DEVICE NUMBER IN XR1
INDEV EQU      *-1      DEV NUMBER MAY BE CHANGED
      LD   L1 INPKC     IS NEXT CHAR IN SAVE AREA
      BSI  I1 INSUB,+-  IF NOT, GET ONE
      RTE      16
      SRA      16
      STO  L1 INPKC     CLEAR SAVE BUFFER
      RTE      16
INPT5 LDX  L1 *-*       RESTORE XR1
INPT6 LDX  L2 *-*       RESTORE XR2
      LDX  L3 X         RESTORE XR3
      BSC  I  INPUT
```

```
***************************************************
*     READ EOF ERROR HANDLER                      *
***************************************************
RDEOF LD       INDEV     GET DEVICE NUMBER
      STO      RDE07
      LDX  I1 INPT5+1   RESTORE XR1 FOR PDL
      LDX  L3 X         RESTORE XR3
      LDS      1
      STS  L  ERR56     SET SWITCH FOR NO FLUSH
      BSI   3 ERROR-X   SIGNAL ERROR
      DC       37+@MAJR
***************************************************
RDE07 DC       *-*
***************************************************
*     READCH, READC, PEEKCH, PEEKC                *
***************************************************
      DC       @LAM+1 (LAMBDA (X) ...
REDCH BSI  L  SETIN
      DC       #RDCH
      LDS      0
      MDX      RDCH1
***************************************************
      DC       @LAM+1 (LAMBDA (X) ...
READC BSI  L  SETIN
      DC       #REDC
      LDS      1
      MDX      RDCH1
***************************************************
      DC       @LAM+1 (LAMBDA (X) ...
PEKCH BSI  L  SETIN
      DC       #PKCH
      LDS      2
      MDX      RDCH1
***************************************************
      DC       @LAM+1 (LAMBDA (X) ...
PEEKC BSI  L  SETIN
      DC       #PEKC
      LDS      3
RDCH1 STS      RDCH2     SAVE INDICATOR
      LD       RDCH2
      STO      RDCH5
RDCH2 LDS      *-*
      BSC  L  RDCH3,C
RDCHX BSI  L  INPUT     INPUT CHAR
      BSC  L  RDCHX,+Z  TRY AGAIN IF KEYBOARD KILL
      MDX      RDCH4
RDCH3 BSI  L  PEEK      PEEK AT CHAR
      BSC  L  RDCH4,-   BRANCH UNLESS KEYBOARD KILL
      BSI  L  INPUT     INPUT CHAR
      MDX      RDCH3     GO PEEK AGAIN
RDCH4 SRT      16
      BSI   3 XCONS-X   MAKE CHAR LIST
      OR       RDCH9     SET ATOM MARK
      RTE      16
      LD       RDCH8     SET STRING VALUE MARKER
      BSI   3 XCONS-X
RDCH5 LDS      *-*
      BSC  L  RDCH6,O
      STO   3 @ARG1-X   INTERN STRING IF
      BSI   3 PUSHJ-X    READCH OR PEEKCH
      DC       INTRN
RDCH6 BSI   3 POPJ-X
***************************************************
RDCH8 DC       @STR
RDCH9 DC       @ATOM
***************************************************
*     TYI FUNCTION                                *
***************************************************
      DC       @LAM+1    (LAMBDA (X) ...
TYI   BSI  L  SETIN     SET INPUT DEVICE
      DC       #TYI
      BSI  L  INPUT     READ A CHAR
      STO      TYI1+1
TYI1  LD    L  *-*      GET EBCDIC
      SRA      8
      BSI    3 MKFXN-X  MAKE IT A NUMBER
      BSI    3 POPJ-X
***********************************************
*     INTERN FUNCTION                         *
***********************************************
      DC       @LAM+1   (LAMBDA (X) ...
INTRN LD     3 @ARG1-X  CHECK FOR VALID ARGUMENT
      BSC      +-
      BSI    3 POPJ-X   OK IF NIL (RETURN)
      S        INT99
      BSC   L  INT10,+Z
      S        INT98
      BSC   L  INT10,-  BAD IF OUTSIDE FREE STG
      A        INT97
      STO      INT05+1
INT05 LD    L  *-*      GET CAR
      BSC   L  INT15,+Z BAD IF NOT ATOM
INT10 LD     3 @ARG1-X
      STO      INT12
      BSI    3 ERROR-X
      DC       17+@MAJR
INT12 DC       *-*
INT15 STO      INT96    SAVE ADR OF PNAME STRING
      SLA      1
      BSC   L  INT20,Z  ERROR IF PNAME NULL
      BSI    3 ERROR-X
      DC       18+@MAJR
INT20 LD       INT95    GET ADR OF ATOM OBLIST
INT25 STO      INT94
      LD       INT96
      STO      INT90    SAVE ADR OF PNAME OF ARG
      LD    I  INT94
      BSC   L  INT45,+- BRANCH IF END OF OBLIST
      BSI    3 XCAR-X
      STO      INT93    SAVE ADR OF ATOM
      BSI    3 XCAR-X
      BSC   L  INT30,Z  ERROR IF NULL
      BSI    3 ERROR-X
      DC       12+@DUMP
INT30 STO      INT92
      BSC   L  INT35,Z  BRANCH IF MORE LETTERS
      MDX   L  INT90,0  SKIP IF NONE LEFT IN ARG
      MDX      INT40    TEST NEXT ATOM
      LD       INT93    CHECK IF NIL
      S        INT87
      BSC      Z        RETURN CLEAR ACC IF SO
      LD       INT93    RETURN EQUIVALENT ATOM
      BSI    3 POPJ-X
INT35 LD       INT92    GET CHAR OF ATOM
      BSI    3 XCAR-X
      STO      INT91
      LD       INT90    GET CHAR OF ARG
      BSC   L  INT45,+-
      BSI    3 XCAR-X
      S        INT91
      BSC   L  INT45,+Z BRANCH TO INSERT
      BSC   L  INT40,Z  BRANCH TO TEST NEXT ATOM
      LD    I  INT90    TEST NEXT CHAR
      STO      INT90
      LD    I  INT92
      MDX      INT30
INT40 LD    I  INT94    TEST NEXT ATOM ON OBLIST
      MDX      INT25
***********************************************
INT82 DC       #C@R
INT83 DC       @R
INT84 DC       @D-@A
INT85 DC       @A
INT86 DC       @C
```

```
INT87 DC     #NIL
INT88 DC     NIL        PROTECTED BY TEMLIST
INT89 DC     @UNDF
INT90 DC     *-*
INT91 DC     *-*
INT92 DC     *-*
INT93 DC     *-*
INT94 DC     *-*
INT95 DC     #OBLS
INT96 DC     *-*
INT97 DC     1+E@FST
INT98 DC     E@FST-S@FST
INT99 DC     S@FST
**************************************************
INT45 LDD  I  @ARG1     CREATE NEW ATOM WITH
      LD      INT89      UNDEFINED VALUE, AND
      BSI   3 XCONS-X    PROTECT IT FROM
      STO     INT88      GARBAGE COLLECTION
      BSI   3 XCAR-X    GET PRINT NAME
      STO     INT90     SAVE, CHECK FOR C-R ATOM
      BSI   3 XCAR-X
      EOR     INT86     IS FIRST CHAR C
      BSC  L  INT60,Z   BRANCH IF NOT
INT50 LD   I  INT90     GET REST OF CHARS
      STO     INT90
      BSC  L  INT60,+-  BRANCH IF NONE LEFT
      LD   I  INT90
      BSC  L  INT55,+-  BRANCH IF ONLY ONE LEFT
      LD      INT90     TEST NEXT CHAR
      BSI   3 XCAR-X
      S       INT85     IS IT A
      BSC     Z
      S       INT84     IS IT D
      BSC  L  INT60,Z   BRANCH IF NEITHER
      MDX     INT50
INT55 LD      INT90     TEST LAST CHAR
      BSI   3 XCAR-X
      EOR     INT83     IS IT R
      BSC  L  INT60,Z   BRANCH IF NOT
      LD      INT82     GIVE THE ATOM A SPECIAL
      RTE     16         C-R FUNCTION VALUE
      LD      INT88
      BSI   3 XCONS-X
      STO  I  INT88
INT60 LD      INT88
      RTE     16
      LD   I  INT94     INSRT NEW ATOM INTO OBLIST
      BSI   3 XCONS-X
      STO  I  INT94
      LD      INT88     GET NEW ATOM
      RTE     16        SAVE IN EXT
      SRA     16
      STO     INT88     RESET PROTECTED LOC TO NIL
      RTE     16        GET ATOM FROM EXT
      BSI   3 POPJ-X
**************************************************
```

```
      EJCT
*************************************************
*     READ FUNCTION                             *
*************************************************
      DC       @LAM+1     (LAMBDA (X) ...
READ  BSI  L  SETIN       SET INPUT DEVICE
      DC       #READ
RD005 STX   2 RD035+1     SAVE XR2 (READSTR ENTRY)
      LDX  I2 INDEV       (SETIN SET INDEV PROPERLY)
RD010 BSI  L  PEEK        PEEK AT A CHAR
      S        RD901
      BSC      Z
      S        RD902
      BSC  L  RD020,Z     BRANCH UNLESS BLANK OR )
      STO  L2 INPKC       ELSE FORCE NEXT PEEK TO
      MDX      RD010       GET A NEW CHAR
RD020 SRA      16         SET REDSW SO EOF CARD
      STO      REDSW       IS AN ERROR
RD025 BSI   3 PUSHJ-X     CALL RECURSIVE
      DC       RD050       S-EXPRESSION READER
      BSC  L  RD010,+Z    TRY AGAIN IF KEYBOARD KILL
      STX      REDSW      RESET REDSW
RD035 LDX  L2 *-*         RESTORE XR2
      BSI   3 POPJ-X
*************************************************
RD901 DC       @
RD902 DC       @RPAR-@
*************************************************
REDSW DC       *-*+*      0 MEANS EOF CARD IS ERROR
*************************************************
RD050 BSI  L  INPUT       INPUT A CHAR
      BSC      +Z
      BSI   3 POPJ-X      POP OUT IF KEYBOARD KILL
      STO      RD910      SAVE CHAR
      S        RD901
      BSC  L  RD050,+-    TRY AGAIN IF BLANK
      S        RD911
      BSC      Z
      S        RD912
      BSC  L  RD060,Z     BRANCH UNLESS . OR )
      STX   2 RD055
      BSI   3 ERROR-X     ERROR - INVALID EXPRESSION
      DC       20+@MAJR
RD055 DC       *-*
RD060 S        RD913
      BSC  L  RD600,+-    BRANCH IF (
      S        RD914
      BSC  L  RD580,+-    BRANCH IF ' (QUOTE)
      S        RD915
      BSC  L  RD500,+-    BRANCH IF , (COMMA)
      MDX      RD100
*************************************************
RD910 DC       *-*
RD911 DC       @PER-@
RD912 DC       @RPAR-@PER
RD913 DC       @LPAR-@RPAR
RD914 DC       @QUOT-@LPAR
RD915 DC       @COMA-@QUOT
*************************************************
RD100 SLT      32
      BSI   3 XCONS-X     CREATE ATOM HEADER
      STO      RD920      SAVE ADR FOR PROTECTION
      A        RD921
      STO      RD922      SAVE ADR TO APPEND CHARS
      STX   2 RD200+1     SAVE XR2
      LDX   2 7           SET BITS IN CHAR TYPE SW
      STX   2 RD923        (ONLY 2 BITS ARE NOW USED)
      LDX   2 3           SET XR2 TO SHIFT FIRST CHAR
      LD       RD910      CHECK CHAR
RD110 S        RD924
      BSC  L  RD120,Z     BRANCH UNLESS AMPERSAND
      STO      RD923      ZERO CHAR TYPE SWITCH
      BSI  L  INPUT      USE NEXT CHAR
      STO     RD910
      BSC     +Z
      BSI   3 POPJ-X     POP OUT IF KEYBOARD KILL
RD120 LD   I  RD910      GET INDICATOR BITS FOR CHAR
      SRA   2            SHIFT 3 IF 1ST CHAR, ELSE 0
      AND     RD923      AND OVER BITS ALREADY THERE
      STO     RD923
      LD      RD910      APPEND CHAR TO PNAME
      SRT     16
      BSI   3 XCONS-X
      MDX   2 0          IF FIRST CHAR
      OR      RD925       OR IN ATOM MARK TO ADR
      STO  I  RD922      PUT ADR IN LAST NODE
      STO     RD922      SAVE AS ADR FOR NEXT APPEND
      LDX   2 0          ZERO XR2 FOR OTHER CHARS
      BSI  L  PEEK       PEEK AT NEXT CHAR
      S       RD901      IS IT BLANK
      BSC     Z
      S       RD911      IS IT . (DOT)
      BSC     Z
      S       RD912      IS IT )
      BSC     Z
      S       RD913      IS IT (
      BSC     Z
      S       RD914      IS IT ' (QUOTE)
      BSC     Z
      S       RD915      IS IT , (COMMA)
      BSC  L  RD200,+-   BRANCH IF ANY ONE OF THEM
      BSI  L  INPUT      INPUT THE CHAR PEEKED AT
      STO     RD910      SAVE IT
      BSC  L  RD110,-
      BSI   3 POPJ-X     POP OUT IF KEYBOARD KILL
****************************************************
RD920 DC      NIL        PROTECTED BY TEMLIST
RD921 DC      1
RD922 DC      *-*
RD923 DC      *-*
RD924 DC      @AMPR
RD925 DC      @ATOM      ALSO CONSTANT /8000
****************************************************
RD200 LDX  L2 *-*        RESTORE XR2
      LD      RD923      CHECK CHAR TYPE SWITCH
      BSC  L  RD250,E    BRANCH IF DECIMAL NUMBER
      SRA     2
      BSC  L  RD300,E    BRANCH IF HEX NUMBER
RD210 LD      RD920      SET UP ARG FOR INTERN
      STO   3 @ARG1-X
      SRA     16         CLEAR PROTECTED LOC TO NIL
      STO     RD920
      BSC  L  INTRN      INTERN ATOM AS RESULT
RD250 MDX  L  RD922,1    IS LAST CHAR + OR -
      LD   I  RD922       (IF SO, IT IS ONLY CHAR)
      S       RD930
      BSC     Z
      S       RD931
      BSC  L  RD210,+-   IF SO, BRANCH TO INTERN IT
      STX     RD932      SET SIGN SWITCH FOR +
      SRA     16
      STO     RD933      CLEAR VALUE
      LD      RD920
      STO     RD285      SAVE ATOM HEADER FOR ERROR
      BSI   3 XCAR-X     GET ADR OF PNAME STRING
      STO     RD922
      BSI   3 XCAR-X     CHECK FIRST CHAR
      S       RD930
      BSC  L  RD260,+-   BRANCH IF +
      S       RD931
      BSC  L  RD265,Z    BRANCH UNLESS -
      STO     RD932      CLEAR SIGN SWITCH FOR -
RD260 LD   I  RD922      SKIP FIRST CHAR
      STO     RD922
```

```
RD265 LD      RD922
RD267 BSI   3 XCAR-X    GET DIGIT
      S       RD934
      STO     RD935+1   PUT IN WD 2 OF 2-WORD ZERO
      LD      RD933
      M       RD936     MULTIPLY OLD VALUE BY 10
      AD      RD935     ADD IN NEW DIGIT
      BSC   L RD280,Z   BRANCH IF OVERFLOW
      SLT     16
      STO     RD933     SAVE VALUE
      LD    I RD922     CHAIN DOWN LIST OF DIGITS
      STO     RD922
      BSC   L RD267,Z   BRANCH UNLESS DONE
      STO     RD920     CLEAR PROTECTED LOC TO NIL
      LD      RD933     GET VALUE
      BSC   L RD270,-   OKAY IF NON-NEGATIVE
      SLA     1
      BSC   L RD280,Z   ERROR IF LARGER THAN 32768
      MDX   L RD932,0   ERROR IF EXACTLY 32768
      MDX     RD280      BUT SIGN IS +
RD270 LD      RD933     GET VALUE
      MDX   L RD932,0   SKIP IF SIGN SWITCH -
      MDX     RD273
      SRA     16        COMPLEMENT NUMBER
      S       RD933
RD273 BSI   3 MKFXN-X   MAKE A NUMBER
      BSI   3 POPJ-X    POP OUT
RD280 STX   2 RD287
      BSI   3 ERROR-X   ERROR - NUMBER OVERFLOW
      DC      22+@MINR
RD285 DC      *-*
RD287 DC      *-*
      LD      RD925
      LD      RD925     USE -32768 IF NEGATIVE
      MDX   L RD932,0
      LD      RD937     USE 32767 IF POSITIVE
      MDX     RD273
**************************************************
RD930 DC      @PLUS
RD931 DC      @DASH-@PLUS
RD932 DC      *-*
RD933 DC      *-*
RD934 DC      @0
RD935 DEC     0         TWO-WORD ZERO, EVEN LOC
RD936 DC      10
RD937 DC      /7FFF
**************************************************
RD300 MDX   L RD922,1   IS LAST CHAR /
      LD    I RD922      (IF SO, IT IS ONLY CHAR)
      S       RD940
      BSC   L RD210,+-  IF SO, BRANCH TO INTERN
      LD      RD920
      STO     RD325     SAVE ATOM HEADER FOR ERROR
      BSI   3 XCAR-X    GET ADR OF PNAME STRING
      BSI   3 XCDR-X    SKIP FIRST CHAR
      STO     RD932     SAVE ADR OF STRING
      SRA     16
      STO     RD920     CLEAR PROTECTED LOC TO NIL
      STO     RD933     CLEAR VALUE
      LDX   2 4         SET COUNTER FOR 4 DIGITS
RD310 LD      RD933     SHIFT VALUE OVER BY 4 BITS
      SLA     4          (MULTIPLY BY 16)
      STO     RD933
      LD      RD932
      BSI   3 XCAR-X    GET NEXT DIGIT
      S       RD934
      BSC     +Z        SKIP IF 0-9
      A       RD941     ELSE CORRECT FOR A-F
      OR      RD933     OR NEW DIGIT INTO VALUE
      STO     RD933
      LD    I RD932     CHAIN DOWN LIST OF DIGITS
      BSC   L RD320,+-  BRANCH IF DONE
      STO     RD932
      MDX   2 0         SKIP IF XR2 ALREADY 0
      MDX   2 -1        ELSE DECR BY 1
      NOP
      MDX     RD310
RD320 MDX   2 0         SKIP IF XR2 IS ZERO
      MDX     RD330
      LDX  I2 RD200+1   RESTORE XR2
      STX   2 RD327     MORE THAN 4 HEX DIGITS WERE
      BSI   3 ERROR-X    INPUT - PRINT WARNING
      DC      42+@MINR
RD325 DC      *-*
RD327 DC      *-*
RD330 LDX  I2 RD200+1   RESTORE XR2
      LD      RD933     GET VALUE, GO MAKE A NUMBER
      MDX     RD273      AND POP OUT
*************************************************
RD940 DC      @SLSH
RD941 DC      @0-@A+10
*************************************************
RD500 LDD     RD960     CREATE STRING HEADER
      BSI   3 XCONS-X
      STO     RD964     SAVE ADR FOR PROTECTION
      A       RD961
      STO     RD962     SAVE ADR TO APPEND CHARS
RD510 BSI  L  INPUT     INPUT A CHAR
      BSC     +Z
      BSI   3 POPJ-X    POP OUT IF KEYBOARD KILL
      S       RD963
      BSC  L  RD530,+-  BRANCH IF COMMA
RD520 A       RD963
      SRT     16        APPEND CHAR TO PNAME
      BSI   3 XCONS-X
      SRT     15        RD962 IS ODD IF AND ONLY
      LD      RD962      IF THIS IS FIRST CHAR -
      SLT     15         THIS PROVIDES ATOM MARK
      STO  I  RD962
      STO     RD962     SAVE FOR NEW APPEND ADR
      MDX     RD510
RD530 BSI  L  PEEK      PEEK AT NEXT CHAR
      S       RD963
      BSC  L  RD540,Z   BRANCH UNLESS COMMA
      STO  L2 INPKC     ERASE CHAR BUFFERED BY PEEK
      MDX     RD520     GO USE COMMA AS NEXT CHAR
RD540 LD      RD964     SAVE ADR OF STRING
      STO     RD962
      SRA     16
      STO     RD964
      LD      RD962     RETURN STRING
      BSI   3 POPJ-X
*************************************************
      BSS  E  0
RD960 DC      @STR
      DC      /8000
RD961 DC      1
RD962 DC      *-*
RD963 DC      @COMA
RD964 DC      NIL       PROTECTED BY TEMLIST
*************************************************
RD580 BSI   3 PUSHJ-X   READ S-EXPRESSION
      DC      RD050      FOLLOWING QUOTE
      BSC     +Z
      BSI   3 POPJ-X    POP OUT IF KEYBOARD KILL
      SRT     16
      BSI   3 XCONS-X   FORM LIST
      RTE     16
      LD      RD968
      RTE     16
      BSI   3 XCONS-X   FORM LIST (QUOTE S-EXPR)
      BSI   3 POPJ-X
*************************************************
RD968 DC      #QUOT
```

```
****************************************************
RD600 BSI   3 PUSHA-X   PUT A NIL (EMPTY LIST)
      LD    L 1          ON STACK, AND THEN ADR OF
      BSI   3 PUSHA-X    THAT NIL FOR APPENDING
RD610 BSI   L PEEK      PEEK AHEAD FOR A NON-BLANK
      BSC   L RD700,+Z  BRANCH IF KEYBOARD KILL
      S       RD971
      BSC   L RD620,Z
      STO  L2 INPKC     FORCE NEXT PEEK TO NEW CHAR
      MDX     RD610
RD620 S       RD972
      BSC   L RD680,+-  BRANCH IF )
      S       RD973
      BSC   L RD630,+-  BRANCH IF . (DOT)
      BSI   3 PUSHJ-X   READ AN S-EXPRESSION
      DC      RD050
      BSC   L RD700,+Z  BRANCH IF KEYBOARD KILL
      SRT     16
      BSI   3 XCONS-X
      STO  I1 0         APPEND TO LIST
      STO   1 0         SAVE ADR FOR NEXT APPEND
      MDX     RD610
RD630 LD    1 1         BRANCH UNLESS NO
      BSC   L RD640,Z    ITEM PRECEDED DOT
      STX   2 RD635
      BSI   3 ERROR-X   PRINT ERROR MESSAGE
      DC      23+@MAJR
RD635 DC      *-*
RD640 SRA     16
      STO  L2 INPKC     CLEAR PEEKED DOT
      BSI   3 PUSHJ-X   READ S-EXPRESSION
      DC      RD050
      BSC   L RD700,+Z  BRANCH IF KEYBOARD KILL
      STO  I1 0         PATCH ONTO END OF LIST
      BSI   3 POPA-X    POP OFF TEMP ADR FOR APPEND
RD650 BSI   L PEEK      PEEK AHEAD FOR A NON-BLANK
      S       RD971
      BSC   L RD660,Z
      STO  L2 INPKC     FORCE NEXT PEEK TO NEW CHAR
      MDX     RD650
RD660 S       RD972     IS CHAR A )
      BSC   L RD670,Z   BRANCH IF NOT
      STO  L2 INPKC
RD665 BSI   3 POPA-X    POP OFF NEWLY CREATED LIST
      BSI   3 POPJ-X    POP OUT
RD670 BSI   3 POPA-X    POP LIST OFF STACK
      STO     RD675      AND USE IN ERROR MESSAGE
      STX   2 RD677
      BSI   3 ERROR-X
      DC      21+@MAJR
RD675 DC      *-*
RD677 DC      *-*
RD680 STO  L2 INPKC     CLEAR PEEKED ) (ACC = 0)
      BSI   3 POPA-X    POP OFF TEMP ADR FOR APPEND
      MDX     RD665
RD700 STO     RD962     SAVE KEYBOARD KILL FLAG
      BSI   3 POPA-X    POP TWO THINGS OFF STACK
      BSI   3 POPA-X
      LD      RD962     GET FLAG
      BSI   3 POPJ-X
****************************************************
RD971 DC      @
RD972 DC      @RPAR-@
RD973 DC      @PER-@RPAR
****************************************************
      LIST
      HDNG    19 KEYBOARD INPUT DEVICE HANDLER
*************************************************
*     KEYBOARD HANDLER                          *
*************************************************
      AIF     (@KBRD EQ YES),.YES
IKBRD EQU     0
FKBRD EQU     0
      AGO     .NO
.YES  ANOP
IKBRD DC      *-*
      MDX  L  IKB98,0   SKIP IF NO CHAR IN BUFFER
      MDX     IKB55
IKB05 LD   L  #DDTI
      STO     IKB95     INIT END SWITCH
IKB07 LD      IKB96     INIT BUFFER POINTER
      STO     IKB97
      STX     IKB86     RESET BS SWITCH
      SRA     16
      STO     IKB98     INIT BUFFER COUNTER
IKB10 LD      *-1       PUT ODD NUMBER IN ACC
      LIBF    KBCP0     READ A CHAR
      RTE     20
      AND     IKB99
      BSC  L  IKB30,+-  BRANCH UNLESS CONTROL
      SLA     1
      BSC  L  IKB15,Z   BRANCH UNLESS CR
      STX     IKB95     SET END SWITCH
      MDX     IKB38     USE BLANK FOR CHAR
IKB15 MDX  L  IKB98,0
      BSC     +-Z
      MDX     IKB10     BRANCH IF NO CHARS IN BUF
      BSC  L  IKB25,-   BRANCH UNLESS BS
      MDX  L  IKB97,-1  DECR POINTER AND COUNTER
      MDX  L  IKB98,-1
      NOP
      LD      IKB93
      MDX  L  IKB86,0   SKIP IF LAST CHAR WAS BS
      BSI     IKB70     ECHO NUMBER SIGN
      SRA     16        SET SW - LAST CHAR WAS BS
      STO     IKB86
      LD   I  IKB97
      BSI     IKB70     ECHO LAST CHAR TYPED
      MDX  L  IKB98,0
      MDX     IKB10
      LD      IKB93     IF NONE LEFT ECHO NUMBER,CR
      BSI     IKB70
      LD      IKB94
      BSI     IKB70
      MDX     IKB10
IKB25 LD      IKB93     ECHO NUMBER, NUMBER, CR
      BSI     IKB70
      LD      IKB93
      BSI     IKB70
      LD      IKB94
      BSI     IKB70
      MDX     IKB07     REINITIALIZE FOR INPUT
IKB30 RTE     12
      STO     IKB92     SAVE CHAR
      S       IKB89     IS IT A 0-8-2
      BSC  L  IKB60,+-  BRANCH IF SO
      LDX   2 -L@EBC    SEARCH TABLE
IKB35 LD   L2 CRDTB+L@EBC
      EOR     IKB92
      BSC  L  IKB40,+-
      MDX   2 1
      MDX     IKB35
IKB38 LDX   2 -L@EBC    USE BLANK IF NOT FOUND
IKB40 LD   L  2         CALCULATE ADR
      A       IKB91
IKB45 STO  I  IKB97     PUT IN BUFFER
      LD      IKB93
      MDX  L  IKB86,0   SKIP IF LAST CHAR WAS BS
      BSC      +-Z
      BSI      IKB70     IF SO, ECHO NUMBER SIGN
      STX      IKB86     RESET BS SWITCH
      LD   I   IKB97     GET CHAR AGAIN
      MDX  L   IKB97,1
      MDX  L   IKB98,1
      BSI      IKB70     ECHO IT
      MDX  L   IKB95,0
      MDX      IKB50     BRANCH IF END SW SET
      LD       IKB98
      S        IKB90
      BSC  L   IKB10,+Z  BRANCH UNLESS BUFFER FULL
IKB50 MDX  L   #DDTI,0
      MDX      IKB52
      LD       IKB94     ECHO CR
      BSI      IKB70
IKB52 LD       IKB96     SET ADR FOR CHAR FETCH
      STO      IKB97
IKB55 LD   I   IKB97     GET CHAR FROM BUFFER
      MDX  L   IKB97,1
      MDX  L   IKB98,-1
      NOP
      BSC  I   IKBRD
IKB60 LD       IKB93     ECHO OR,QUES,OR
      BSI      IKB70
      LD       IKB87
      BSI      IKB70
      LD       IKB93
      BSI      IKB70
      LD       IKB88     USE -1 FOR KEYBOARD KILL
      STX      IKB95     SET END SWITCH
      MDX      IKB45
*************************************************
IKB86 DC       *-*+*     ZERO = LAST CHAR WAS BS
IKB87 DC       @QUES
IKB88 DC       -1
IKB89 DC       /2820     0-8-2
IKB90 DC       L@KBD     LENGTH OF BUFFER
IKB91 DC       EBCTB+L@EBC
IKB92 DC       *-*
IKB93 DC       @NMBR
IKB94 DC       @CR
IKB95 DC       *-*
IKB96 DC       IKBBF
IKB97 DC       *-*
IKB98 DC       0         CHAR COUNTER
IKB99 DC       /E000
*************************************************
IKB70 DC       *-*       ECHO CHAR UNLESS KBECHO NIL
IKB72 MDX  L   #KBEC,0
      BSC      +-Z
      MDX      IKB75
      LDX   2 1          SET OUTPUT FOR TYPEWRITER
      STX  L2 OUTDV
      BSI  L  OUTPT
      LDX  I3 $XR3X      SET XR3 FOR LIBFS
IKB75 BSC  I  IKB70
*************************************************
FKBRD DC       *-*
      LD       IKB94     PRINT CARRIAGE RETURN
      BSI      IKB70
      LD       IKB96     RESET BUFFER POINTER
      STO      IKB97
      SRA      16
      STO      IKB98     RESET CHAR COUNTER
      STO  L   INPKC+6   CLEAR PEEK BUFFER
      BSC  I  FKBRD
*************************************************
IKBBF BSS      100       KEYBOARD INPUT BUFFER
L@KBD EQU      *-IKBBF   LENGTH OF BUFFER
*************************************************
.NO   ANOP
```

```
      HDNG    EVAL FUNCTION
*************************************************
*     EVAL FUNCTION                             *
*************************************************
      DC      @LAM+1    (LAMBDA (ARG) ...
EVAL  LD    3 @ARG1-X   GET ARG
      BSC     +-        SKIP UNLESS NIL
      BSI   3 POPJ-X    RETURN NIL IF ARG IS NIL
      S       EV901
      BSC  L  EV010,+Z  BRANCH IF NUMBER
      S       EV902
      BSC  L  EV020,+Z  BRANCH UNLESS NUMBER
EV010 LD    3 @ARG1-X   RETURN ARG (NUMBER/STRING)
      BSI   3 POPJ-X
EV020 LD    3 @ARG1-X   GET CAR OF ARG
      A       EV903
      STO     EV030+1
EV030 LD   L  *-*
      BSC  L  EV100,-   BRANCH UNLESS ATOM
      LD   I  @ARG1     IS IT AN UNDEFINED ATOM
      S       EV903
      BSC  L  EV050,Z   BRANCH IF NOT
      LD    3 @ARG1-X
      STO     EV040
      BSI   3 ERROR-X   UNBOUND VARIABLE ERROR
      DC      23+@MAJR
EV040 DC      *-*
EV050 S       EV903     IS IT A STRING
      BSC  L  EV010,+-  BRANCH IF SO
      LD   I  @ARG1     RETURN VALUE OF ATOM
      BSI   3 POPJ-X
*************************************************
EV901 DC      S@FST
EV902 DC      E@FST-S@FST
EV903 DC      1
*************************************************
EV100 BSC     +-        RESULT IS NIL IF
      BSI   3 POPJ-X     (CAR ARG) IS NIL
      STO     EV120     SAVE (CAR ARG)
      S       EV901
      BSC  L  EV110,+Z  BRANCH IF NUMBER
      S       EV902
      BSC  L  EV130,+Z  BRANCH UNLESS NUMBER
EV110 BSI   3 ERROR-X   INVALID FUNCTION ERROR
      DC      24+@MAJR
EV120 DC      *-*
EV130 LD      EV120     GET (CAR ARG)
      A       EV903
      STO     EV140+1
EV140 LD   L  *-*
      BSC  L  EV200,-   BRANCH IF NOT AN ATOM
      LD   I  EV120     GET VALUE OF ATOM
      S       EV903
      BSC  L  EV160,Z   BRANCH UNLESS UNDEFINED
      LD      EV120
      STO     EV150
      BSI   3 ERROR-X   UNDEFINED FUNCTION ERROR
      DC      25+@MAJR
EV150 DC      *-*
EV160 S       EV903     IS IT A CHAR STRING
      BSC  L  EV110,+-  BRANCH IF SO (ERROR)
      LD   I  EV120     GET VALUE OF (CAR ARG)
      RTE     16
      LD   I  @ARG1     GET (CDR ARG)
      BSI   3 XCONS-X   CONS THEM
      STO   3 @ARG1-X   SAVE AS NEW ARG
      MDX     EVAL       AND EVALUATE
EV200 S       EV930     TEST (CAAR ARG)
      BSC  L  EV250,+-  BRANCH IF SUBR
      S       EV931
      BSC  L  EV300,+-  BRANCH IF LAMBDA
      S       EV932
      BSC  L  EV350,+-  BRANCH IF NLAMBDA
      S       EV933
      BSC  L  EV400,+-  BRANCH IF MLAMBDA
      S       EV934
      BSC  L  EV300,+-  BRANCH IF C-R
      S       EV935
      BSC  L  EV500,+-  BRANCH IF LABEL
      LD   I  @ARG1     GET (CDR ARG)
      BSI   3 PUSHA-X   SAVE ON STACK
      LD      EV120     GET (CAR ARG)
      STO   3 @ARG1-X
      BSI   3 PUSHJ-X   EVALUATE IT
      DC      EVAL
      RTE     16
      BSI   3 POPA-X    POP (CDR ARG) OFF STACK
      BSI   3 XCONS-X   CONS THEM
      STO   3 @ARG1-X   SAVE AS NEW ARG
      MDX     EVAL       AND EVALUATE
**************************************************
EV930 DC      #SUBR
EV931 DC      #LAM-#SUBR
EV932 DC      #NLAM-#LAM
EV933 DC      #MLAM-#NLAM
EV934 DC      #C@R-#MLAM
EV935 DC      #LABL-#C@R
**************************************************
EV250 LD   I  EV120     GET (CDAR ARG)
      STO     EV260+1
EV260 LD   L  *-*       GET WORD POINTED TO
      SRA     14        GET TOP TWO BITS
      A       EV940
      STO     EV270+1
EV270 BSC  I  *-*       BRANCH ON TWO BITS
**************************************************
EV940 DC      EV941
EV941 DC      EV300     00 = LAMBDA
      DC      EV350     01 = NLAMBDA
      DC      EV400     10 = MLAMBDA
      DC      EV400     11 = UNASSIGNED (MLAMBDA)
**************************************************
EV300 LD      EV120     GET (CAR ARG)
      BSI   3 PUSHA-X   PUSH ON STACK
      LD   I  @ARG1     GET (CDR ARG)
      BSI   3 PUSHA-X   PUSH ON STACK
      SRA     16
      BSI   3 PUSHA-X   PUSH NIL ON STACK
      LD   L  1
      BSI   3 PUSHA-X   PUSH ADR OF NULL LIST
      LD    1 2         GET (CDR ARG) LIST OF FORMS
EV310 BSC  L  EV330,+-  DONE IF ALL FORMS EVALED
      STO   1 2         SAVE LIST OF FORMS
      A       EV950
      STO     EV320+1
EV320 LD   L  *-*       GET CAR OF LIST
      STO   3 @ARG1-X
      BSI   3 PUSHJ-X   EVAL IT
      DC      EVAL
      RTE     16
      SRA     16
      BSI   3 XCONS-X   CONS IT WITH NIL
      STO  I1 0         APPEND TO LIST OF RESULTS
      STO   1 0         SAVE ADR FOR NEXT APPEND
      LD   I1 2         TAKE CDR OF LIST OF FORMS
      MDX     EV310
EV330 BSI   3 POPA-X    POP OFF ADR FOR APPENDING
      BSI   3 POPA-X    POP OFF LIST OF RESULTS
      STO   3 @ARG2-X   SAVE AS ARG 2 FOR APPLY
      BSI   3 POPA-X    POP OFF LIST OF FORMS
      BSI   3 POPA-X    POP OFF FUNCTION
      STO   3 @ARG1-X   SAVE AS ARG 1 FOR APPLY
      BSC  L  APPLY     GO APPLY FN TO RESULTS
EV350 LD   I  @ARG1     GET (CDR ARG)
```

```
      STO   3 @ARG2-X   SAVE AS ARG 2 FOR APPLY
      LD    L EV120     GET (CAR ARG)
      STO   3 @ARG1-X   SAVE AS ARG 1 FOR APPLY
      BSC   L APPLY     APPLY FN TO UNEVALED FORMS
EV400 LD    I @ARG1     GET (CDR ARG)
      STO   3 @ARG2-X   SAVE AS ARG 2 FOR APPLY
      LD    L EV120     GET (CAR ARG)
      STO   3 @ARG1-X   SAVE AS ARG 1 FOR APPLY
      BSI   3 PUSHJ-X   APPLY FN TO UNEVALED FORMS
      DC      APPLY
      STO   3 @ARG1-X
      BSC   L EVAL      NOW GO EVAL RESULT
**************************************************
EV950 DC      1
**************************************************
EV500 LD    I EV120     GET (CDAR ARG)
      BSC   L EV600,+-  ERROR IF NIL
      A       EV950
      STO     EV510+1
EV510 LD    L *-*       GET (CADAR ARG)
      STO     EV530     SAVE IT
      S       EV960
      BSC   L EV520,+Z  BRANCH IF NUMBER OR NIL
      S       EV961
      BSC   L EV520,-   BRANCH IF NUMBER
      LD      EV530
      A       EV950
      STO     EV515+1
EV515 LD    L *-*
      BSC   L EV550,+Z  BRANCH IF ATOM
EV520 BSI   3 ERROR-X   ERROR - BAD FIRST ARG
      DC      26+@MAJR
EV530 DC      *-*
EV550 LD    I EV530     TEST ATOM
      S       EV962
      BSC   L EV520,+-  ERROR IF ACTUALLY A STRING
      LD      EV530
      STO     EV560
      BSI   3 PUSHS-X   SAVE ATOM'S OLD VALUE
EV560 DC      *-*        ON SPEC PDL
      LD    I EV120     GET (CDAR ARG)
      STO     EV570+1
EV570 LD    L *-*       GET (CDDAR ARG)
      BSC   L EV600,+-  ERROR IF NIL
      STO     EV580+1
      A       EV950
      STO     EV590+1
EV580 LD    L *-*       GET (CDDDAR ARG)
      BSC   L EV600,Z   ERROR IF NOT NIL
EV590 LD    L *-*       GET (CADDAR ARG)
      STO   I EV560     SAVE AS NEW VALUE OF ATOM
      RTE     16
      LD    I @ARG1     GET (CDR ARG)
      BSI   3 XCONS-X   CONS THEM
      STO   3 @ARG1-X   SAVE AS ARG
      BSI   3 PUSHJ-X   EVAL IT
      DC      EVAL
      STO     EV530     SAVE VALUE TEMPORARILY
      BSI   3 POPS-X    RESTORE OLD VALUE OF ATOM
      LD      EV530     RETURN VALUE SAVED
      BSI   3 POPJ-X
EV600 LD    L EV120     ERROR - WRONG NUMBER
      STO     EV610      OF ARGS FOR LABEL
      BSI   3 ERROR-X
      DC      27+@MAJR
EV610 DC      *-*
**************************************************
EV960 DC      S@FST
EV961 DC      E@FST-S@FST
EV962 DC      @STR
**************************************************
```

```
      HDNG    APPLY FUNCTION
*************************************************
*     APPLY FUNCTION                            *
*************************************************
      DC       @LAM+2    (LAMBDA (FN ARGS) ...
APPLY LD     3 @ARG1-X   IS FN AN ATOM
      BSC      +-
      BSI    3 POPJ-X    RETURN NIL IF NIL
      BSI    3 XATOM-X
      BSC   L  AP050,+-
AP020 LD     3 @ARG2-X   IF SO, SAVE ARGS
      BSI    3 PUSHA-X
      BSI    3 PUSHJ-X   EVAL FN
      DC       EVAL
      STO    3 @ARG1-X   SAVE AS NEW FN
      BSI    3 POPA-X    GET ARGS
      STO    3 @ARG2-X
      MDX      APPLY     NOW TRY AGAIN TO APPLY
AP050 LD     3 @ARG1-X   GET (CAR FN)
      A        AP901
      STO      AP060+1
AP060 LD    L  *-*
      S        AP902     IS IT LAMBDA
      BSC      Z
      S        AP903     IS IT NLAMBDA
      BSC      Z
      S        AP904     IS IT MLAMBDA
      BSC   L  AP400,Z   IF NOT, BRANCH
      STO      AP910     CLEAR COUNT OF ARG BINDINGS
      LD    I  @ARG1     GET (CDR FN)
      A        AP901
      STO      AP110+1
AP110 LD    L  *-*       GET (CADR FN)
      STO      AP130     SAVE ADR OF PARAMS
AP120 STO      AP190     SAVE REST OF PARAMS
      BSC   L  AP140,Z   BRANCH UNLESS NONE LEFT
      LD     3 @ARG2-X   GET ARGS
      BSC   L  AP300,+-  BRANCH IF NONE LEFT
      BSI    3 ERROR-X   ERROR - TOO MANY ARGS
      DC       28+@MAJR
AP130 DC       *-*
AP140 LD       AP190     CHECK PARAM LIST
      S        AP912
      BSC   L  AP150,+Z  BRANCH IF NUMBER OR NIL
      S        AP913
      BSC   L  AP170,+Z  BRANCH UNLESS NUMBER
AP150 LD       AP130
      STO      AP160
      BSI    3 ERROR-X   ERROR - BAD PARAM LIST
      DC       30+@MAJR
AP160 DC       *-*
AP170 LD       AP190     TEST IF PARAM LIST
      A        AP901      IS AN ATOM
      STO      AP180+1
AP180 LD    L  *-*
      BSC   L  AP200,-   BRANCH IF NOT
      LD    I  AP190     TEST IF IT IS A STRING
      S        AP914
      BSC   L  AP150,+-  BRANCH IF SO - ERROR
      BSI    3 PUSHS-X   PUSH OLD VALUE ON SPEC PDL
AP190 DC       *-*
      LD     3 @ARG2-X   BIND ATOM TO LIST OF
      STO   I  AP190      REST OF ARGS
      MDX   L  AP910,-1  INCR NEG COUNT OF BINDINGS
      NOP
      MDX      AP300
AP200 LD     3 @ARG2-X   TEST ARGS
      BSC   L  AP220,Z   BRANCH UNLESS NONE LEFT
      LD       AP130
      STO      AP210
      BSI    3 ERROR-X   ERROR - TOO FEW ARGS
      DC       29+@MAJR
```

```
AP210 DC        *-*
AP220 LD        AP190      GET CAR OF PARAM LIST
      A         AP901
      STO       AP230+1
AP230 LD    L   *-*
      STO       AP250
      S         AP912
      BSC   L   AP150,+Z   BRANCH IF NUMBER OR NIL
      S         AP913
      BSC   L   AP160,-    BRANCH IF NUMBER
      LD        AP250
      A         AP901
      STO       AP240+1
AP240 LD    L   *-*
      BSC   L   AP150,-    BRANCH UNLESS ATOM
      LD    I   AP250
      S         AP914
      BSC   L   AP150,+-   BRANCH IF STRING
      BSI    3  PUSHS-X    PUSH OLD VALUE ON SPEC PDL
AP250 DC        *-*
      LD     3  @ARG2-X    GET CAR OF ARG LIST
      A         AP901
      STO       AP260+1
AP260 LD    L   *-*
      STO   I   AP250      BIND ATOM TO NEW VALUE
      MDX   L   AP910,-1   INCR NEG COUNT OF BINDINGS
      NOP
      LD    I   @ARG2      GET REST OF ARGS
      STO    3  @ARG2-X     AND SAVE
      LD    I   AP190      GET REST OF PARAMS
      MDX       AP120       AND GO BIND THEM TOO
*************************************************
AP901 DC        1
AP902 DC        #LAM
AP903 DC        #NLAM-#LAM
AP904 DC        #MLAM-#NLAM
AP910 DC        *-*
AP912 DC        S@FST
AP913 DC        E@FST-S@FST
AP914 DC        @STR
*************************************************
AP300 LD        AP910      PUSH NEG COUNT OF BINDINGS
      BSI    3  PUSHA-X
      SRA       16
      BSI    3  PUSHA-X    PUSH A NIL FOR RESULT
      BSI    3  PUSHA-X    PUSH A NIL FOR FORMS
      LD     3  @ARG1-X
      STO       AP310+1
AP310 LD    I   *-*        GET (CDDR FN)
AP320 BSC   L   AP340,+-   DONE IF NO FORMS LEFT
      STO    1  0
      A         AP901
      STO       AP330+1
AP330 LD    L   *-*        GET NEXT FORM ON LIST
      STO    3  @ARG1-X
      BSI    3  PUSHJ-X    EVAL IT
      DC        EVAL
      STO    1  1          SAVE RESULT
      LD    I1  0          GET REST OF FORMS
      MDX       AP320      GO EVAL THEM TOO
AP340 BSI    3  POPA-X     POP OFF LIST OF FORMS
      BSI    3  POPA-X     POP OFF FINAL RESULT
      BSI    3  POPN-X     UNBIND VARIABLES
      BSI    3  POPJ-X     RETURN
*************************************************
AP400 S         AP930      TEST (CAR FN)
      BSC   L   AP700,Z    BRANCH UNLESS SUBR
      LD    I   @ARG1      GET ADDRESS OF SUBR
      STO       AP933
      S         AP931      MUST BE BETWEEN END OF
      BSC   L   AP410,+Z    RESIDENT MONITOR AND
      S         AP932       START OF DATA AREA
      BSC  L  AP430,+Z
AP410 LD   I  @ARG1
      STO     AP420
      LD    3 @ARG1-X
      STO     AP425
      BSI   3 ERROR-X   ERROR IF AT INVALID ADR
      DC      34+@MAJR
AP420 DC      *-*
AP425 DC      *-*
AP430 LD   I  AP933     GET ARGUMENT INDICATOR
      AND     AP934     AND OUT TOP 8 BITS
      STO     AP935     SAVE NUMBER OF ARGS WANTED
      S       AP936
      BSC  L  AP500,+Z  BRANCH IF LESS THAN 8
      BSC  L  AP440,Z   ERROR IF MORE THAN 8
      LD   I  AP933
      SLA     2
      BSC  L  AP500,-   ERROR IF 8 PLUS A LIST
AP440 LD    3 @ARG1-X
      STO     AP445
      LD      AP933
      STO     AP450
      BSI   3 ERROR-X
      DC      33+@MAJR
AP445 DC      *-*
AP450 DC      *-*
***************************************************
AP930 DC      #SUBR-#MLAM
AP931 DC      $ZEND
AP932 DC      S@FXB-$ZEND
AP933 DC      *-*
AP934 DC      /00FF
AP935 DC      *-*
AP936 DC      8         NUMBER OF @ARGN LOCATIONS
***************************************************
AP500 LD      AP940     SET UP TO SPREAD ARGS
      STO     AP941
      LD    3 @ARG1-X   SAVE ARG 1 FOR ERROR MSGS
      STO     AP530
      STO     AP560
      LD    3 @ARG2-X
AP510 STO     AP942     SAVE REST OF ARGS
      MDX  L  AP935,0   SKIP IF NO MORE WANTED
      MDX     AP550
      LD   I  AP933
      SLA     2
      BSC  L  AP520,-   BRANCH UNLESS LIST WANTED
      LD      AP942
      STO  I  AP941     PUT INTO NEXT ARG
      MDX     AP600
AP520 LD      AP942     NO MORE ARGS WANTED AT ALL
      BSC  L  AP600,+-  BRANCH IF NONE LEFT
      LD      AP933
      STO     AP540
      BSI   3 ERROR-X   ERROR - TOO MANY ARGS
      DC      31+@MAJR
AP530 DC      *-*
AP540 DC      *-*
AP550 LD      AP942     MORE ARGS WANTED
      BSC  L  AP580,Z   BRANCH UNLESS NONE LEFT
      LD      AP933
      STO     AP570
      BSI   3 ERROR-X   ERROR - TOO FEW ARGS
      DC      32+@MAJR
AP560 DC      *-*
AP570 DC      *-*
AP580 LD      AP942     GET (CAR ARGS)
      A       AP943
      STO     AP590+1
AP590 LD   L  *-*
      STO  I  AP941     PUT INTO NEXT ARG
      MDX  L  AP941,1   INCR ARG POINTER
```