```
      MDX  L  AP935,-1  DECR ARG COUNT
      NOP
      LD   I  AP942     GET REST OF ARGS AND
      MDX     AP510      GO SPREAD THEM TOO
AP600 MDX  L  AP933,1   INCR TO GET ENTRY POINT
      BSC  I  AP933     APPLY'S VALUE IS SUBR'S
*************************************************
AP940 DC      @ARG1
AP941 DC      *-*
AP942 DC      *-*
AP943 DC      1
*************************************************
AP700 S       AP950     TEST (CAR FN)
      BSC  L  AP820,Z   BRANCH UNLESS C-R
      LD    3 @ARG2-X   TEST LIST OF ARGS
      BSC  L  AP710,+-  BRANCH IF NONE
      LD   I  @ARG2
      BSC  L  AP740,+-  BRANCH UNLESS TWO OR MORE
AP710 LD    3 @ARG1-X
      STO     AP720
      LD    3 @ARG2-X
      STO     AP730
      BSI   3 ERROR-X   WRONG NUMBER OF ARGS
      DC      35+@MAJR
AP720 DC      *-*
AP730 DC      *-*
AP740 SRA     16
      STO     AP941     CLEAR COUNTER
      LD    3 @ARG2-X
      A       AP943
      STO     AP745+1
AP745 LD   L  *-*       GET ARG
      STO     AP942     SAVE ARG
      LD   I  @ARG1     GET (CDR FN) C-R TYPE ATOM
      A       AP943
      STO     AP750+1
AP750 LD   L  *-*       GET PRINT NAME
      STO     AP760+1   SKIP FIRST CHAR
AP760 LD   L  *-*
      STO     AP760+1   SAVE LIST OF CHARS
      A       AP943
      STO     AP770+1
      LD   I  AP760+1
      BSC  L  AP780,+-  BRANCH IF LAST CHAR
AP770 LD   L  *-*       GET CHAR
      BSI   3 PUSHA-X   PUSH ON STACK
      MDX  L  AP941,1   INCR COUNT
      MDX     AP760
AP780 LD      AP941
      BSC  L  AP810,+-  BRANCH IF NO A'S OR D'S
AP785 BSI   3 POPA-X    POP OFF AN A OR A D
      EOR     AP951
      BSC  L  AP790,Z   BRANCH UNLESS A
      LD      AP942
      BSI   3 XCAR-X    TAKE CAR
      MDX     AP800
AP790 LD      AP942
      BSI   3 XCDR-X    ELSE TAKE CDR
AP800 STO     AP942
      MDX  L  AP941,-1  COUNT A'S AND D'S
      MDX     AP785
AP810 LD      AP942     RETURN RESULT
      BSI   3 POPJ-X
*************************************************
AP950 DC      #C@R-#SUBR
AP951 DC      @A
*************************************************
AP997 DC      #LABL
AP998 DC      #NLAM
AP999 DC      #MLAM
*************************************************
```

```
      HDNG    COND, SET/SETQ/SETQQ FUNCTIONS
***************************************************
*     COND FUNCTION                               *
***************************************************
      DC       @NLAM+@LIST    (LAMBDA X ...
COND  LD    3 @ARG1-X   GET LIST OF LISTS
      BSI   3 PUSHA-X   SPACE FOR LISTS AND RESULT
      BSI   3 PUSHA-X   SPACE FOR CURRENT LIST
COND2 BSC  L  COND6,+-  NO MORE LISTS - RESULT NIL
      BSI   3 XCAR-X    GET NEXT LIST
      STO   1 0         SAVE
      BSI   3 XCAR-X    GET FIRST ITEM
      STO   3 @ARG1-X
      BSI   3 PUSHJ-X   EVAL IT
      DC      EVAL
      BSC  L  COND4,Z   BRANCH UNLESS NIL
      LD   I1 1         GET REST OF LISTS
      STO   1 1
      MDX     COND2     GO TRY NEXT ONE
COND4 STO   1 1         SAVE VALUE OF ITEM
      LD   I1 0         GET REST OF ITEMS
      BSC  L  COND6,+-  BRANCH IF NONE LEFT
      STO   1 0         SAVE
      BSI   3 XCAR-X    GET NEXT ITEM
      STO   3 @ARG1-X
      BSI   3 PUSHJ-X   EVAL IT
      DC      EVAL
      MDX     COND4     GO TRY REST OF ITEMS
COND6 BSI   3 POPA-X    POP CURRENT LIST
      BSI   3 POPA-X    POP RESULT
      BSI   3 POPJ-X    RETURN
***************************************************
*     SET/SETQ/SETQQ FUNCTION                     *
***************************************************
      DC      @LAM+2    (LAMBDA (X Y) ...
SET   MDX     SET10
***************************************************
      DC      @NLAM+2   (NLAMBDA (X Y) ...
SETQQ MDX     SET10
***************************************************
      DC      @NLAM+2   (NLAMBDA (X Y) ...
SETQ  LD    3 @ARG1-X
      BSI   3 PUSHA-X   SAVE FIRST ARG
      LD    3 @ARG2-X
      STO   3 @ARG1-X
      BSI   3 PUSHJ-X   EVAL SECOND ARG
      DC      EVAL
      STO   3 @ARG2-X   SAVE RESULT
      BSI   3 POPA-X
      STO   3 @ARG1-X   RESTORE FIRST ARG
SET10 LD    3 @ARG1-X   CHECK FIRST ARG
      S       SET90
      BSC  L  SET30,+Z  ERROR IF NUMBER OR NIL
      S       SET91
      BSC  L  SET30,-   ERROR IF NUMBER
      A       SET92
      STO     SET20+1
SET20 LD   L  *-*
      BSC  L  SET30,-   ERROR IF NOT ATOM
      LD   I  @ARG1
      S       SET93
      BSC  L  SET50,Z   ERROR IF STRING
SET30 LD    3 @ARG1-X
      STO     SET40
      BSI   3 ERROR-X   BAD FIRST ARG FOR SET
      DC      36+@MAJR
SET40 DC      *-*
SET50 LD    3 @ARG2-X   SET ATOM TO VALUE
      STO  I  @ARG1      OF SECOND ARG AND
      BSI   3 POPJ-X     RETURN THAT VALUE
***************************************************
SET90 DC      S@FST
```

```
SET91 DC      E@FST-S@FST
SET92 DC      1+E@FST
SET93 DC      @STR
**************************************************
*     LIST FUNCTION                              *
**************************************************
      DC      @LAM+@LIST
LIST  LD    3 @ARG1-X   GET LIST OF (EVALED) ARGS
      BSI   3 POPJ-X     AND RETURN
**************************************************
```

```
      HDNG    1442 CARD READER INPUT HANDLER
**************************************************
*     1442 INPUT HANDLER                         *
**************************************************
      AIF     (@READ EQ YES),.YES
I1442 EQU     0
F1442 EQU     0
      AGO     .NO
.YES  ANOP
I1442 DC      *-*
      MDX  L  I1499,0   SKIP UNLESS FLUSH REQUESTED
      MDX     I1450     GO FLUSH
      MDX  L  I1498,0   SKIP IF NO CHARS LEFT
      MDX     I1430
I1410 BSI     I1460     READ CARD, SKIP IF EOF
      MDX     I1420
      MDX  L  REDSW,0   SKIP IF IN MIDDLE OF READ
      MDX     I1410
      BSC  L  RDEOF     GO HANDLE READ EOF ERROR
I1420 LD      I14BF     SET CHAR COUNT
      STO     I1498
      LD      I1496     SET CHAR POINTER
      STO     I1495
I1430 LD   I  I1495     GET CHAR
      MDX  L  I1495,1   INCR POINTER
      MDX  L  I1498,-1  DECR COUNT
      NOP
      STO     I1494     SAVE CHAR
      LDX   2 -L@EBC    SEARCH TABLE
I1435 LD   L2 CRDTB+L@EBC
      EOR     I1494
      BSC  L  I1440,+-
      MDX   2 1
      MDX     I1435
      LDX   2 -L@EBC    USE BLANK IF NOT FOUND
I1440 LD   L  2         CALCULATE ADR
      A       I1493
      BSC  I  I1442     RETURN
I1450 SRA     16
      STO     I1499     CLEAR FLUSH SWITCH
I1455 BSI     I1460     READ CARD, SKIP IF EOF
      MDX     I1455     IF NOT, TRY AGAIN
      MDX     I1410     IF SO, TRY TO READ A CHAR
I1460 DC      *-*
      LIBF    CARD0     READ A CARD
      DC      /1000
      DC      I14BF
I1465 LIBF    CARD0     WAIT FOR IT
      DC      0
      MDX     I1465
      LDX   2 5         COMPARE TO /*/*/
I1470 LD   L2 I14BF
      EOR  L2 I1492-1
      BSC  I  I1460,Z   RETURN IF NON-MATCH
      MDX   2 -1
      MDX     I1470
      MDX  L  I1460,1   SKIP ON RETURN IF EOF
      BSC  I  I1460
**************************************************
I1492 DC      /3000     CARD CODE /
      DC      /4220     CARD CODE *
      DC      /3000     CARD CODE /
      DC      /4220     CARD CODE *
      DC      /3000     CARD CODE /
I1493 DC      EBCTB+L@EBC
I1494 DC      *-*
I1495 DC      I14BF+73
I1496 DC      I14BF+1
I1498 DC      0
I1499 DC      0         NON-ZERO = FLUSH REQUEST
**************************************************
F1442 DC      *-*
```

```
      STX      I1499     SET FLUSH SWITCH
      BSC  I  F1442
**************************************************
I14BF DC       72        1442 CARD INPUT BUFFER
      BSS      72
**************************************************
.NO   ANOP
```

```
      HDNG     1132 PRINTER OUTPUT HANDLER
***************************************************
*     1132 PRINTER OUTPUT HANDLER                 *
***************************************************
      AIF      (@1132 EQ YES),.YES
O1132 EQU      0
P1132 EQU      0
      AGO      .NO
.YES  ANOP
O1132 DC       *-*
      LD   L   2         CHECK CHAR
      S        O1199
      BSC  L   O1135,+-  BRANCH IF CARRIAGE RETURN
      LD       O1198
      EOR      O1197     FLIP BIT 0 OF POINTER
      BSC      -
      A        O1196     IF BIT 0 IS NOW 0, INCR
      STO      O1198     SAVE POINTER
      SLA      1         PUT BIT 0 IN CARRY
      LD   L2 EBCTB      GET CHAR FROM EBCDIC TABLE
      BSC  L   O1120,C   BRANCH IF BIT 0 WAS 1
      AND      O1195     AND FLAG BITS OUT OF WORD
      MDX      O1125
O1120 SRA      8         SHIFT TO RIGHT-HAND HALF
      OR   I   O1198     OR IN LAST CHAR
O1125 STO  I   O1198     SAVE IN BUFFER
      MDX  L   O11BF,1   INCR COUNT (NO. OF CHARS+1)
O1127 BSC  I   O1132
O1135 LD       O11BF     CHECK COUNT
      SRA      1         DIVIDE BY 2
      BSC  L   O1150,+   BRANCH IF NO CHARS IN BUF
      STO      O11BF     SAVE NUMBER OF WORDS
      LIBF     PRNT1     PRINT LINE
      DC       /2000
      DC       O11BF
      DC       O1170
O1140 LIBF     PRNT1     WAIT UNTIL DONE
      DC       0
      MDX      O1140
      LD       O1196     RESET COUNT
      STO      O11BF
      LD       O1194     RESET POINTER
      STO      O1198
      MDX      O1127
O1150 LIBF     PRNT1     PRINT BLANK LINE (SKIP)
      DC       /3D00
      MDX      O1127
***************************************************
O1170 DC       *-*       PRNT1 ERROR HANDLER
      BSC  I   O1170     DON'T DO ANYTHING SPECIAL
***************************************************
O1194 DC       O11BF+/8000
O1195 DC       /FF00
O1196 DC       1
O1197 DC       /8000
O1198 DC       O11BF+/8000
O1199 DC       @CR-EBCTB
***************************************************
O11BF DC       1         1132 OUTPUT BUFFER
      BSS      60
***************************************************
P1132 DC       *-*       1132 PRINTER PAGESKIP
      LIBF     PRNT1     SKIP TO CHANNEL 1
      DC       /3100
      BSC  I   P1132
***************************************************
.NO   ANOP
***************************************************
```

```
      HDNG    ARITHMETIC FUNCTIONS
***************************************************
*     NUMERIC FUNCTIONS                           *
***************************************************
*     THIS ROUTINE HANDLES ALL ARITHMETIC         *
*     FUNCTIONS WHICH TAKE ANY NUMBER OF ARGS.    *
*     THESE INCLUDE BOOLE, PLUS, DIFF, TIMES,     *
*     QUOTIENT, REMAINDER, MAX, MIN, GCD.         *
***************************************************
      DC        @LAM+2+@LIST
BOOLE LD    3 @ARG1-X   GET FIRST ARG
      BSI   3 XNCHK-X   CHECK IT
      DC        #BOOL
      AND       BOOL9     GET LOW 4 BITS
      STX   2 NF035+1   SAVE XR2
      STO  L  2           PUT FN INDICATOR IN XR2
      LD    3 @ARG2-X
      STO   3 @ARG1-X
      LD    3 @ARG3-X
      STO   3 @ARG2-X
      BSI       NFNCS
      DC        #BOOL     BOOLE
***************************************************
BOOL9 DC        /000F
***************************************************
      DC        @LAM+1+@LIST
PLUS  STX   2 NF035+1   SAVE XR2
      LDX   2 16        SET FN INDICATOR
      BSI       NFNCS
      DC        #PLUS
***************************************************
      DC        @LAM+1+@LIST
DIFF  STX   2 NF035+1   SAVE XR2
      LDX   2 17        SET FN INDICATOR
      BSI       NFNCS
      DC        #DIFF     DIFF
***************************************************
      DC        @LAM+1+@LIST
TIMES STX   2 NF035+1   SAVE XR2
      LDX   2 18        SET FN INDICATOR
      BSI       NFNCS
      DC        #TIMS     TIMES
***************************************************
      DC        @LAM+1+@LIST
QUO   STX   2 NF035+1   SAVE XR2
      LDX   2 19        SET FN INDICATOR
      BSI       NFNCS
      DC        #QUO      QUOTIENT
***************************************************
      DC        @LAM+1+@LIST
REM   STX   2 NF035+1   SAVE XR2
      LDX   2 20        SET FN INDICATOR
      BSI       NFNCS
      DC        #REM      REMAINDER
***************************************************
      DC        @LAM+1+@LIST
MAX   STX   2 NF035+1   SAVE XR2
      LDX   2 21        SET FN INDICATOR
      BSI       NFNCS
      DC        #MAX      MAX
***************************************************
      DC        @LAM+1+@LIST
MIN   STX   2 NF035+1   SAVE XR2
      LDX   2 22        SET FN INDICATOR
      BSI       NFNCS
      DC        #MIN      MIN
***************************************************
      DC        @LAM+1+@LIST
GCD   STX   2 NF035+1   SAVE XR2
      LDX   2 23        SET FN INDICATOR
      BSI       NFNCS
      DC        #GCD      GCD
```

```
*************************************************
NFNCS DC        *-*
      LD    I  NFNCS     GET FUNCTION NAME
      STO       NF020
      STO       NF050
      LD     3 @ARG1-X   GET FIRST ARG
      BSI    3 XNCHK-X   CHECK IT
NF020 DC        *-*
      STO       NF911    SAVE IT
      LD     3 @ARG2-X
NF030 BSC   L  NF040,Z   BRANCH UNLESS NONE LEFT
      LD        NF911    GET RESULT
      BSI    3 MKFXN-X   MAKE IT A NUMBER
NF035 LDX   L2 *-*       RESTORE XR2
      BSI    3 POPJ-X    RETURN
NF040 BSI    3 XCAR-X    GET NEXT ARG
      BSI    3 XNCHK-X   CHECK IT
NF050 DC        *-*
      STO       NF912    SAVE IT (B)
      LD        NF911    GET LAST PARTIAL RESULT (A)
      BSC   I2 NF055     BRANCH TO DO FUNCTION
*************************************************
NF055 DC        NFA10    0
      DC        NFB10    A AND B
      DC        NFC10    (NOT A) AND B
      DC        NFD10    B
      DC        NFE10    A AND (NOT B)
      DC        NFF10    A
      DC        NFG10    A EOR B
      DC        NFH10    A OR B
      DC        NFI10    (NOT A) AND (NOT B)
      DC        NFJ10    A EQV B
      DC        NFK10    NOT A
      DC        NFL10    (NOT A) OR B
      DC        NFM10    NOT B
      DC        NFN10    A OR (NOT B)
      DC        NFO10    (NOT A) OR (NOT B)
      DC        NFP10    1
      DC        NFQ10    A+B
      DC        NFR10    A-B
      DC        NFS10    A*B
      DC        NFT10    A/B
      DC        NFU10    A REMAINDER B
      DC        NFV10    A MAX B
      DC        NFW10    A MIN B
      DC        NFX10    A GCD B
*************************************************
NF060 STO       NF911    SAVE NEW PARTIAL RESULT
      LD    I  @ARG2     CHAIN DOWN LIST OF ARGS
      STO    3 @ARG2-X
      MDX       NF030
*************************************************
NF911 DC        *-*      PARTIAL RESULT
NF912 DC        *-*      NEXT ARG
NF913 DC        /FFFF
*************************************************
NFA10 SRA       16       0
      MDX       NF060
*************************************************
NFB10 AND       NF912    A AND B
      MDX       NF060
*************************************************
NFC10 EOR       NF913    (NOT A) AND B
      MDX       NFB10
*************************************************
NFD10 LD        NF912    B
      MDX       NF060
*************************************************
NFE10 LD        NF912    A AND (NOT B)
      EOR       NF913
      AND       NF911
      MDX       NF060
```

```
*****************************************************
NFF10 EQU     NF060     A
*****************************************************
NFG10 EOR     NF912     A EOR B
      MDX     NF060
*****************************************************
NFH10 OR      NF912     A OR B
      MDX     NF060
*****************************************************
NFI10 OR      NF912     NOT (A OR B)
      MDX     NFK10
*****************************************************
NFJ10 EOR     NF913     (NOT A) EOR B
      MDX     NFG10
*****************************************************
NFK10 EOR     NF913     NOT A
      MDX     NF060
*****************************************************
NFL10 EOR     NF913     (NOT A) OR B
      MDX     NFH10
*****************************************************
NFM10 LD      NF912     NOT B
      MDX     NFK10
*****************************************************
NFN10 EOR     NF913     NOT ((NOT A) AND B)
NFO10 AND     NF912     NOT (A AND B)
      MDX     NFK10
*****************************************************
NFP10 LD      NF913     1
*****************************************************
NFQ10 A       NF912     A+B
      MDX     NF060
*****************************************************
NFR10 S       NF912     A-B
      MDX     NF060
*****************************************************
NFS10 M       NF912     A*B
      SLT     16
      MDX     NF060
*****************************************************
NFT10 SRT     16        A/B
      D       NF912
      MDX     NF060
*****************************************************
NFU10 SRT     16        A REMAINDER B
      D       NF912
      RTE     16
      MDX     NF060
*****************************************************
NFV10 LDS     0         A MAX B
      S       NF912
      BSC     0
      EOR     NF913
      BSC  L  NFD10,+Z
NFV15 LD      NF911
      MDX     NF060
*****************************************************
NFW10 LDS     0         A MIN B
      S       NF912
      BSC     0
      EOR     NF913
      BSC  L  NFD10,-
      MDX     NFV15
*****************************************************
NFX10 SRT     16
      D       NF912     DIVIDE ACC BY EXT
      RTE     16
      BSC  L  NFX30,+-  BRANCH IF ZERO REMAINDER
      RTE     16        SAVE REMAINDER
      LD      NF912     GET B
      RTE     16
      STO     NF912     MAKE LAST REMAINDER B
      RTE      16        PUT OLD B IN ACC
      MDX      NFX10
NFX30 LD       NF912     RETURN B
      MDX      NF060
*****************************************************
*     MINUS FUNCTION                                *
*****************************************************
      DC       @LAM+1
MINUS LD     3 @ARG1-X   GET ARG
      BSI    3 XNCHK-X   CHECK IT
      DC       #MNUS
MNUS5 SRA      16        (ABS BRANCHES HERE)
      S    I   @ARG1     GET NEGATIVE OF ARG
      BSI    3 MKFXN-X
      BSI    3 POPJ-X
*****************************************************
*     ABS FUNCTION                                  *
*****************************************************
      DC       @LAM+1
ABS   LD     3 @ARG1-X   GET ARG
      BSI    3 XNCHK-X   CHECK IT
      DC       #ABS
      BSC  L   MNUS5,+Z  IF NEGATIVE, GO NEGATE
      LD     3 @ARG1-X   ELSE RETURN ARG
      BSI    3 POPJ-X
*****************************************************
*     ZEROP FUNCTION                                *
*****************************************************
      DC       @LAM+1
ZEROP LD     3 @ARG1-X   GET ARG
      BSI    3 XNCHK-X   CHECK IT
      DC       #ZERP
      BSC  L   ZERP5,Z
      LD     3 @TRUE-X   RETURN T IF ZERO
      BSI    3 POPJ-X
ZERP5 SRA      16        ELSE NIL (MINUSP USES THIS)
      BSI    3 POPJ-X
*****************************************************
*     MINUSP FUNCTION                               *
*****************************************************
      DC       @LAM+1
MNUSP LD     3 @ARG1-X   GET ARG
      BSI    3 XNCHK-X   CHECK IT
      DC       #MNSP
      BSC  L   ZERP5,-
      LD     3 @TRUE-X   RETURN T IF NEGATIVE
      BSI    3 POPJ-X
*****************************************************
*     ADD1 FUNCTION                                 *
*****************************************************
      DC       @LAM+1
ADD1  LD     3 @ARG1-X   GET ARG
      BSI    3 XNCHK-X   CHECK IT
      DC       #ADD1
      S        ADD19     ADD ONE
      BSI    3 MKFXN-X
      BSI    3 POPJ-X
*****************************************************
ADD19 DC       /FFFF     -1
*****************************************************
*     SUB1 FUNCTION                                 *
*****************************************************
      DC       @LAM+1
SUB1  LD     3 @ARG1-X   GET ARG
      BSI    3 XNCHK-X   CHECK IT
      DC       #SUB1
      A        ADD19     SUBTRACT ONE
      BSI    3 MKFXN-X
      BSI    3 POPJ-X
*****************************************************
*     LSH FUNCTION                                  *
*****************************************************
```

```
      DC       @LAM+2
LSH   LD     3 @ARG1-X   CHECK FIRST ARG
      BSI    3 XNCHK-X
      DC       #LSH
      LD     3 @ARG2-X   CHECK SECOND ARG
      BSI    3 XNCHK-X
      DC       #LSH
      BSC  L   LSH2,+Z   BRANCH IF NEGATIVE
      AND      LSH9      SET UP LEFT SHIFT
      OR       LSH8
      MDX      LSH4
LSH2  EOR      ADD19     NEGATE SECOND ARG
      S        ADD19
      AND      LSH9      SET UP RIGHT SHIFT
      OR       LSH7
LSH4  STO      LSH5
      LD   I   @ARG1
LSH5  SLA      *-*       SLA OR SRA GETS PUT HERE
      BSI    3 MKFXN-X
      BSI    3 POPJ-X
**************************************************
LSH7  SRA      0
LSH8  SLA      0
LSH9  DC       /003F
**************************************************
*     LESSP FUNCTION                             *
**************************************************
      DC       @LAM+1+@LIST
LESSP LD     3 @ARG1-X
      BSI    3 XNCHK-X   CHECK FIRST ARG
      DC       #LESP
      STO      LES99     SAVE IT
      LD     3 @ARG2-X
LES10 BSC  L   LES20,Z   BRANCH UNLESS NO ARGS LEFT
      LD     3 @TRUE-X   RETURN TRUE - ALL TESTS OK
      BSI    3 POPJ-X
LES20 BSI    3 XCAR-X    GET NEXT ARG
      BSI    3 XNCHK-X   CHECK IT
      DC       #LESP
      STO      LES98     SAVE IT
      LD       LES99     COMPARE TO LAST
      LDS      0
      S        LES98
      BSC      0
      EOR      *-1
      BSC  L   LES60,-   BRANCH IF LAST GE THIS
LES30 LD       LES98     SAVE THIS ARG TO
      STO      LES99      COMPARE TO NEXT
LES50 LD   I   @ARG2     CHAIN DOWN ARG LIST
      STO    3 @ARG2-X
      MDX      LES10
LES60 SRA      16        RETURN NIL IF ANY
      BSI    3 POPJ-X     RELATION UNSATISFIED
**************************************************
LES97 DC       *-*
LES98 DC       *-*
LES99 DC       *-*
**************************************************
*     OR FUNCTION                                *
**************************************************
      DC       @NLAM+@LIST
OR    LD     3 @ARG1-X   GET LIST OF ARGS
      BSI    3 PUSHA-X   SAVE ON STACK
OR2   BSC  L   OR6,+-    BRANCH IF NONE LEFT
      BSI    3 XCAR-X    GET NEXT ARG
      STO    3 @ARG1-X
      BSI    3 PUSHJ-X   EVAL IT
      DC       EVAL
      BSC  L   OR4,Z     BRANCH UNLESS NIL
      LD    I1 0
      STO    1 0
      MDX      OR2
```

```
OR4   LD    3 @TRUE-X   RETURN T
OR6   RTE     16
      BSI   3 POPA-X    POP OFF LIST OF ARGS
      RTE     16
      BSI   3 POPJ-X
***************************************************
*     AND FUNCTION                                *
***************************************************
      DC      @NLAM+@LIST
AND   LD    3 @ARG1-X   GET LIST OF ARGS
      BSI   3 PUSHA-X   SAVE ON STACK
AND2  BSC  L  OR4,+-    BRANCH IF NONE LEFT
      BSI   3 XCAR-X    GET NEXT ARG
      STO   3 @ARG1-X
      BSI   3 PUSHJ-X   EVAL IT
      DC      EVAL
      BSC  L  OR6,+-    BRANCH UNLESS NON-NIL
      LD   I1 0
      STO   1 0
      MDX     AND2
***************************************************
*     EXAM FUNCTION                               *
***************************************************
      DC      @LAM+1
EXAM  LD    3 @ARG1-X   GET ARG
      BSI   3 XNCHK-X   CHECK IT
      DC      #EXAM
      STO     EXAM3+1
EXAM3 LD   L  *-*       GET WORD AT GIVEN ADR
      BSI   3 MKFXN-X
      BSI   3 POPJ-X
***************************************************
*     DEP FUNCTION                                *
***************************************************
      DC      @LAM+2
DEP   LD    3 @ARG1-X   GET ARG 1
      BSI   3 XNCHK-X   CHECK IT
      DC      #DEP
      STO     DEP3+1    SAVE ADR
      LD    3 @ARG2-X   GET ARG 2
      BSI   3 XNCHK-X   CHECK IT
      DC      #DEP
DEP3  STO  L  *-*       PUT WORD AT GIVEN ADR
      LD    3 @ARG2-X   RETURN ARG 2
      BSI   3 POPJ-X
***************************************************
*     SWITCH FUNCTION                             *
***************************************************
      DC      @LAM+1
SWTCH LD    3 @ARG1-X   GET ARG
      BSI   3 XNCHK-X   CHECK IT
      DC      #SWCH
      AND     SWCH9     TAKE LOW 4 BITS
      OR      SWCH8     CONSTRUCT SHIFT
      STO     SWCH3
      XIO     SWCH7     READ SWITCHES
      LD      SWCH6     GET SWITCHES
SWCH3 SLA     *-*       PUT PROPER BIT IN BIT 0
      BSC  L  SWCH4,-
      LD    3 @TRUE-X
      BSI   3 POPJ-X
SWCH4 SRA     16
      BSI   3 POPJ-X
***************************************************
SWCH6 DC      *-*
      BSS  E  0
SWCH7 DC      SWCH6     IOCC TO READ DATA SWITCHES
      DC      /3A00
SWCH8 SLA     0
SWCH9 DC      /000F
***************************************************
*     1442 CARD PUNCH OUTPUT HANDLER              *
```

```
**************************************************
      AIF     (@PNCH EQ YES),.YES
O1442 EQU     0
P1142 EQU     0
      AGO     .NO
.YES  AIF     (@READ EQ YES),.YES
O1442 EQU     0
P1142 EQU     0
      AGO     .NO
.YES  ANOP
O1442 DC      *-*
      LD   L  2          CHECK CHAR
      S       O1499
      BSC  L  O1425,+-   BRANCH IF CARRIAGE RETURN
      LD   L2 CRDTB      GET CARD CODE CHAR
      STO  I  O1498      PUT IN BUFFER
      MDX  L  O1498,1    INCR POINTER
      MDX  L  O14BF,1    INCR COUNT
O1420 BSC  I  O1442
O1425 BSI  L  F1442      FLUSH CARD READER INPUT
O1427 LIBF    CARD0      READ A CARD
      DC      /1000
      DC      I14BF
O1430 LIBF    CARD0      WAIT FOR IT
      DC      0
      MDX     O1430
      LDX   2 72         CHECK FOR BLANK
O1435 LD   L2 I14BF
      BSC  L  O1460,Z    BRANCH IF NON-BLANK
      MDX   2 -1
      MDX     O1435
      LIBF    CARD0      SELECT STACKER 2
      DC      /4000
      LD      O14BF
      BSC  I  O1442,+-   RETURN IF NO CHARS TO PUNCH
      LIBF    CARD0      PUNCH A CARD
      DC      /2000
      DC      O14BF
O1445 LIBF    CARD0      WAIT FOR IT
      DC      0
      MDX     O1445
      SRA     16
      STO     O14BF      RESET CHAR COUNT
      LD      O1497
      STO     O1498      RESET POINTER
      MDX     O1420
O1460 MDX  L  $IOCT,0    WAIT OUT ALL PENDING
      MDX     O1460       I/O INTERRUPTS
      LD      O1496      PUT /100B FLAG IN ACC
      BSI  L  $PRET      WAIT FOR OPERATOR
      MDX     O1427
**************************************************
O1496 DC      /100B      FLAG FOR NON-BLANK WAIT
O1497 DC      O14BF+1
O1498 DC      O14BF+1
O1499 DC      @CR-EBCTB
**************************************************
O14BF DC      0          1442 CARD OUTPUT BUFFER
      BSS     72
P1442 DC      *-*        1442 CARD PUNCH PAGESKIP
      LDX   2 -6
P1450 LD   L2 P1499+6    OUTPUT '/*/*/',CR
      BSI  L  OUTPT       (EOF CARD)
      MDX   2 1
      MDX     P1450
      BSC  I  P1442
**************************************************
P1499 DC      @SLSH      /
      DC      @STAR      *
      DC      @SLSH      /
      DC      @STAR      *
      DC      @SLSH      /
      DC      @CR       CR
**************************************************
.NO   ANOP
**************************************************
*     PROG FUNCTION                              *
**************************************************
      DC      @NLAM+1+@LIST
PROG  SRA     16        ZERO COUNT OF BINDINGS
      STO     PRG98
      LD    3 @ARG1-X
      STO     PRG17
PRG05 BSC  L  PRG30,+-  BRANCH IF NO MORE TO BIND
      BSI   3 XCAR-X    GET NEXT ITEM
      STO     PRG22
      S       PRG97
      BSC  L  PRG15,+Z  BRANCH IF NUMBER OR NIL
      S       PRG96
      BSC  L  PRG15,-   BRANCH IF NUMBER
      A       PRG95
      STO     PRG10+1
PRG10 LD   L  *-*
      BSC  L  PRG15,-   BRANCH IF NON-ATOM
      LD   I  PRG22
      EOR     PRG94
      BSC  L  PRG20,Z   BRANCH UNLESS STRING
PRG15 BSI   3 ERROR-X   ERROR IF ANY OF THESE
      DC      43+@MAJR
PRG17 DC      *-*
PRG20 BSI   3 PUSHS-X   PUSH OLD VALUE
PRG22 DC      *-*
      SRA     16        BIND ATOM TO NIL VALUE
      STO  I  PRG22
      MDX  L  PRG98,-1  INCR NEG COUNT OF BINDINGS
      NOP
      LD   I  @ARG1     CHAIN DOWN VAR LIST
      STO   3 @ARG1-X
      MDX     PRG05
PRG30 LD      PRG98     PUSH NEG COUNT OF BINDINGS
      BSI   3 PUSHA-X
      LD    3 @SPDL-X   SAVE CURRENT SPEC PDL LEVEL
      STO     PRG98
      BSI   3 PUSHS-X   PUSH LAST SPEC PDL LEVEL
      DC      PRG99
      LD      PRG98     PUT THIS LEVEL IN SWITCH
      STO     PRG99
      BSI   3 PUSHS-X   PUSH REG PDL LEVEL
      DC      1
      LD    3 @ARG2-X   SAVE LIST OF FORMS TWICE
      BSI   3 PUSHA-X    ONCE FOR GO SEARCHES
      BSI   3 PUSHA-X    ONCE FOR PROG EVALUATION
PRG35 BSC  L  PRG45,+-  BRANCH IF NO FORMS LEFT
      BSI   3 XCAR-X    GET NEXT FORM
      STO   3 @ARG1-X   SAVE AS ARG 1
      BSI   3 XATOM-X
      BSC  L  PRG40,Z   BRANCH IF ATOM
      BSI   3 PUSHJ-X   EVAL FORM
      DC      EVAL
PRG40 LD   I1 0         CHAIN DOWN LIST OF FORMS
      STO   1 0
      MDX     PRG35
PRG45 BSI   3 POPS-X    POP REG PDL LEVEL
      BSI   3 POPS-X    POP SPEC PDL LEVEL SW
      BSI   3 POPN-X    POP BINDINGS
      SRA     16        RETURN NIL
      BSI   3 POPJ-X
**************************************************
PRG94 DC      @STR
PRG95 DC      1+E@FST
PRG96 DC      E@FST-S@FST
PRG97 DC      S@FST
PRG98 DC      *-*
PRG99 DC      0
```

```
**************************************************
*     GO FUNCTION                                *
**************************************************
      DC       @NLAM+1
GO    LD     3 @ARG1-X   GET ARG
      MDX  L  PRG99,0   SKIP IF NOT INSIDE PROG
      MDX      GO20
      STO      GO10
      BSI    3 ERROR-X   ERROR
      DC       46+@MAJR
GO10  DC       *-*
GO20  BSI    3 XATOM-X
      BSC  L  GO30,Z    BRANCH IF ARG IS ATOM
      BSI    3 PUSHJ-X   ELSE EVAL AND TRY AGAIN
      DC       EVAL
      STO    3 @ARG1-X
      MDX      GO20
GO30  LD       PRG99     POP JUNK OFF SPEC PDL
      S      3 @SPDL-X
      SRT      1
      A        GO35       EXCEPT OLD SPEC PDL LEVEL
      BSI    3 PUSHA-X
      BSI    3 POPN-X
      BSI    3 PUSHS-X   RE-PUSH REG PDL LEVEL
GO35  DC       1         ADR OF XR1 AND CONSTANT 1
      MDX    1 -2        RETRIEVE TWO THINGS
      LD     1 1         SEARCH FOR GO TAG
GO40  BSC  L  GO43,+-   BRANCH IF NONE LEFT
      STO      PRG98
      BSI    3 XCAR-X
      EOR    3 @ARG1-X
      BSC  L  GO50,+-   BRANCH IF MATCH
      LD   I  PRG98     ELSE CHAIN DOWN FORMS
      MDX      GO40
GO43  LD     3 @ARG1-X
      STO      GO45
      BSI    3 ERROR-X   ELSE ERROR
      DC       44+@MAJR
GO45  DC       *-*
GO50  LD   I  PRG98     GET REST OF FORMS
      STO    1 0         SAVE FOR PROG TO DO
      MDX      PRG35     GO HAVE PROG DO THEM
**************************************************
*     RETURN FUNCTION                            *
**************************************************
      DC       @LAM+1
RETRN MDX  L  PRG99,0   SKIP IF NOT INSIDE A PROG
      MDX      RET20
      LD     3 @ARG1-X
      STO      RET10
      BSI    3 ERROR-X   ERROR IF SO
      DC       45+@MAJR
RET10 DC       *-*
RET20 LD   L  PRG99     POP JUNK OFF SPEC PDL
      S      3 @SPDL-X    (REG PDL GETS RESTORED)
      SRT      1
      BSI    3 PUSHA-X
      BSI    3 POPN-X
      BSI    3 POPN-X    POP PROG BINDINGS
      LD     3 @ARG1-X   RETURN ARG
      BSI    3 POPJ-X
**************************************************
*     RPLACA/RPLACD FUNCTIONS                    *
**************************************************
      DC       @LAM+2
RPLCA LD       RPLC9     SET UP RPLACA
      MDX      RPLC1
**************************************************
      DC       @LAM+2
RPLCD SRA      16        SET UP RPLACD
RPLC1 STO      RPLC8
      LD     3 @ARG1-X   GET ARG 1
      BSC      +-        SKIP IF NON-NIL
      LD       RPLC7     USE ADR OF NIL IF NIL
      A        RPLC8     GET ADR TO REPLACE
      STO      RPLC3+1
      LD    3 @ARG2-X
RPLC3 STO  L  *-*        SHOVE SECOND ARG THERE
      LD    3 @ARG1-X    RETURN (ALTERED) ARG 1
      BSI   3 POPJ-X
**************************************************
RPLC7 DC       #NIL
RPLC8 DC       *-*
RPLC9 DC       1
**************************************************
*     ASSOC/SASSOC FUNCTIONS                     *
**************************************************
      DC       @LAM+2
ASSOC SRA      16        SET ARG 3 (FN) TO NIL
      STO   3 @ARG3-X     FOR SASSOC
      MDX      SASOC
**************************************************
      DC       @LAM+3    (LAMBDA (X L FN)...
SASOC LD    3 @ARG2-X    GET L
SASC1 BSC  L  SASC5,+-   BRANCH IF NONE LEFT
      BSI   3 XCAR-X
      STO      SASC9     SAVE (CAR L)
      BSI   3 XCAR-X     GET (CAAR L)
      EOR   3 @ARG1-X
      BSC  L  SASC3,+-   BRANCH IF (EQ X (CAAR L))
      LD    I  @ARG2     ELSE CHAIN DOWN L
      STO   3 @ARG2-X
      MDX      SASC1
SASC3 LD       SASC9     RETURN (CAR L)
      BSI   3 POPJ-X
SASC5 LD    3 @ARG3-X    GET FN
      BSC      +-
      BSI   3 POPJ-X     RETURN NIL IF NIL
      SRT      16
      BSI   3 XCONS-X
      STO   3 @ARG1-X    ELSE EVAL AS FUNCTION
      BSC  L  EVAL        OF NO ARGS
**************************************************
SASC9 DC       *-*
**************************************************
*     LENGTH FUNCTION                            *
**************************************************
      DC       @LAM+1
LNGTH SRA      16        ZERO COUNT
      STO      LNTH9
      LD    3 @ARG1-X    GET ARG
LNTH3 BSC  L  LNTH6,+-   BRANCH IF END
      STO      LNTH4+1
LNTH4 LD   L  *-*        ELSE CHAIN DOWN ONE
      MDX  L  LNTH9,1     AND INCR COUNT
      MDX      LNTH3
LNTH6 LD       LNTH9     RETURN COUNT
      BSI   3 MKFXN-X
      BSI   3 POPJ-X
**************************************************
LNTH9 DC       *-*
**************************************************
*     TOPL FUNCTION                              *
**************************************************
      DC       @LAM+1
TOPL  LDX   2 0          SET XR2 TO ZERO
      LD    3 @ARG1-X
      BSC  L  TOPL4,+-   BRANCH IF ARG NIL
      LDX   2 1          SET XR2 TO ONE
      EOR   3 @TRUE-X
      BSC  L  TOPL4,+-   BRANCH IF ARG IS T
      LDX   2 2          SET XR2 TO TWO
      LD    3 @ARG1-X
TOPL4 STO  L  TOPFN      SET TOPFN FOR TOP LEVEL
```

```
      STX  L2 TOPLV     SET TOPLEVEL SW FROM XR2
      SRA     16        RETURN NIL
      BSI   3 POPJ-X
**************************************************
*     TYP/TEND FUNCTIONS                         *
**************************************************
      DC      @NLAM
TYP   LDD     TYTN9     GET DEVICE NUMBERS FOR TYP
      LDX   2 0         SET XR2 TO ZERO
      MDX     TYTN3
**************************************************
      DC      @NLAM
TEND  LDD   3 @SYSP-X   GET DEVICE NUMBERS FOR TEND
      LDX   2 1         SET XR2 TO ONE
TYTN3 STX  L2 TOPLV     SET TOPLEVEL SW FROM XR2
      STO     TYTN8     SAVE OUTPUT DEV NUMBER
      STD   3 @SYSO-X   SET DEFAULT DEV NUMBERS
      RTE     16
      BSI   3 MKFXN-X
      STO  L  #SYSI     SET SYSIN
      LD      TYTN8
      BSI   3 MKFXN-X
      STO  L  #SYSO     SET SYSOUT
      SRA     16
      STO  L  TOPFN     RESET TOPFN
      BSI   3 POPJ-X    RETURN NIL
**************************************************
TYTN8 DC      *-*
      BSS  E  0
TYTN9 DC      1         TYPEWRITER DEV NUMBER
      DC      6         KEYBOARD DEV NUMBER
**************************************************
*     MEMBER FUNCTION                            *
**************************************************
      DC      @LAM+2
MEMBR LD    3 @ARG1-X   SAVE ARG 1
      STO     MEMB9
      LD    3 @ARG2-X   SAVE ARG 2
MEMB1 STO     MEMB4+1
      BSC     +-        SKIP IF ANY LEFT
      BSI   3 POPJ-X    ELSE RETURN NIL
      BSI   3 XCAR-X    GET NEXT ITEM OF ARG 2
      STO   3 @ARG2-X
      LD      MEMB9
      STO   3 @ARG1-X
      BSI   3 PUSHJ-X   COMPARE ARG 1 TO ITEM
      DC      EQUAL
      BSC  L  MEMB6,Z   BRANCH IF EQUAL
MEMB4 LD   L  *-*       ELSE CHAIN DOWN ARG 2
      MDX     MEMB1
MEMB6 LD      MEMB4+1   RETURN WHAT'S LEFT OF ARG 2
      BSI   3 POPJ-X
**************************************************
MEMB9 DC      *-*
**************************************************
*     EQUAL FUNCTION                             *
**************************************************
      DC      @LAM+2    (LAMBDA (X Y)...
EQUAL LD    3 @ARG1-X   COMPARE X AND Y
      EOR   3 @ARG2-X
      BSC  L  EQL15,Z   BRANCH UNLESS (EQ X Y)
EQL10 LD    3 @TRUE-X   RETURN T
      BSI   3 POPJ-X
EQL15 LD    3 @ARG1-X   CHECK X
      BSI   3 XATOM-X
      BSC  L  EQL50,+-  BRANCH UNLESS (ATOM X)
      LD    3 @ARG2-X
      BSI   3 XATOM-X
      BSC  L  EQL25,Z   BRANCH IF (ATOM Y)
EQL20 SRA     16        RETURN NIL
      BSI   3 POPJ-X
EQL25 LD    3 @ARG1-X
```

```
      BSI   3 XNMBP-X
      BSC  L  EQL30,+-  BRANCH UNLESS (NUMBERP X)
      LD    3 @ARG2-X
      BSI   3 XNMBP-X
      BSC  L  EQL20,+-  BRANCH UNLESS (NUMBERP Y)
      LD   I  @ARG1     COMPARE TWO NUMBERS
      EOR  I  @ARG2
      BSC  L  EQL10,+-  RETURN T IF SAME VALUE
      MDX     EQL20     ELSE NIL
EQL30 LD    3 @ARG1-X
      BSI   3 XSTRP-X
      BSC  L  EQL20,+-  BRANCH UNLESS (STRINGP X)
      LD    3 @ARG2-X
      BSI   3 XSTRP-X
      BSC  L  EQL20,+-  BRANCH UNLESS (STRINGP Y)
      MDX  L  @ARG1,1   GET PNAME OF ARG 1
      MDX  L  @ARG2,1   GET PNAME OF ARG 2
EQL35 LD   I  @ARG1     CHAIN DOWN PNAME 1
      STO   3 @ARG1-X
      BSI   3 XCAR-X    GET NEXT CHAR
      STO     EQL99     SAVE IT
      LD   I  @ARG2     CHAIN DOWN PNAME 2
      BSC  L  EQL20,+-  BRANCH IF NONE LEFT
      STO   3 @ARG2-X
      BSI   3 XCAR-X    GET NEXT CHAR
      EOR     EQL99     COMPARE TO OTHER CHAR
      BSC  L  EQL20,Z   BRANCH IF UNEQUAL
      MDX     EQL35     ELSE CHECK REST OF CHARS
EQL40 LD   I  @ARG2     CHECK ARG 2
      BSC  L  EQL10,Z   BRANCH IF NO CHARS LEFT
      MDX     EQL20     ELSE GO RETURN NIL
EQL50 LD    3 @ARG2-X   CHECK Y
      BSI   3 XATOM-X
      BSC  L  EQL20,Z   BRANCH IF (ATOM Y)
      LD   I  @ARG1     SAVE CDR OF EACH ARG
      BSI   3 PUSHA-X
      LD   I  @ARG2
      BSI   3 PUSHA-X
      LD    3 @ARG1-X   GET CAR OF EACH ARG
      BSI   3 XCAR-X
      STO   3 @ARG1-X
      LD    3 @ARG2-X
      BSI   3 XCAR-X
      STO   3 @ARG2-X
      BSI   3 PUSHJ-X   COMPARE TWO CARS
      DC      EQUAL
      BSC  L  EQL55,Z   BRANCH IF EQUAL
      BSI   3 POPA-X    POP TWO CDRS OFF STACK
      BSI   3 POPA-X
      MDX     EQL20     GO RETURN NIL
EQL55 BSI   3 POPA-X    POP TWO CDRS
      STO   3 @ARG2-X
      BSI   3 POPA-X
      STO   3 @ARG1-X
      MDX     EQUAL     GO COMPARE THEM
*************************************************
EQL99 DC      *-*
*************************************************
*     LAST FUNCTION                             *
*************************************************
      DC      @LAM+1
LAST  LD    3 @ARG1-X   GET ARG
      BSC     +-
      BSI   3 POPJ-X    RETURN NIL IF NIL
LAST3 LD   I  @ARG1     IS (CDR ARG) NIL
      BSC  L  LAST5,Z
      LD    3 @ARG1-X   IF SO, RETURN ARG
      BSI   3 POPJ-X
LAST5 STO   3 @ARG1-X   ELSE CHAIN DOWN ARG
      MDX     LAST3
*************************************************
*     RANDOM FUNCTION                           *
```

```
**************************************************
      DC        @LAM+1
RANDM LD     3 @ARG1-X GET ARG
      BSI    3 XNCHK-X   CHECK IT
      DC        #RAND
      BSC   L  RAN40,Z   BRANCH UNLESS ZERO
      LDD       RAN98     DO TWO DISK SEEK OPERATIONS
      BSI       RAN20
      LDD       RAN96
      BSI       RAN20
      SRA       16        RETURN NIL
      BSI    3 POPJ-X
**************************************************
RAN20 DC        *-*
      BSI   L  DISKZ     DO DISK SEEK
      LD        RAN95
RAN30 A         RAN94     WHILE WAITING, KEEP
      MDX   L  $DBSY,0    ALTERING SEED (THIS IS A
      MDX       RAN30      FAIRLY RANDOM PROCESS)
      AND       RAN93     AND OUT HIGH BIT
      OR        RAN92     MAKE SURE IT'S ODD
      STO       RAN95     SAVE IT
      BSC   I  RAN20
**************************************************
RAN40 LD        RAN95     MULTIPLY SEED BY MAGIC
      M         RAN94      NUMBER (899) FOR 1130
      SLT       16         POWER-RESIDUE METHOD
      AND       RAN93     AND OUT HIGH BIT
      STO       RAN95     SAVE IT
      M     I  @ARG1      TREAT AS A 15-BIT FRACTION
      SLT       1          AND MULTIPLY BY ARG
      BSI    3 MKFXN-X   RETURN TRSULT AS NUMBER
      BSI    3 POPJ-X
**************************************************
RAN91 DC        899
RAN92 DC        1
RAN93 DC        /7FFF
RAN94 DC        /2B95     NUMBER GOT BY COIN FLIPS
RAN95 DC        *-*
      BSS   E  0
RAN96 DC        0
      DC        RAN97
RAN97 DC        0
      DC        0
RAN98 DC        0
      DC        RAN99
RAN99 DC        0
      DC        8*20
**************************************************
*     APPEND FUNCTION                            *
**************************************************
      DC        @LAM+@LIST
APPND LD     3 @ARG1-X   GET LIST OF ARGS
      BSC       +-
      BSI    3 POPJ-X    RETURN NIL IF NONE
      BSI    3 PUSHA-X   PUSH LIST OF ARGS
      BSI    3 PUSHA-X   PUSH ROOM FOR FINAL RESULT
      LD    L  1
      BSI    3 PUSHA-X   PUSH ADR FOR APPENDING
APN10 LD    I1 2         IS THERE ONLY ONE LIST LEFT
      BSC   L  APN20,Z   BRANCH IF NOT .
      LD     1 2
      BSI    3 XCAR-X    ELSE GET IT
      STO   I1 0         APPEND AT END
      BSI    3 POPA-X    POP APPEND ADR
      BSI    3 POPA-X    POP RESULT
      RTE       16        SAVE IT
      BSI    3 POPA-X    POP LIST OF ARGS
      RTE       16        GET RESULT AND RETURN
      BSI    3 POPJ-X
APN20 LD     1 2         GET NEXT LIST
      BSI    3 XCAR-X
```

```
APN30 BSC  L  APN40,+-  BRANCH IF NONE OF IT LEFT
      STO     APN35+1   ELSE SAVE IT
      BSI   3 XCAR-X
      SRT     16
      BSI   3 XCONS-X    AND COPY IT
      STO  I1 0         APPEND ITEM TO NEW LIST AND
      STO   1 0          SAVE ADR AS NEW APPEND ADR
APN35 LD   L  *-*       CHAIN DOWN LIST
      MDX     APN30
APN40 LD   I1 2         CHAIN DOWN LIST OF ARGS
      STO   1 2
      MDX     APN10     GO APPEND NEXT ONE
**************************************************
*     MAP/MAPC/MAPLIST/MAPCAR FUNCTIONS          *
**************************************************
      DC      @LAM+1+@LIST
MAP   LDS     2         SET FOR MAP
      MDX     MAP10
**************************************************
      DC      @LAM+1+@LIST
MAPC  LDS     3         SET FOR MAPC
      MDX     MAP10
**************************************************
      DC      @LAM+1+@LIST
MAPLS LDS     0         SET FOR MAPLIST
      MDX     MAP10
**************************************************
      DC      @LAM+1+@LIST
MAPCR LDS     1         SET FOR MAPCAR
MAP10 STS     MAP20     SAVE STATUS BITS
      LD      MAP20      C = DO NOT SAVE RESULTS
      STO     MAP40      O = TAKE CARS OF LISTS
      STO     MAP45
      STO     MAP65
      LDX   2 0         ZERO COUNT OF ARG LISTS
      LD    3 @ARG2-X
      BSC     +-
      BSI   3 POPJ-X    RETURN NIL IF NONE
MAP15 STO   3 @ARG2-X
      BSI   3 XCAR-X
      BSI   3 PUSHA-X   ELSE PUSH AND COUNT
      MDX   2 1          THE ARG LISTS
      LD    I @ARG2
      BSC  L  MAP15,Z
      LD    3 @ARG1-X   SAVE FN
      BSI   3 PUSHA-X
      STX   1 MAP35+1   SAVE ADR TO GET FN AND ARGS
      STX   1 MAP55+1
      STX   1 MAP57+1
MAP20 LDS     *-*
      BSC  L  MAP25,C   BRANCH IF MAP/MAPC
      SRA     16
      BSI   3 PUSHA-X   PUSH NULL RESULT LIST
      LD   L  1
      BSI   3 PUSHA-X   PUSH ADR FOR APPENDS
MAP25 LD   L  2
      BSI   3 PUSHA-X   PUSH NUMBER OF ARG LISTS
MAP30 BSI   3 PUSHA-X   PUSH ROOM FOR NEW ARG LIST
      LD   L  1
      BSI   3 PUSHA-X   PUSH ADR FOR APPENDS
      LD    1 2
      STO  L  2         PUT ARG LIST COUNT IN XR2
MAP35 LD   L2 *-*       GET AN ARG LIST
      BSC  L  MAP60,+-  BRANCH IF EXHAUSTED
MAP40 LDS     *-*
      BSC     O         SKIP IF MAP/MAPLIST
      BSI   3 XCAR-X    TAKE CAR IF MAPC/MAPCAR
      SRT     16
      BSI   3 XCONS-X
      STO  I1 0         APPEND TO NEW ARG LIST
      STO   1 0
      MDX   2 -1        COUNT LISTS
      MDX      MAP35
      BSI   3 POPA-X    POP APPEND ADR
      BSI   3 POPA-X    POP NEW ARG LIST
      STO   3 @ARG2-X
      LD    I  MAP35+1   GET FN
      STO   3 @ARG1-X
      BSI   3 PUSHJ-X   APPLY FN TO ARGS
      DC       APPLY
MAP45 LDS      *-*      SWITCH (ALSO TEMP STORAGE)
      BSC  L  MAP50,C   BRANCH IF MAP/MAPC
      SRT      16
      BSI   3 XCONS-X
      STO  I1 1         APPEND RESULT TO LIST
      STO   1 1
MAP50 LD    1 0
      STO  L  2         PUT ARG LIST COUNT IN XR2
MAP55 LD   I2 *-*       TAKE CDR OF EACH ARG LIST
MAP57 STO  L2 *-*
      MDX   2 -1
      MDX      MAP55
      MDX      MAP30    GO MAP NEXT SET OF ARGS
MAP60 BSI   3 POPA-X    POP APPEND ADR
      BSI   3 POPA-X    POP NEW ARG LIST (UNNEEDED)
      BSI   3 POPA-X    POP ARG LIST COUNT
      STO  L  2         PUT IN XR2
      SRA      16
      STO      MAP45    SET UP NIL RESULT VALUE
MAP65 LDS      *-*
      BSC  L  MAP70,C   BRANCH IF MAP/MAPC
      BSI   3 POPA-X    POP APPEND ADR
      BSI   3 POPA-X    POP RESULT LIST
      STO      MAP45    MAKE IT THE RESULT
MAP70 BSI   3 POPA-X    POP FN
MAP73 BSI   3 POPA-X    POP ARG LISTS
      MDX   2 -1
      MDX      MAP73
      LD       MAP45    RETURN RESULT
      BSI   3 POPJ-X
**************************************************
*     PROG2 FUNCTION                             *
**************************************************
      DC       @LAM+2+@LIST
PROG2 LD    3 @ARG2-X
      BSI   3 POPJ-X
**************************************************
*     REVERSE FUNCTION                           *
**************************************************
      DC       @LAM+1
REVRS SLT      16       SET RESULT IN EXT TO NIL
      LD    3 @ARG1-X   SAVE ARG IN CASE OF GC
      BSI   3 PUSHA-X
RVRS2 BSC  L  RVRS5,+-  BRANCH IF NONE LEFT
      BSI   3 XCAR-X    GET NEXT ITEM OF ARG LIST
      RTE      16
      BSI   3 XCONS-X   CONS ONTO HEAD OF NEW LIST
      RTE      16
      LD   I1 0         CHAIN DOWN ARG LIST
      STO   1 0
      MDX      RVRS2
RVRS5 BSI   3 POPA-X    POP ARG OFF STACK
      RTE      16       GET RESULT FROM EXT
      BSI   3 POPJ-X
**************************************************
*     SUBST FUNCTION                             *
**************************************************
      DC       @LAM+3
SUBST LD    3 @ARG1-X
      STO      SBS99    SAVE ARG 1
      LD    3 @ARG2-X
      STO      SBS98    SAVE ARG 2
      BSI   3 PUSHJ-X   CALL RECURSIVE SUBST-ER
      DC       SBS10
      RTE      16         SAVE RESULT IN EXT
      SRA      16         CLEAR PROTECTED LOCS TO NIL
      STO      SBS98
      STO      SBS99
      RTE      16         RETURN RESULT
      BSI   3 POPJ-X
**************************************************
SBS10 LD    3 @ARG3-X    COMPARE ARG 2 AND ARG 3
      STO   3 @ARG2-X
      LD       SBS98
      STO   3 @ARG1-X
      BSI   3 PUSHJ-X    PUSHJ-X
      DC       EQUAL
      BSC  L  SBS20,+-   BRANCH IF UNEQUAL
      LD       SBS99     ELSE RETURN ARG 1
      BSI   3 POPJ-X
SBS20 LD    3 @ARG3-X    IS ARG 3 AN ATOM
      BSI   3 XATOM-X
      BSC  L  SBS30,+-   BRANCH IF NOT
      LD    3 @ARG3-X    ELSE RETURN ARG 3
      BSI   3 POPJ-X
SBS30 LD   I  @ARG3
      BSI   3 PUSHA-X    SAVE CDR OF ARG 3
      LD    3 @ARG3-X
      BSI   3 XCAR-X     GET CAR OF ARG 3
      STO   3 @ARG3-X
      BSI   3 PUSHJ-X    SUBST INTO CAR
      DC       SBS10
      RTE      16        SAVE RESULT IN EXT
      LD    1 0          GET CDR
      STO   3 @ARG3-X
      RTE      16
      STO   1 0          PUT RESULT ON STACK
      BSI   3 PUSHJ-X    SUBST INTO CDR
      DC       SBS10
      RTE      16
      BSI   3 POPA-X
      RTE      16
      BSI   3 XCONS-X    CONS TWO RESULTS
      BSI   3 POPJ-X
**************************************************
SBS98 DC       NIL       PROTECTED BY TEMLIST
SBS99 DC       NIL       PROTECTED BY TEMLIST
**************************************************
*     REVSTR FUNCTION                            *
**************************************************
      DC       @LAM+1
RVSTR LD    3 @ARG1-X    GET ARG
      BSI   3 XSCHK-X    CHECK IT
      DC       #RVST
      STO   3 @ARG1-X
      BSI   3 PUSHJ-X    REVERSE CHAR LIST
      DC       REVRS
      OR       RVST9     MAKE A STRING OF RESULT
      RTE      16
      LD       RVST8
      BSI   3 XCONS-X
      BSI   3 POPJ-X
**************************************************
RVST9 DC       /8000
RVST8 DC       @STR
**************************************************
*     STRLENGTH FUNCTION                         *
**************************************************
      DC .     @LAM+1
SLNTH LD    3 @ARG1-X    GET ARG
      BSI   3 XSCHK-X    CHECK FOR STRING
      DC       #SLTH
      STO   3 @ARG1-X
      BSC  L  LNGTH      GET LENGTH OF CHAR LIST
**************************************************
*     PNAME FUNCTION                             *
```

```
**************************************************
      DC       @LAM+1
PNAME LDD  I  @ARG1      GET TOP NODE OF ARG
      LD       PNAM9     USE STRING VALUE
      BSI   3 XCONS-X    MAKE A STRING
      BSI   3 POPJ-X
**************************************************
PNAM9 DC       @STR
**************************************************
*     GENSYM FUNCTION                            *
**************************************************
      DC       @LAM+@LIST
GNSYM LD    3 @ARG1-X    IS THERE AN ARG
      BSC  L  GNS20,+-   BRANCH IF NOT
      BSI   3 XCAR-X     IF SO, GET IT
      BSI   3 XSCHK-X    CHECK IT (SHOULD BE STRING)
      DC       #GNSM
      BSC  L  GNS20,+-   BRANCH IF NULL STRING
      STO   3 @ARG1-X
      BSI   3 PUSHJ-X    REVERSE LIST OF CHARS
      DC       REVRS
      STO      GNS99     SAVE LIST OF CHARS
      MDX      GNS40
GNS20 LD       GNS99     GET LIST OF CHARS
GNS25 BSC  L  GNS40,+-   BRANCH IF NONE LEFT
      STO      GNS30+1
GNS30 LDD  L  *-*        GET FIRST NODE
      RTE      16
      S        GNS98     IS CHAR A NUMBER
      BSC  L  GNS40,+Z   BRANCH IF NOT
      A        GNS97     INCREMENT IT
      RTE      16
      STD  I  GNS30+1    PUT IT BACK IN LIST
      RTE      16
      S        GNS96     IS IT NOW OVER 9
      BSC  L  GNS40,+    BRANCH IF NOT
      LD       GNS98     ELSE RESET TO 0
      RTE      16
      STD  I  GNS30+1
      MDX      GNS25     NOW GO INCR NEXT ONE
GNS40 LD       GNS99     GET LIST OF CHARS
      STO   3 @ARG1-X
      BSI   3 PUSHJ-X    REVERSE IT
      DC       REVRS
      OR       GNS95     MAKE IT AN ATOM
      LD       GNS94     VALUE IS UNDEFINED
      BSI   3 XCONS-X
      BSC  L  INTRN      INTERN THE ATOM
**************************************************
GNS94 DC       @UNDF
GNS95 DC       /8000
GNS96 DC       @9
GNS97 DC       @1        EQUALS @0+1
GNS98 DC       @0
GNS99 DC       $GNSM     PROTECTED BY TEMLIST
**************************************************
*     FLATSIZE/FLATC/PRIN1STR/PRINCSTR FUNCTIONS *
**************************************************
      DC       @LAM+1
FLTSZ LD       FLT99     SET UP FOR FLATSIZE
      LDS      0
      MDX      FLT10
**************************************************
      DC       @LAM+1
FLATC LD       FLT99     SET UP FOR FLATC
      LDS      1
      MDX      FLT10
**************************************************
      DC       @LAM+1
PRN1S LD       FLT98     SET UP FOR PRIN1STR
      LDS      0
      MDX      FLT10
```

```
**************************************************
      DC       @LAM+1
PRNCS LD       FLT98      SET UP FOR PRINCSTR
      LDS      1
FLT10 STO   L  OUTSB      SET OUTSB FOR I/O HANDLER
      SRA      16
      STO   L  OUTDV      SET DEVICE NUMBER TO 0
      STO      FLT94      ZERO FLATSIZE/FLATC COUNT
      BSC      0          SKIP IF FLATSIZE/PRIN1STR
      LD       *-1        ELSE SET ACC NON-ZERO
      STO   L  AMPSW      SET AMPSW
      SRA      16
      BSI   3  PUSHA-X    PUSH NULL CHAR LIST
      STX   1  FLT65+1    SAVE ADR FOR APPENDS
      STX   L  OUTCH      SET OUTCH POSITIVE
      LD    3  @ARG1-X    'PRINT' EXPRESSION ONTO
      BSI   3  PUSHJ-X     'DEVICE 0' I/O HANDLER
      DC       PREXP
      BSI   3  POPA-X     POP CHAR LIST
      OR       FLT92      OR IN ATOM MARK
      STO      FLT97      SAVE IT
      LD    L  OUTSB
      EOR      FLT99
      BSC   L  FLT30,Z    BRANCH IF PRIN1STR/PRINCSTR
      LD       FLT94      RETURN COUNT OF CHARS
      BSI   3  MKFXN-X
      BSI   3  POPJ-X
FLT30 LDD      FLT96      RETURN STRING OF CHARS
      BSI   3  XCONS-X
      BSI   3  POPJ-X
**************************************************
FLT92 DC       /8000
FLT93 DC       EBCTB
FLT94 DC       *-*
      BSS   E  0
FLT96 DC       @STR
FLT97 DC       *-*
FLT98 DC       FLT60
FLT99 DC       FLT50
**************************************************
FLT50 DC       *-*        FLATSIZE/FLATC
      MDX   L  FLT94,1    INCR CHAR COUNT
      NOP
      STX   L  OUTCH      SET OUTCH POSITIVE
      BSC   I  FLT50
**************************************************
FLT60 DC       *-*        PRIN1STR/PRINCSTR
      LD    L  2          GET ADR FO CHAR
      A        FLT93
      SRT      16
      LDX   L3 X          XR3 MUST BE SET FOR THIS
      BSI   3  XCONS-X    APPEND TO LIST
FLT65 STO   L  *-*
      STO      FLT65+1
      STX   L  OUTCH      SET OUTCH POSITIVE
      BSC   I  FLT60
**************************************************
*     DEFINEDP FUNCTION                          *
**************************************************
      DC       @LAM+1
DEFNP LD    3  @ARG1-X    CHECK ARG
      S        DEFP9
      BSC   L  DEFP4,+Z   BRANCH IF NUMBER OR NIL
      S        DEFP8
      BSC   L  DEFP4,-    BRANCH IF NUMBER
      A        DEFP7
      STO      DEFP2+1
DEFP2 LD    L  *-*
      BSC   L  DEFP4,-    BRANCH UNLESS ATOM
      LD    I  @ARG1
      S        DEFP6
      BSC   L  DEFP4,Z    BRANCH UNLESS UNDEFINED
      SRA      16         RETURN NIL
      BSI   3 POPJ-X
DEFP4 LD    3 @TRUE-X    RETURN T
      BSI   3 POPJ-X
*****************************************************
DEFP6 DC      @UNDF
DEFP7 DC      1+E@FST
DEFP8 DC      E@FST-S@FST
DEFP9 DC      S@FST
*****************************************************
*     CATENATE FUNCTION                              *
*****************************************************
      DC      @LAM+@LIST
CATN  SRA     16
      BSI   3 PUSHA-X   PUSH NULL LIST OF LISTS
      STX   1 CATN4+1   SAVE ADR FOR APPENDS
      LD    3 @ARG1-X
      BSI   3 PUSHA-X   SAVE LIST OF ARGS
CATN2 BSC  L  CATN6,+-  BRANCH IF NONE LEFT
      BSI   3 XCAR-X    GET NEXT ARG
      BSI   3 XSCHK-X   CHECK IT
      DC      #CATN
      SRT     16
      BSI   3 XCONS-X   APPEND CHAR LIST TO
CATN4 STO  L  *-*        LIST OF LISTS
      STO     CATN4+1
      LD   I1 0         CHAIN DOWN LIST OF ARGS
      STO   1 0
      MDX     CATN2
CATN6 BSI   3 POPA-X    POP LIST OF ARGS
      BSI   3 POPA-X    POP LIST OF CHAR LISTS
      STO   3 @ARG1-X
      BSI   3 PUSHJ-X   APPEND THEM ALL
      DC      APPND
      OR      CATN9     MAKE STRING OF RESULT
      RTE     16
      LD      CATN8
      BSI   3 XCONS-X
      BSI   3 POPJ-X
*****************************************************
CATN8 DC      @STR
CATN9 DC      /8000
*****************************************************
*     REMOB FUNCTION                                 *
*****************************************************
      DC      @NLAM+1
REMOB LD      REMO9     GET ADR OF OBLIST
REMO2 STO     REMO4+1
      LD   I  REMO4+1   GET NEXT ITEM DOWN
      BSC  L  REMO6,+-  BRANCH IF NONE LEFT
      BSI   3 XCAR-X    ELSE COMPARE
      EOR   3 @ARG1-X    IT TO ARG
      BSC  L  REMO4,+-  BRANCH IF THE SAME
      LD   I  REMO4+1   ELSE CHAIN DOWN OBLIST
      MDX     REMO2
REMO4 LD   I  *-*       REMOVE ATOM FROM OBLIST
      STO  I  REMO4+1
REMO6 SRA     16        RETURN NIL
      BSI   3 POPJ-X
*****************************************************
REMO9 DC      #OBLS
*****************************************************
*     SUBSTR FUNCTION                                *
*****************************************************
      DC      @LAM+2+@LIST
SBSTR LD    3 @ARG1-X   GET FIRST ARG
      BSI   3 XSCHK-X   CHECK IT
      DC      #SSTR
      BSC  L  SST27,+-  BRANCH IF NULL STRING
SST10 STO     SST99     SAVE CHAR LIST
      LD    3 @ARG2-X
      BSI   3 XNCHK-X   CHECK SECOND ARG
      DC       #SSTR
      BSC      +
      LD       SST98     USE 1 IF NON-POSITIVE
      STO      SST97
SST15 MDX   L  SST97,-1  COUNT DOWN ARG 2
      BSC      +-Z
      MDX      SST20     BRANCH IF DONE
      LD    I  SST99     CHOP ONE CHAR OFF STRING
      BSC   L  SST27,+-  BRANCH IF NONE LEFT
      STO      SST99
      MDX      SST15
SST20 LD     3 @ARG3-X   CHECK FOR THIRD ARG
      BSC   L  SST30,Z   BRANCH UNLESS NONE
      LD       SST99
SST27 OR       SST96     MAKE A STRING AND RETURN
      RTE      16
      LD       SST95
      BSI    3 XCONS-X
      BSI    3 POPJ-X
SST30 BSI    3 XCAR-X    GET THIRD ARG
      BSI    3 XNCHK-X   CHECK IT
      DC       #SSTR
      BSC   L  SST35,-Z  BRANCH IF POSITIVE
      SRA      16        ELSE RETURN NULL STRING
      MDX      SST27
SST35 STO      SST97     SAVE ARG 3
      LD       SST99
      BSI    3 PUSHA-X   SAVE ROOM FOR CHAR LIST
      STX    1 SST99     SAVE ADR FOR APPENDING
      BSI    3 PUSHA-X   SAVE LIST OF CHARS
SST40 LD     1 0
      BSI    3 XCAR-X    GET A CHAR
      SRT      16
      BSI    3 XCONS-X
      STO   I  SST99     APPEND TO NEW LIST
      STO      SST99
      LD    I1 0         CHAIN DOWN LIST OF CHARS
      BSC   L  SST50,+-  BRANCH IF NO MORE LEFT
      STO    1 0
      MDX   L  SST97,-1  SKIP IF ARG 3 COUNTED OUT
      MDX      SST40
SST50 BSI    3 POPA-X    POP OLD CHAR LIST
      BSI    3 POPA-X    POP NEW CHAR LIST
      MDX      SST27     GO MAKE A STRING
**************************************************
SST95 DC       @STR
SST96 DC       /8000
SST97 DC       *-*
SST98 DC       1
SST99 DC       *-*
**************************************************
*     STRINDEX FUNCTION                          *
**************************************************
      DC       @LAM+2
SINDX LD     3 @ARG1-X
      BSI    3 XSCHK-X   CHECK ARG 1
      DC       #SIDX
      BSC   L  SID45,+-  BRANCH IF NULL STRING
      STO      SID99     SAVE CHAR LIST
      LD     3 @ARG2-X
      BSI    3 XSCHK-X   CHECK ARG 2
      DC       #SIDX
      BSC   L  SID10,Z   BRANCH UNLESS NULL STRING
      LD       SID97     RETURN 1
      MDX      SID45
SID10 STO      SID15+1   SAVE CHAR LIST
      BSI    3 XCAR-X    GET FIRST CHAR OF ARG 2
      STO      SID96     SAVE IT
SID15 LD    L  *-*       GET REST OF ARG 2 CHARS
      STO      SID98     SAVE THEM
      SRA      16
      STO      SID95     ZERO INDEX COUNT
      LD      SID99
SID20 MDX  L  SID95,1   INCR INDEX COUNT
      BSI   3 XCAR-X    GET NEXT CHAR OF ARG 1
      EOR     SID96     COMPARE TO CHAR 1 OF ARG 2
      BSC  L  SID40,Z   BRANCH UNLESS EQUAL
      LD      SID98     COMPARE REST OF ARG 2...
      BSC  L  SID50,+-  BRANCH IF ARG 2 WAS 1 CHAR
      STO     SID94
      LD   I  SID99
SID30 BSC  L  SID45,+-  BRANCH IF ARG 1 NOW SHORT
      STO     SID93     ELSE SAVE REST
      BSI   3 XCAR-X
      STO     SID92     SAVE NEXT CHAR
      LD      SID94
      BSI   3 XCAR-X    GET NEXT CHAR OF ARG 2
      EOR     SID92
      BSC  L  SID40,Z   BRANCH UNLESS CHARS EQUAL
      LD   I  SID94     CHAIN DOWN ARG 2
      BSC  L  SID50,+-  BRANCH IF NONE LEFT
      STO     SID94
      LD   I  SID93     CHAIN DOWN ARG 1 CHARS
      MDX     SID30
SID40 LD   I  SID99     CHAIN DOWN ARG 1
      STO     SID99
      BSC  L  SID20,Z   BRANCH UNLESS NONE LEFT
SID45 BSI   3 MKFXN-X   MAKE A NUMBER AND RETURN
      BSI   3 POPJ-X
SID50 LD      SID95     RETURN STRING POSITION
      MDX     SID45
**************************************************
SID92 DC      *-*
SID93 DC      *-*
SID94 DC      *-*
SID95 DC      *-*
SID96 DC      *-*
SID97 DC      1
SID98 DC      *-*
SID99 DC      *-*
**************************************************
*     PAUSE FUNCTION                             *
**************************************************
      DC      @NLAM
PAUSE MDX  L  $IOCT,0   WAIT OUT ALL PENDING
      MDX     PAUSE      I/O INTERRUPTS
      LDD     PAUS9     PUT PRETTY BITS IN ACC AND
      BSI  L  $PRET      EXT LIGHTS AND WAIT
      SRA     16        REYURN NIL
      BSI   3 POPJ-X
**************************************************
      BSS  E  0
PAUS9 DC      /AAAA
      DC      /5555
**************************************************
*     QUIT FUNCTION                              *
**************************************************
      DC      @NLAM
QUIT  BSI   3 ERROR-X   PRINT SIGN-OFF MESSAGE
      DC      48+@INFO
      EXIT
**************************************************
*     REMOVE FUNCTION                            *
**************************************************
      DC      @LAM+3
REMOV LD    3 @ARG3-X
      BSI   3 XNCHK-X   CHECK ARG 3
      DC      #RMOV
      BSC  L  RMV10,-Z  BRANCH IF POSITIVE
      LD    3 @ARG2-X   ELSE RETURN ARG 2
      BSI   3 POPJ-X
RMV10 STO     RMV99     SAVE ARG 3
      LD    3 @ARG2-X   GET ARG 2
      BSI   3 PUSHA-X   SAVE ROOM FOR NEW LIST
      STX   1 RMV30+1   SAVE ADR FOR APPENDS
      BSI   3 PUSHA-X   SAVE ARG 2
      LD    3 @ARG1-X
      BSI   3 PUSHA-X   SAVE ARG 1
      LD    1 1
RMV20 BSC  L  RMV50,+-  BRANCH IF ARG 2 DONE
      BSI   3 XCAR-X    ELSE GET NEXT ITEM
      STO     RMV98+1
      STO   3 @ARG2-X
      LD    1 0
      STO   3 @ARG1-X
      BSI   3 PUSHJ-X   COMPARE TO ARG 1
      DC      EQUAL
      BSC  L  RMV40,Z   BRANCH UNLESS UNEQUAL
      LDD     RMV98
      BSI   3 XCONS-X   APPEND ITEM TO NEW LIST
RMV30 STO  L  *-*
      STO     RMV30+1
RMV35 LD   I1 1         CHAIN DOWN ARG 2
      STO   1 1
      MDX     RMV20
RMV40 MDX  L  RMV99,-1  DECR COUNT FOR REMOVALS
      MDX     RMV35     IF NOT ZERO TRY AGAIN
      LD   I1 1         ELSE SIMPLY APPEND REST
      STO  I  RMV30+1    OF ARG 2 TO NEW LIST
RMV50 BSI   3 POPA-X    POP ARG 1
      BSI   3 POPA-X    POP ARG 2
      BSI   3 POPA-X    POP RESULT
      BSI   3 POPJ-X
*************************************************
      BSS  E  0
RMV98 DC      NIL
      DC      *-*
RMV99 DC      *-*
*************************************************
*     EXPT FUNCTION                             *
*************************************************
      DC      @LAM+2
EXPT  LD    3 @ARG2-X
      BSI   3 XNCHK-X   CHECK ARG 2
      DC      #EXPT
      STO   3 @ARG2-X   SAVE IT
      LD    3 @ARG1-X
      BSI   3 XNCHK-X   CHECK ARG 1
      DC      #EXPT
      BSC  L  EXP70,+-  RESULT 0 OF BASE=0
      S       EXP99
      BSC  L  EXP20,+-  RESULT 1 IF BASE=1
      A       EXP98
      BSC  L  EXP40,Z   BRANCH UNLESS BASE =-1
      LD    3 @ARG2-X
      BSC  L  EXP30,E   BRANCH IF ODD EXPONENT
EXP20 LD      EXP99     RETURN 1
      MDX     EXP70
EXP30 LD      EXP98     RETURN -1
      MDX     EXP70
EXP40 LD    3 @ARG2-X   CHECK EXPONENT
      BSC  L  EXP50,-
      SRA     16        RETURN 0 IF NEGATIVE
      MDX     EXP70
EXP50 BSC  L  EXP20,+   RE TURN 1 IF ZERO
      LD      EXP99     PUT 1 IN ACC
EXP60 M    I  @ARG1     MULTIPLY BY BASE
      RTE     16
      MDX  L  @ARG2,-1  DO IT 'EXPONENT' TIMES
      MDX     EXP60
EXP70 BSI   3 MKFXN-X   MAKE A NUMBER AND RETURN
      BSI   3 POPJ-X
*************************************************
EXP97 DC      -1
EXP98 DC      2
EXP99 DC      1
```

```
**************************************************
*     READSTR FUNCTION                           *
**************************************************
      DC       @LAM+1
RDSTR LD    3 @ARG1-X
      BSI   3 XSCHK-X   CHECK ARG
      DC       #RDST
      BSI   3 PUSHA-X   SAVE CHAR LIST ON STACK
      STO      RDS55     AND IN OTHER PLACES
      STO      RDS65+1
      STO      RDS75
      SRA      16
      STO   L  INPKC    CLEAR DEVICE 0 PEEK CHAR
      STO   L  INDEV    SET INPUT DEV NUMBER TO 0
      BSI   3 PUSHJ-X   READ FROM 'DEVICE 0'
      DC       RD005
      RTE      16       SAVE RESULT
      BSI   3 POPA-X    POP CHAR LIST
      RTE      16       RETURN RESULT FROM READ
      BSI   3 POPJ-X
**************************************************
RDS50 DC       *-*
      LDX  L3 X         XR3 MUST BE SET FOR THIS
      LD       RDS65+1  ARE THERE ANY CHARS LEFT
      BSC   L  RDS60,Z  BRANCH IF SO
      BSI   3 ERROR-X   ELSE ERROR
      DC       49+@MAJR
RDS55 DC       *-*
RDS60 BSI   3 XCAR-X    GET NEXT CHAR
      RTE      16
RDS65 LD    L  *-*      CHAIN DOWN CHAR LIST
      STO      RDS65+1
      RTE      16       RETURN CHAR
      BSC   I  RDS50
**************************************************
RDS70 DC       *-*
      BSI   3 ERROR-X   READ MUST HAVE CAUSED AN
      DC       50+@MAJR  ERROR - KICK IN ANOTHER
RDS75 DC       *-*       TWO CENTS' WORTH
**************************************************
*     SUBLIS FUNCTION                            *
**************************************************
      DC       @LAM+2
SBLIS LD    3 @ARG1-X   SAVE ARG 1 IN CASE OF GC
      STO      SBL99
      BSI   3 PUSHJ-X   CALL RECURSIVE SUBLIS-ER
      DC       SBL10
      SRA      16       CLEAR PROTECTED
      STO      SBL99     LOC TO NIL
      RTE      16       RETURN RESULT
      BSI   3 POPJ-X
**************************************************
SBL10 LD    3 @ARG2-X   CHECK ARG 2
      BSI   3 XATOM-X
      BSC   L  SBL40,+- BRANCH UNLESS ATOM
      LD       SBL99    SEARCH ARG 1...
SBL20 BSC   L  SBL35,+- BRANCH IF NONE LEFT
      STO      SBL98
      BSI   3 XCAR-X    GET CAR OF ARG 1
      STO      SBL25+1
      BSI   3 XCAR-X    GET CAAR OF ARG 1
      EOR   3 @ARG2-X   COMPARE TO ARG 2
      BSC   L  SBL30,Z  BRANCH UNLESS EQUAL
SBL25 LD    L  *-*      RETURN CDAR OF ARG 1
SBL27 RTE      16
      LD       *-1      NON-ZERO ACC MEANS CHANGE
      BSI   3 POPJ-X
SBL30 LD    I  SBL98    CHAIN DOWN ARG 1
      MDX      SBL20
SBL33 BSI   3 POPA-X
SBL35 LD    3 @ARG2-X   RETURN ARG 2
      RTE      16
      SRA      16        ZERO ACC MEANS NO CHANGE
      BSI    3 POPJ-X
SBL40 LD     3 @ARG2-X
      BSI    3 XCAR-X
      BSI    3 PUSHA-X   SAVE CAR OF ARG 2
      LD     I @ARG2
      STO    3 @ARG2-X   GET CDR OF ARG 2
      BSI    3 PUSHJ-X   SUBLIS IT
      DC       SBL10
      STO      SBL98     SAVE FLAG
      LD     1 0         GET CAR OF ARG 2
      STO    3 @ARG2-X
      RTE      16
      STO    1 0         SAVE SUBLIS RESULT
      LD       SBL98
      BSI    3 PUSHA-X   SAVE FLAG
      BSI    3 PUSHJ-X   SUBLIS THE CAR
      DC       SBL10
      STO      SBL98     SAVE FLAG
      BSI    3 POPA-X    GET FLAG FROM CDR
      OR       SBL98     DID EITHER CHANGE
      BSC    L SBL33,+-  BRANCH IF NOT
      BSI    3 POPA-X    POP SUBLIS OF CDR
      BSI    3 XCONS-X   CONS THE SUBLIS RESULTS
      MDX      SBL27
**************************************************
SBL98 DC       *-*
SBL99 DC       NIL       PROTECTED BY TEMLIST
**************************************************
*     PGSKP FUNCTION                             *
**************************************************
      DC       @LAM+1
PGSKP BSI    L STOUT     SET OUTPUT DEVICE
      DC       #PSKP
      LD       PSKP9
      BSI    L OUTPT     OUTPUT CARRIAGE RETURN
      LDX   I2 OUTDV
      LDX   I3 $XR3X
      BSI   I2 OPSKP     CALL PAGESKIP SUBROUTINE
      LDX   L3 X
      LD     3 @ARG1-X   RETURN ARG (DEV NUMBER)
      BSI    3 POPJ-X
**************************************************
PSKP9 DC       @CR
**************************************************
*     LET/FLET FILE NAME LOOKUP ROUTINE          *
**************************************************
**************************************************
      AIF      (@IDSK EQ YES),.YES
      AIF      (@ODSK EQ NO),.NO
.YES  ANOP
LTFLT DC       *-*
      STX   L2 LT270+1   SAVE XR2
      LD     I LTFLT     GET FN NAME
      STO      LT010
      STO      LT060
      MDX    L LTFLT,1
      LD     3 @ARG2-X   GET ARG 2
      BSC    L LT020,+-  ASSUME 0 IF NONE
      BSI    3 XCAR-X
      BSI    3 XNCHK-X   ELSE CHECK FOR NUMBER
LT010 DC       *-*
LT020 STO      LT040
      STO      LT130
      STO      LT250
      STO    L LT300
      SLA      12        SAVE IN SHIFTED FORM
      STO    3 @ARG2-X
      LDX   I2 LT040     PUT DRIVE NUMBER IN XR2
      BSC    L LT030,+Z  BAD IF NEGATIVE
      S        LT901
      BSC    L LT030,-   BAD IF GREATER THAN 4
      LD   L2 $ULET     GET DISK ADR OF LET
      BSC  L  LT050,Z   BAD IF NONE ON CURRENT JOB
LT030 BSI   3 ERROR-X   ERROR - LOGICAL DRIVE
      DC      59+@MAJR   NOT ON CURRENT JOB
LT040 DC      *-*
LT050 SLT     32
      STD     LT902     CLEAR FILE NAME WORK AREA
      LD    3 @ARG1-X   GET ARG 1
      STO     LT090
      STO     LT120
      STO     LT240
      STO     LT290
      BSI   3 XSCHK-X   CHECK FOR STRING
LT060 DC      *-*
      LDX   2 25
LT070 BSC  L  LT100,+-  DONE IF NO CHARS LEFT
      STO     LT080+1   SAVE CHARS
      BSI   3 XCAR-X
      BSI   3 XCDR-X    GET EBCDIC TABLE ENTRY
      AND     LT903     TRUNCATE EBCDIC TO 6 BITS
      SRT     24
      SLT   2           POSITION FOR NEXT CHAR
      SRT     1
      AD      LT902     PUT INTO NAME CODE
      STD     LT902
LT080 LD   L  *-*       CHAIN DOWN LIST OF ADRS
      MDX   2 -6
      MDX     LT070
      BSC  L  LT100,+-  BRANCH UNLESS MORE THAN 5
      BSI   3 ERROR-X   PRINT WARNING - USE FIRST 5
      DC      55+@MINR
LT090 DC      *-*
LT100 LDX  I2 LT040     PUT DRIVE NUMBER IN XR2
      LD      LT902     CHECK FILE NAME
      OR      LT902+1
      BSC  L  LT200,Z   BRANCH UNLESS NULL/BLANK
      LD   L2 $FPAD     USE WORKING STORAGE
      AND     LT904     COMPUTE ITS LENGTH
      STO     LT902
      LD      LT905
      S       LT902
      BSC  L  LT140,-Z  ERROR IF NOT EVEN 1 SECTOR
LT110 BSI   3 ERROR-X
      DC      60+@MAJR
LT120 DC      *-*
LT130 DC      *-*
LT140 RTE     16        PUT LENGTH IN EXT
      LD   L2 $FPAD     GET DISK ADR IN ACC
      MDX     LT270
*************************************************
LT901 DC      /5000
LT902 BSS  E  2
LT903 DC      /3F00
LT904 DC      /0FFF
LT905 DC      8*200
*************************************************
LT200 SRA     16        CLEAR CUMULATIVE
      STO     LT910     DISK BLOCK COUNT
      LD   L2 $ULET     GET DISK ADR OF LET
LT210 OR    3 @ARG2-X   OR IN DRIVE CODE
      STO  L  DSKBF+1   SAVE IN DISK BUFFER
      LDD     LT911     READ A SECTOR OF LET/FLET
      BSI  L  DISKZ
LT220 MDX  L  $DBSY,0
      MDX     LT220
      LD      LT912     GET 3 TIMES NUMBER OF
      S    L  DSKBF+5    ENTRIES IN THIS LET/FLET
      STO     LT913      SECTOR AND SAVE
      LDX  L2 -315
LT230 LD   L2 DSKBF+323 GET NAME FROM NEXT ENTRY
      RTE     16
      LD   L2 DSKBF+322
      SD       LT902     COMPARE TO REQUESTED NAME
      SLT      2
      BSC      +-
      RTE      16
      BSC  L  LT280,Z   BRANCH IF DIFFERENT
      LD   L2 DSKBF+322 CHECK TYPE CODE
      SRT      14
      BSC  L  LT260,+ZE BRANCH UNLESS NOT DATA FILE
      BSI   3 ERROR-X
      DC       54+@MAJR
LT240 DC       *-*
LT250 DC       *-*
LT260 LD   L2 DSKBF+324 GET DISK BLOCK COUNT
      SRA      4         CONVERT TO SECTOR COUNT
      BSC  L  LT110,+   ERROR IF LESS THAN 1
      RTE      16
      LD       LT910     COMPUTE SECTOR ADDRESS
      SRA      4
      A    L  DSKBF+3
      OR    3 @ARG2-X   OR IN DRIVE CODE
LT270 LDX  L2 *-*       RESTORE XR2
      BSC  I  LTFLT     RETURN
LT280 LD       LT910     INCREMENT CUMULATIVE
      A    L2 DSKBF+324  DISK BLOCK COUNT OF
      STO      LT910      ENTRIES ALREADY SEEN
      MDX   2 3
      NOP
      MDX  L  LT913,-3  SKIP IF ALL ENTRIES SEEN
      MDX      LT230     ELSE GO LOOK AT NEXT ONE
      LD   L  DSKBF+6   GET LET/FLET CHAIN ADR
      BSC  L  LT310,Z   BRANCH UNLESS NO MORE
      BSI   3 ERROR-X   FILE NAME NOT IN LET/FLET
      DC       53+@MAJR
LT290 DC       *-*
LT300 DC       *-*
LT310 OR    3 @ARG2-X   OR IN DRIVE CODE
      S    L  DSKBF+1   COMPARE SECTOR ADRS
      BSC  L  LT320,-   BRANCH UNLESS NEXT IS FLET
      SRA      16        START OF FLET - CLEAR
      STO      LT910     CUMULATIVE DB COUNT
LT320 LD   L  DSKBF+6   GET CHAIN ADR AGAIN
      MDX      LT210     GO GET NEXT LET/FLET SECTOR
*****************************************************
LT910 DC       *-*
      BSS  E  0
LT911 DC .     0
      DC       DSKBF
LT912 DC       315
LT913 DC       *-*
*****************************************************
.NO   ANOP
*****************************************************
*     DISK FILE INPUT DEVICE HANDLER                *
*****************************************************
      AIF      (@IDSK EQ YES),.YES
IDISK EQU      0
FDISK EQU      0
      AGO      .NO
.YES  ANOP
IDISK DC       *-*
      MDX  L  IDKBF+1,0 SKIP IF NO INPUT FILE OPEN
      MDX      ID010
      LDX  I1 INPT5+1   RESTORE XR1
      LDX  L3 X         RESTORE XR3
      BSI   3 ERROR-X   ERROR - NO INPUT FILE
      DC       51+@MAJR
ID010 MDX  L  ID901,0   SKIP UNLESS FLUSH REQUESTED
      MDX      ID100     GO FLUSH
      MDX  L  ID902,0   SKIP IF NO CHARS LEFT
      MDX      ID030
ID015 BSI      ID200     GET RECORD, SKIP IF EOF
      MDX      ID020
      MDX  L  REDSW,0   SKIP IF IN MIDDLE OF READ
      MDX     ID010     ELSE GO TRY AGAIN
      BSC  L  RDEOF     GO HANDLE READ EOF ERROR
ID020 STO     ID040+1   SAVE POINTER TO RECORD
      LD      ID904
      STO     ID902     SET CHAR COUNT
ID030 LD      ID040+1   GET POINTER
      EOR     ID905     FLIP BIT 0
      BSC     +Z        SKIP IF BIT 0 IS NOW 0
      A       ID906     ELSE INCR POINTER
      STO     ID040+1
      SLA     1         PUT BIT 0 IN CARRY
ID040 LD   L  *-*       GET CHAR IN RIGHT-HAND
      BSC     C          HALF OF ACC
      SRA     8
      AND     ID907
      STO     ID903     SAVE EBCDIC CHAR
      LDX   2 -L@EBC    SEARCH TABLE
ID050 LD   L2 EBCTB+L@EBC
      SRA     8
      EOR     ID903
      BSC  L  ID060,+-
      MDX   2 1
      MDX     ID050
      LDX   2 -L@EBC    USE BLANK IF NOT FOUND
ID060 LD   L  2         CALCULATE ADR
      A       ID050+1
      MDX  L  ID902,-1  DECR CHAR COUNT
      NOP
      BSC  I  IDISK     RETURN
ID100 SRA     16
      STO     ID901     CLEAR FLUSH SWITCH
ID110 BSI     ID200     READ CARD, SKIP IF EOF
      MDX     ID110     IF NOT, TRY AGAIN
      MDX     ID015     IF SO, TRY TO READ CHAR
***************************************************
ID901 DC      0         NON-ZERO = FLUSH REQUEST
ID902 DC      *-*
ID903 DC      *-*
ID904 DC      72
ID905 DC      /8000
ID906 DC      1
ID907 DC      /00FF
***************************************************
ID200 DC      *-*
      MDX  L  ID940,0   SKIP IF NO RECORD LEFT
      MDX     ID240
      MDX  L  ID941,0   SKIP IF NO SECTOR LEFT
      MDX     ID220
ID205 LD   L  IDKBF+1   GET LOGICAL DRIVE NUMBER
      SRA     12
      STO     ID210
      SRA     4
      STO  L  IDKBF+1   CLEAR SECTOR ADR IN BUFFER
      STO  L  IDKBF+1
      LDX  I1 INPT5+1   RESTORE XR1
      LDX  L3 X         RESTORE XR3
      BSI   3 ERROR-X   ERROR - FILE EXHAUSTED
      DC      56+@MAJR
ID210 DC      *-*
ID220 LDD     ID942
      BSI  L  DISKZ     READ NEXT SECTOR
ID230 MDX  L  $DBSY,0
      MDX     ID230
      MDX  L  IDKBF+1,1 INCR SECTOR ADR
      MDX  L  ID941,-1  DECR SECTOR COUNT
      NOP
      LD      ID943     SET RECORD COUNT
      STO     ID940
      LD      ID944     SET RECORD POINTER
      STO     ID945
ID240 MDX  L  ID945,40  INCR RECORD POINTER
```

```
      MDX  L  ID940,-1  DECR RECORD COUNT
      NOP
      LD      ID945
      A       ID906
      STO     ID250+1
      A       ID946
      STO     ID260+1
ID250 LDD  L  *-*       GET FIRST FOUR CHARS
      SD      ID947
      BSC     +-
      RTE     16
      BSC  L  ID280,Z   BRANCH UNLESS /*/*
ID260 LD   L  *-*       GET FIFTH CHAR
      EOR     ID947
      SRA     8
      BSC  L  ID270,Z   BRANCH UNLESS /
      MDX  L  ID200,1   INCR RETURN ADR FOR EOF
ID270 LD      ID945     RETURN RECORD POINTER
      BSC  I  ID200
ID280 LDD  I  ID250+1   GET FIRST FOUR CHARS
      SD      ID948
      BSC     +-
      RTE     16
      BSC  L  ID905,+-  BRANCH IF ALL 0-8-2 PUNCHES
      MDX     ID270     ELSE GO RETURN
**************************************************
ID940 DC      0         ZERO = NO RECORD LEFT
ID941 DC      0         ZERO = NO SECTOR LEFT
      BSS  E  0
ID942 DC      0
      DC      IDKBF
ID943 DC      8
ID944 DC      IDKBF+1-40
ID945 DC      *-*
ID946 DC      2
      BSS  E  0
ID947 EBC     ./*/*.
ID948 EBC     ..    FOUR 0-8-2 PUNCHES
**************************************************
FDISK DC      *-*
      STX     ID901     SET FLUSH SWITCH
      BSC  I  FDISK
**************************************************
*     INDISK FUNCTION                            *
**************************************************
      DC      @LAM+1+@LIST
INDSK BSI  L  LTFLT     LOOK UP FILE IN LET/FLET
      DC      #IDSK
      STO     IDKBF+1   SAVE DISK ADR
      RTE     16
      STO     ID941     SAVE SECTOR COUNT
      SRA     16
      STO     ID940     CLEAR RECORD COUNT
      STO     ID902     CLEAR CHAR COUNT
      STO     ID901     CLEAR FLUSH SWITCH
      BSI   3 POPJ-X
**************************************************
      BSS  E  0
IDKBF DC      320
      DC      0         ZERO = NO FILE OPENED
      BSS  E  320
**************************************************
.NO   ANOP        THIS IS AN IMPORTANT CARD ABE
      LIST
      HDNG    101 FIXED-POINT NUMBER SPACE
*************************************************
*     FXS - FIXED-POINT NUMBER SPACE            *
*************************************************
S@FXB BSS     16        BIT TABLE FOR FXS GC
E@FXB EQU     *
L@FXB EQU     E@FXB-S@FXB
S@FXS EQU     *         FIXED-POINT NUMBER SPACE
*************************************************
*     FIXED-POINT NUMBERS                       *
*************************************************
$SYSR DC      @ISTD     SYSREAD VALUE
$SYSP DC      @OSTD     SYSPRINT VALUE
$SYSH DC      @PSTD     SYSPUNCH VALUE
$SYSI DC      @ISTD     SYSIN VALUE
$SYSO DC      @OSTD     SYSOUT VALUE
*************************************************
S@FXF BSS     16*L@FXB-*+S@FXS
E@FXS EQU     *
L@FXS EQU     E@FXS-S@FXS
L@FXF EQU     E@FXS-S@FXF
      LIST
      HDNG    102 FREE STORAGE (START)
***************************************************
*     FST - FREE STORAGE SPACE                    *
***************************************************
S@FST BSS  E  0          FREE STORAGE SPACE
***************************************************
@UNDF EQU     1          MARKER FOR UNDEFINED VALUE
@STR  EQU     2          MARKER FOR CHARACTER STRING
@LAM  EQU     /0000      FUNCTION
@NLAM EQU     /4000       TYPE
@MLAM EQU     /8000       INDICATORS
@LIST EQU     /2000      1 MORE ARG FOR LIST
@ATOM EQU     /8000      BIT 0 IN CAR MARKS AN ATOM
***************************************************
*     LIST OF ALL ATOMS (OBLIST)                  *
***************************************************
$OBLS BSS  E  0
#@CR  ATOM1   #@CR,CR
#ABS  SUBR3   ABS,A,B,S
#ADD1 SUBR4   ADD1,A,D,D,1
#AND  SUBR3   AND,A,N,D
#APND SUBR6   APPND,A,P,P,E,N,D
#APPL SUBR5   APPLY,A,P,P,L,Y
#ASOC SUBR5   ASSOC,A,S,S,O,C
#ATOM SUBR4   ATOM,A,T,O,M
#BOOL SUBR5   BOOLE,B,O,O,L,E
#C@R  ATOM3   @UNDF,C,DASH,R
#CAR  SUBR3   CAR,C,A,R
#CATN SUBR8   CATN,C,A,T,E,N,A,T,E
#CDR  SUBR3   CDR,C,D,R
#CHRC SUBR5   CHRCT,C,H,R,C,T
#COND SUBR4   COND,C,O,N,D
#CONS SUBR4   CONS,C,O,N,S
#CR   ATOM2   #@CR,C,R
#DDTI ATOM5   NIL,D,D,T,I,N
#DEFP SUBR8   DEFNP,D,E,F,I,N,E,D,P
#DEP  SUBR3   DEP,D,E,P
#DIFF SUBR4   DIFF,D,I,F,F
#EQ   SUBR2   EQ,E,Q
#EQL  SUBR5   EQUAL,E,Q,U,A,L
#ERR  SUBR3   XERR,E,R,R
#ERLS ATOM7   NIL,E,R,R,L,I,S,T
#ERST SUBR6   ERSET,E,R,R,S,E,T
#EVAL SUBR4   EVAL,E,V,A,L
#EXAM SUBR4   EXAM,E,X,A,M
#EXPT SUBR4   EXPT,E,X,P,T
#FLTC SUBR5   FLATC,F,L,A,T,C
#FLSZ SUBR8   FLTSZ,F,L,A,T,S,I,Z,E
#GC   SUBR2   GC,G,C
#GCD  SUBR3   GCD,G,C,D
#GCGA ATOM5   #T,G,C,G,A,G
#GNSM SUBR6   GNSYM,G,E,N,S,Y,M
#GO   SUBR2   GO,G,O
#HEX  ATOM3   NIL,H,E,X
#IDVP SUBR6   IDEVP,I,N,D,E,V,P
#IDSK SUBR6   INDSK,I,N,D,I,S,K
#INTN SUBR6   INTRN,I,N,T,E,R,N
#KBEC ATOM6   #T,K,B,E,C,H,O
#LABL ATOM5   @UNDF,L,A,B,E,L
#LAM  ATOM6   @UNDF,L,A,M,B,D,A
#LAST SUBR4   LAST,L,A,S,T
#LNTH SUBR6   LNGTH,L,E,N,G,T,H
#LESP SUBR5   LESSP,L,E,S,S,P
#LINE SUBR5   LINEL,L,I,N,E,L
#LIST SUBR4   LIST,L,I,S,T
#LSH  SUBR3   LSH,L,S,H
#MAP  SUBR3   MAP,M,A,P
#MAPC SUBR4   MAPC,M,A,P,C
#MPCR SUBR6   MAPCR,M,A,P,C,A,R
#MPLS SUBR7   MAPLS,M,A,P,L,I,S,T
#MAX  SUBR3   MAX,M,A,X
#MEMB SUBR6   MEMBR,M,E,M,B,E,R
```

```
#MIN  SUBR3   MIN,M,I,N
#MNUS SUBR5   MINUS,M,I,N,U,S
#MNSP SUBR6   MNUSP,M,I,N,U,S,P
#MLAM ATOM7   @UNDF,M,L,A,M,B,D,A
#NIL  ATOM3   NIL,N,I,L
#NLAM ATOM7   @UNDF,N,L,A,M,B,D,A
#NOT  SUBR3   NOT,N,O,T
#NULL SUBR4   NULL,N,U,L,L
#NMBP SUBR7   NMBRP,N,U,M,B,E,R,P
#OBLS ATOM6   $OBLS,O,B,L,I,S,T
#OR   SUBR2   OR,O,R
#ODVP SUBR7   ODEVP,O,U,T,D,E,V,P
#PAUS SUBR5   PAUSE,P,A,U,S,E
#PEKC SUBR5   PEEKC,P,E,E,K,C
#PKCH SUBR6   PEKCH,P,E,E,K,C,H
#PSKP SUBR5   PGSKP,P,G,S,K,P
#PLUS SUBR4   PLUS,P,L,U,S
#PNAM SUBR5   PNAME,P,N,A,M,E
#PRNC SUBR5   PRINC,P,R,I,N,C
#PRCS SUBR8   PRNCS,P,R,I,N,C,S,T,R
#PRNT SUBR5   PRINT,P,R,I,N,T
#PRN1 SUBR5   PRIN1,P,R,I,N,1
#PR1S SUBR8   PRN1S,P,R,I,N,1,S,T,R
#PROG SUBR4   PROG,P,R,O,G
#PRG2 SUBR5   PROG2,P,R,O,G,2
#QUIT SUBR4   QUIT,Q,U,I,T
#QUOT SUBR5   QUOTE,Q,U,O,T,E
#QUO  SUBR8   QUO,Q,U,O,T,I,E,N,T
#RAND SUBR6   RANDM,R,A,N,D,O,M
#READ SUBR4   READ,R,E,A,D
#REDC SUBR5   READC,R,E,A,D,C
#RDCH SUBR6   REDCH,R,E,A,D,C,H
#RDST SUBR7   RDSTR,R,E,A,D,S,T,R
#REM  SUBR9   REM,R,E,M,A,I,N,D,E,R
#RMOB SUBR5   REMOB,R,E,M,O,B
#RMOV SUBR6.  REMOV,R,E,M,O,V,E
#RTRN SUBR6   RETRN,R,E,T,U,R,N
#RVRS SUBR7   REVRS,R,E,V,E,R,S,E
#RVST SUBR6   RVSTR,R,E,V,S,T,R
#RPLA SUBR6   RPLCA,R,P,L,A,C,A
#RPLD SUBR6   RPLCD,R,P,L,A,C,D
#SASC SUBR6   SASOC,S,A,S,S,O,C
#SET  SUBR3   SET,S,E,T
#SETQ SUBR4   SETQ,S,E,T,Q
#STQQ SUBR5   SETQQ,S,E,T,Q,Q
#SIDX SUBR8   SINDX,S,T,R,I,N,D,E,X
#STRP SUBR7   STRP,S,T,R,I,N,G,P
#SLTH SUBR9   SLNTH,S,T,R,L,E,N,G,T,H
#SBLS SUBR6   SBLIS,S,U,B,L,I,S
#SUBR ATOM4   @UNDF,S,U,B,R
#SBST SUBR5   SUBST,S,U,B,S,T
#SSTR SUBR6   SBSTR,S,U,B,S,T,R
#SUB1 SUBR4   SUB1,S,U,B,1
#SWCH SUBR6   SWTCH,S,W,I,T,C,H
#SYSI ATOM4   #SYSI,S,Y,S,I
#SYSO ATOM4   #SYSO,S,Y,S,O
#SYSH ATOM6   $SYSH,S,Y,S,P,C,H
#SYSP ATOM5   $SYSP,S,Y,S,P,R
#SYSR ATOM5   $SYSR,S,Y,S,R,D
#T    ATOM1   #T,T
#TEND SUBR4   TEND,T,E,N,D
#TIMS SUBR5   TIMES,T,I,M,E,S
#TOPL SUBR4   TOPL,T,O,P,L
#TYI  SUBR3   TYI,T,Y,I
#TYO  SUBR3   TYO,T,Y,O
#TYP  SUBR3   TYP,T,Y,P
#ZERP SUBR5   ZEROP,Z,E,R,O,P,-1
****************************************************
```

```
      HDNG    LIST OF AREAS PROTECTED FROM GC
*************************************************
*     TEMLIST                                   *
*************************************************
$TMLS DC      *+1
      DC      XCNS9     TEMP FOR XCONS
      DC      *+1
      DC      XCNS9+1   TEMP FOR XCONS
      DC      *+1
      DC      INT88     TEMP FOR INTRN
      DC      *+1
      DC      INT95     #OBLS - OBLIST
      DC      *+1
      DC      XCAR9     #NIL - NIL
      DC      *+1
      DC      PR200+1   #HEX - HEX
      DC      *+1
      DC      GC705+1   #GCGA - GCGAG
      DC      *+1
      DC      IKB05+1   #DDTI - DDTIN
      DC      *+1
      DC      SYS02+1   #SYSO - SYSOUT
      DC      *+1
      DC      RD920     TEMP FOR READ
      DC      *+1
      DC      RD964     TEMP FOR READ
      DC      *+1
      DC      RD968     #QUOT - QUOTE
      DC      *+1
      DC      INT82     #C@R - C-R
      DC      *+1
      DC      IKB72+1   #KBEC - KBECHO
      DC      *+1
      DC      EV930     #SUBR - SUBR
      DC      *+1
      DC      AP997     #LABL - LABEL
      DC      *+1
      DC      AP902     #LAM - LAMBDA
      DC      *+1
      DC      AP998     #NLAM - NLAMBDA
      DC      *+1
      DC      AP999     #MLAM - MLAMBDA
      DC      *+1
      DC      SYSI2+1   #SYSI - SYSIN
      DC      *+1
      DC      TOPFN     FOR HOLDING USER TOPLEVEL
      DC      *+1
      DC      LSPR2+1   #ERLS - ERRLIST
      DC      *+1
      DC      SBS98     TEMP FOR SUBST
      DC      *+1
      DC      SBS99     TEMP FOR SUBST
      DC      *+1
      DC      GNS99     CHAR STRING FOR GENSYM
      DC      *+1
      DC      SBL99     TEMP FOR SUBLIS
      DC      NIL
      DC      @TRUE     #T - T
*************************************************
*     LIST OF CHARS FOR GENSYM FUNCTION         *
*************************************************
$GNSM DC      *+1       STRING OF CHARACTERS
      DC      @9         FOR USE BY GENSYM
      DC      *+1        FUNCTION - GENERATED
      DC      @9         ATOMS WILL BE QX000,
      DC      *+1        QX001, QX002, ETC.
      DC      @9
      DC      *+1
      DC      @X
      DC      NIL
      DC      @Q
      LIST
      HDNG    200 END OF FREE STORAGE
**************************************************
S@FSF BSS  E  4800      EMPTY FREE STORAGE SPACE
E@FST EQU     *
L@FST EQU     E@FST-S@FST
L@FSF EQU     E@FST-S@FSF
**************************************************
*     PUSHDOWN LIST SPACE                        *
**************************************************
S@SPD EQU     *         START OF SPECIAL PDL
      BSS     1000      PUSHDOWN LIST SPACE
S@RPD EQU     *-1       START OF REGULAR PDL
**************************************************
      LIST
**************************************************
*     MAIN CONTROL PROGRAM FOR LISP              *
**************************************************
LISP  LDX  L1 S@RPD     INIT XR1 FOR REG PDL
      LDX  L3 X         INIT XR3 FOR SPECIAL AREA
      LDX  L2 L@FSF     CLEAR FREE STORAGE
      SLT     32
LSP10 STD  L2 S@FSF-2
      MDX   2 -2
      MDX     LSP10
      BSI   3 PUSHJ-X   DO GARBAGE COLLECTION
      DC      GC
      LD   L  $KCSW
      BSC  L  LSP20,+-  BRANCH UNLESS // TYP
      BSI   3 PUSHJ-X   SET UP FOR KB INPUT
      DC      TYP
LSP20 BSI   3 ERROR-X   PRINT HEADER
      DC      0+@INFO
      SRA     16        INITIALIZE RANDOM FUNCTION
      BSI   3 MKFXN-X
      STO   3 @ARG1-X
      BSI   3 PUSHJ-X
      DC      RANDM
LSP25 LDX  L2 1         SET TOPLEVEL TYPE -
TOPLV EQU     *-1        0=TYP, 1=TEND, 2=USER FORM
      BSC  I2 LSP30     BRANCH TO PROPER TOPLEVEL
**************************************************
LSP30 DC      LSP35     TYP TOPLEVEL
      DC      LSP35     TEND TOPLEVEL
      DC      LSP60     USER TOPLEVEL
**************************************************
LSP35 BSI  L  SYSOU     SET SYSTEM OUTPUT
      LD   L  OUTDV     SAVE OUTPUT DEV NUMBER
      STO     LSP99
      BSI  L  SYSIN     SET SYSTEM INPUT
      LD   L  INDEV     GET INPUT DEV NUMBER
      BSI   3 MKFXN-X
      STO   3 @ARG1-X
      BSI   3 PUSHJ-X   READ AN EXPRESSION
      DC      READ
      MDX  L  TOPLV,0   SKIP IF TEND TOPLEVEL
      BSI     LSP50     ELSE PRINT EXPRESSION
      STO   3 @ARG1-X
      BSI   3 PUSHJ-X   EVAL EXPRESSION
      DC      EVAL
      BSI     LSP50     PRINT RESULT
      LD      LSP98     PRINT TWO CARRIAGE RETURNS
      BSI  L  OUTPT
      LD      LSP98
      BSI  L  OUTPT
      MDX     LSP25     GO DO IT AGAIN
**************************************************
LSP50 DC      *-*
      SRT     16
      BSI   3 XCONS-X   MAKE LIST OF EXPRESSION
      BSI   3 PUSHA-X   SAVE ON STACK
      LD      LSP99
      BSI   3 MKFXN-X
      STO   3 @ARG1-X   SET OUTPUT DEVICE NUMBER
      BSI   3 POPA-X    GET EXPRESSION
      STO   3 @ARG2-X
      BSI   3 PUSHJ-X   PRINT IT
      DC      PRINT
      BSC  I  LSP50
*****************************************************
TOPFN DC      NIL       PROTECTED BY TEMLIST
LSP98 DC      @CR
LSP99 DC      *-*
*****************************************************
LSP60 LD      TOPFN     GET FORM GIVEN BY USER
      STO   3 @ARG1-X
      BSI   3 PUSHJ-X   EVAL IT
      DC      EVAL
      MDX     LSP25     GO DO IT AGAIN
*****************************************************
*     ERRORS NOT INSIDE AN ERRSET BRANCH HERE      *
*****************************************************
LSPER LDX  L1 S@RPD     INIT XR1 FOR REG PDL
LSPR2 LD   L  #ERLS     (MAPCAR EVAL ERRLIST)...
      BSI   3 PUSHA-X
LSPR3 BSC  L  LSPR5,+-  BRANCH IF NO FORMS LEFT
      BSI   3 XCAR-X    GET NEXT ONE
      STO   3 @ARG1-X
      BSI   3 PUSHJ-X   EVAL IT
      DC      EVAL
      LD   I1 0         CHAIN DOWN LIST OF FORMS
      STO   1 0
      MDX     LSPR3
LSPR5 BSI   3 POPA-X
      MDX     LSP25     GO TO TOP LEVEL
*****************************************************
*     SPARE WORDS FOR MODSF PATCHING               *
*****************************************************
MODSF EBC     .******************************.
      EBC     .******************************.
*****************************************************
@END  EQU     *
      END     LISP
// DUP
*DELETE             LISP
*STORE      WS  UA  LISP
*DELETE             TLISP
*DUMP       UA  WS  LISP
*STORECI    WS  UA  TLISP
```

```
// JOB                                          LISPMACS    G STEELE
// DUP          MACRO LIBRARY FOR LISP ASSEMBLIES
*DELETE             LMACS
*DFILE          FX  LMACS 0008
*MACRO UPDATE
BUILD 'LMACS'
SELECT M
.**********************************************************************
NAME $LABL,$ADR,$1,$
ADD 'SUBR1'
                             LIST    OFF
                       #     SET     $+1
                             DC      #**+#*7
                             DC      *
                       $LABL DC      *+3
                             DC      *+@ATOM
                             DC      NIL
                             DC      @.$1
                             DC      $ADR-1
                             DC      #SUBR
                             LIST
.**********************************************************************
NAME $LABL,$ADR,$1,$2,$
ADD 'SUBR2'
                             LIST    OFF
                       #     SET     $+1
                             DC      #**+#*9
                             DC      *
                       $LABL DC      *+5
                             DC      *+@ATOM
                             DC      *+1
                             DC      @.$1
                             DC      NIL
                             DC      @.$2
                             DC      $ADR-1
                             DC      #SUBR
                             LIST
.**********************************************************************
NAME $LABL,$ADR,$1,$2,$3,$
ADD 'SUBR3'
                             LIST    OFF
                       #     SET     $+1
                             DC      #**+#*11
                             DC      *
                       $LABL DC      *+7
                             DC      *+@ATOM
                             DC      *+1
                             DC      @.$1
                             DC      *+1
                             DC      @.$2
                             DC      NIL
                             DC      @.$3
                             DC      $ADR-1
                             DC      #SUBR
                             LIST
.**********************************************************************
NAME $LABL,$ADR,$1,$2,$3,$4,$
ADD 'SUBR4'
                             LIST    OFF
                       #     SET     $+1
                             DC      #**+#*13
                             DC      *
                       $LABL DC      *+9
                             DC      *+@ATOM
                             DC      *+1
                             DC      @.$1
                             DC      *+1
                             DC      @.$2
                             DC      *+1
                             DC      @.$3
                             DC      NIL
```

```
                        DC      @.$4
                        DC      $ADR-1
                        DC      #SUBR
                        LIST
.**********************************************************************
NAME $LABL,$ADR,$1,$2,$3,$4,$5,$
ADD 'SUBR5'
                        LIST    OFF
                #       SET     $+1
                        DC      #**+#*15
                        DC      *
                $LABL   DC      *+11
                        DC      *+@ATOM
                        DC      *+1
                        DC      @.$1
                        DC      *+1
                        DC      @.$2
                        DC      *+1
                        DC      @.$3
                        DC      *+1
                        DC      @.$4
                        DC      NIL
                        DC      @.$5
                        DC      $ADR-1
                        DC      #SUBR
                        LIST
.**********************************************************************
NAME $LABL,$ADR,$1,$2,$3,$4,$5,$6,$
ADD 'SUBR6'
                        LIST    OFF
                #       SET     $+1
                        DC      #**+#*17
                        DC      *
                $LABL   DC      *+13
                        DC      *+@ATOM
                        DC      *+1
                        DC      @.$1
                        DC      *+1
                        DC      @.$2
                        DC      *+1
                        DC      @.$3
                        DC      *+1
                        DC      @.$4
                        DC      *+1
                        DC      @.$5
                        DC      NIL
                        DC      @.$6
                        DC      $ADR-1
                        DC      #SUBR
                        LIST
.**********************************************************************
NAME $LABL,$ADR,$1,$2,$3,$4,$5,$6,$7,$
ADD 'SUBR7'
                        LIST    OFF
                #       SET     $+1
                        DC      #**+#*19
                        DC      *
                $LABL   DC      *+15
                        DC      *+@ATOM
                        DC      *+1
                        DC      @.$1
                        DC      *+1
                        DC      @.$2
                        DC      *+1
                        DC      @.$3
                        DC      *+1
                        DC      @.$4
                        DC      *+1
                        DC      @.$5
                        DC      *+1
                        DC      @.$6
                        DC      NIL
```

```
                                DC      @.$7
                                DC      $ADR-1
                                DC      #SUBR
                                LIST
.*************************************************************************
NAME $LABL,$ADR,$1,$2,$3,$4,$5,$6,$7,$8,$
ADD 'SUBR8'
                                LIST    OFF
                         #      SET     $+1
                                DC      #**+#*21
                                DC      *
                         $LABL  DC      *+17
                                DC      *+@ATOM
                                DC      *+1
                                DC      @.$1
                                DC      *+1
                                DC      @.$2
                                DC      *+1
                                DC      @.$3
                                DC      *+1
                                DC      @.$4
                                DC      *+1
                                DC      @.$5
                                DC      *+1
                                DC      @.$6
                                DC      *+1
                                DC      @.$7
                                DC      NIL
                                DC      @.$8
                                DC      $ADR-1
                                DC      #SUBR
                                LIST
.*************************************************************************
NAME $LABL,$ADR,$1,$2,$3,$4,$5,$6,$7,$8,$9,$
ADD 'SUBR9'
                                LIST    OFF
                         #      SET     $+1
                                DC      #**+#*23
                                DC      *
                         $LABL  DC      *+19
                                DC      *+@ATOM
                                DC      *+1
                                DC      @.$1
                                DC      *+1
                                DC      @.$2
                                DC      *+1
                                DC      @.$3
                                DC      *+1
                                DC      @.$4
                                DC      *+1
                                DC      @.$5
                                DC      *+1
                                DC      @.$6
                                DC      *+1
                                DC      @.$7
                                DC      *+1
                                DC      @.$8
                                DC      NIL
                                DC      @.$9
                                DC      $ADR-1
                                DC      #SUBR
                                LIST
.*************************************************************************
NAME $LABL,$VAL,$1,$
ADD 'ATOM1'
                                LIST    OFF
                         #      SET     $+1
                                DC      #**+#*5
                                DC      *
                         $LABL  DC      $VAL
                                DC      *+@ATOM
                                DC      NIL
                         DC      @.$1
                         LIST
.*************************************************************************
NAME $LABL,$VAL,$1,$2,$
ADD 'ATOM2'
                         LIST    OFF
                 #       SET     $+1
                         DC      #**+#*7
                         DC      *
                 $LABL   DC      $VAL
                         DC      *+@ATOM
                         DC      *+1
                         DC      @.$1
                         DC      NIL
                         DC      @.$2
                         LIST
.*************************************************************************
NAME $LABL,$VAL,$1,$2,$3,$
ADD 'ATOM3'
                         LIST    OFF
                 #       SET     $+1
                         DC      #**+#*9
                         DC      *
                 $LABL   DC      $VAL
                         DC      *+@ATOM
                         DC      *+1
                         DC      @.$1
                         DC      *+1
                         DC      @.$2
                         DC      NIL
                         DC      @.$3
                         LIST
.*************************************************************************
NAME $LABL,$VAL,$1,$2,$3,$4,$
ADD 'ATOM4'
                         LIST    OFF
                 #       SET     $+1
                         DC      #**+#*11
                         DC      *
                 $LABL   DC      $VAL
                         DC      *+@ATOM
                         DC      *+1
                         DC      @.$1
                         DC      *+1
                         DC      @.$2
                         DC      *+1
                         DC      @.$3
                         DC      NIL
                         DC      @.$4
                         LIST
.*************************************************************************
NAME $LABL,$VAL,$1,$2,$3,$4,$5,$
ADD 'ATOM5'
                         LIST    OFF
                 #       SET     $+1
                         DC      #**+#*13
                         DC      *
                 $LABL   DC      $VAL
                         DC      *+@ATOM
                         DC      *+1
                         DC      @.$1
                         DC      *+1
                         DC      @.$2
                         DC      *+1
                         DC      @.$3
                         DC      *+1
                         DC      @.$4
                         DC      NIL
                         DC      @.$5
                         LIST
.*************************************************************************
NAME $LABL,$VAL,$1,$2,$3,$4,$5,$6,$
```

```
ADD 'ATOM6'
                          LIST    OFF
                    #     SET     $+1
                          DC      #**+#*15
                          DC      *
                    $LABL DC      $VAL
                          DC      *+@ATOM
                          DC      *+1
                          DC      @.$1
                          DC      *+1
                          DC      @.$2
                          DC      *+1
                          DC      @.$3
                          DC      *+1
                          DC      @.$4
                          DC      *+1
                          DC      @.$5
                          DC      NIL
                          DC      @.$6
                          LIST
.*********************************************************************
NAME $LABL,$VAL,$1,$2,$3,$4,$5,$6,$7,$
ADD 'ATOM7'
                          LIST    OFF
                    #     SET     $+1
                          DC      #**+#*17
                          DC      *
                    $LABL DC      $VAL
                          DC      *+@ATOM
                          DC      *+1
                          DC      @.$1
                          DC      *+1
                          DC      @.$2
                          DC      *+1
                          DC      @.$3
                          DC      *+1
                          DC      @.$4
                          DC      *+1
                          DC      @.$5
                          DC      *+1
                          DC      @.$6
                          DC      NIL
                          DC      @.$7
                          LIST
.*********************************************************************
NAME $LABL,$VAL,$1,$2,$3,$4,$5,$6,$7,$8,$
ADD 'ATOM8'
                          LIST    OFF
                    #     SET     $+1
                          DC      #**+#*19
                          DC      *
                    $LABL DC      $VAL
                          DC      *+@ATOM
                          DC      *+1
                          DC      @.$1
                          DC      *+1
                          DC      @.$2
                          DC      *+1
                          DC      @.$3
                          DC      *+1
                          DC      @.$4
                          DC      *+1
                          DC      @.$5
                          DC      *+1
                          DC      @.$6
                          DC      *+1
                          DC      @.$7
                          DC      NIL
                          DC      @.$8
                          LIST
.*********************************************************************
NAME $LABL,$VAL,$1,$2,$3,$4,$5,$6,$7,$8,$9,$
```

```
ADD 'ATOM9'
                        LIST    OFF
                  #     SET     $+1
                        DC      #**+#*21
                        DC      *
                  $LABL DC      $VAL
                        DC      *+@ATOM
                        DC      *+1
                        DC      @.$1
                        DC      *+1
                        DC      @.$2
                        DC      *+1
                        DC      @.$3
                        DC      *+1
                        DC      @.$4
                        DC      *+1
                        DC      @.$5
                        DC      *+1
                        DC      @.$6
                        DC      *+1
                        DC      @.$7
                        DC      *+1
                        DC      @.$8
                        DC      NIL
                        DC      @.$9
                        LIST
.                 ****************************************************
NAME
ADD '      '
                        LIST    ON
                  ****************************************************
                  *     YOU LEFT A BLANK IN HERE, IDIOT **************
                        ORG     * AT THIS ADDRESS, FOOL **************
                  ****************************************************
                        LIST
.*********************************************************************
ENDUP
```

```
// JOB                                KBCP0
// ASM
*LEVEL 4
              HDNG    KEYBOARD/CONSOLE PRINTER I/O ROUTINE
              LIBR
              ISS  02 KBCP0
        **************************************************
        *     COMMON LIBF ENTRY                          *
        **************************************************
        KBCP0 MDX     WHICH     GO PROCESS REQUEST
        EXIT  BSC  I  *-*       RETURN TO LIBF CALLER
        **************************************************
        RESET DC      /0F01     SENSE/RESET IOCC
        **************************************************
        *     COMMON INTERRUPT ENTRY POINT               *
        **************************************************
        INTRP DC      *-*
              XIO     RESET-1   RESET INTERRUPTING DEVICE
              BSC  L  KBINT,-   BRANCH IF KB INTERRUPT
              LD      TBUFR     IS ANOTHER CHAR IN BUFFER
              BSC  L  TTERM,E   BRANCH IF NONE
              STO     CHAR      SAVE CHAR FROM BUFFER
              LDX   1 -15       PUSH DOWN CHARS IN BUFFER
        TMOVE LD   L1 TBUFR+16
              STO  L1 TBUFR+15
              MDX   1 1
              MDX     TMOVE
              MDX  L  POINT,-1  MOVE BACK POINTER
              LD      ONE
              STO     TBUFR+15  CLEAR LAST WORD OF BUFFER
        TREDY XIO     SENSE     CHECK FOR TYPEWRITER READY
              SLA     5
              BSC  L  TTYPE,-   BRANCH IF READY
              LD      FLAG
              BSI  L  $PST4     ELSE WAIT AT $PST4
              MDX     TREDY      AND TRY AGAIN
        TTYPE XIO     PRINT     PRINT NEXT CHAR
              MDX     RETRN+2   RETURN - DON'T DECR I/O CTR
        TTERM SRA     16        CLEAR BUSY SWITCH
              STO     TBUSY
        RETRN MDX  L  $IOCT,-1  DECR I/O COUNTER
              NOP
              BSC  I  INTRP     RETURN TO ILS04
        KBINT XIO     READ      READ CHAR FROM KEYBOARD
              MDX     RETRN
        **************************************************
        *     CONSTANTS                                  *
        **************************************************
              BSS  E  0
        CHAR  DC      *-*
        SENSE DC      /0F00     SENSE WITHOUT RESET IOCC
        READ  DC      INPUT
              DC      /0A00     READ KEYBOARD CHAR IOCC
        PRINT DC      CHAR
              DC      /0900     PRINT CHAR IOCC
        FLAG  DC      /2000     FLAG FOR I/O TRAP WAITS
        SLECT DC      /0C00     SELECT KB IOCC
        BLINK DC      /1111     JUST FOR FUN - PATTERN
              DC      /1111      FOR BLINKING LIGHTS
        ONE   DC      1
        TBUFR DC      1         OUTPUT BUFFER
              DC      1
              DC      1
              DC      1
              DC      1
              DC      1
              DC      1
              DC      1
              DC      1
              DC      1
              DC      1
              DC      1
```

```
      DC       1
      DC       1
      DC       1
      DC       1
POINT DC       TBUFR     BUFFER POINTER
TBUSY DC       0         NON-ZERO=OUTPUT IN PROGRESS
INPUT DC       *-*
*************************************************
*     HANDLE CONSOLE PRINTER OUTPUT             *
*************************************************
WHICH BSC  L  RDCHR,E   /0001 = KB INPUT REQUEST
      STO     INPUT     SAVE CHAR TEMPORARILY
RMCHK LD      TBUFR+15  WAIT FOR SPACE IN BUFFER
      BSC  L  *+1,E
      MDX     RMCHK
      LD      INPUT     PUT CHAR IN BUFFER
      STO  I  POINT
      MDX  L  TBUSY,0   IS OUTPUT GOING ALREADY
      MDX     PUTBF      YES, GO INCR POINTER
      LD      TBUFR      NO, START I/O
      STO     CHAR
      LD      ONE
      STO     TBUFR
READY XIO     SENSE     CHECK FOR TYPEWRITER READY
      SLA     5
      BSC  L  START,-
      LD      FLAG
      BSI  L  $PRET     IF NOT, WAIT IN $PRET
      MDX     READY      AND TRY AGAIN
START MDX  L  $IOCT,1   INCR I/O COUNTER
      STX     TBUSY     SET BUSY SWITCH
      XIO     PRINT     PRINT CHAR
      MDX     EXIT
PUTBF MDX  L  POINT,1   INCR BUFFER POINTER
      MDX     EXIT
*************************************************
*     HANDLE KEYBOARD INPUT                     *
*************************************************
RDCHR MDX  L  $IOCT,1   INCR I/O COUNTER
      STO     INPUT
      XIO     SLECT     SELECT KB FOR INPUT
      LDD     BLINK     BLINK PRETTY LIGHTS
      RTE     1          IN ACC AND EXT
      STD     BLINK
IWAIT LDD     BLINK
      BSI  L  $PRET     WAIT IN $PRET FOR INPUT
      LD      INPUT     DID KB INPUT
      BSC  L  IWAIT,E    NO, WAIT AGAIN
      MDX     EXIT       YES, RETURN WITH CHAR
*************************************************
$PRET EQU     /28       PREOPERATIVE WAIT TRAP
$IOCT EQU     /32       I/O COUNTER
$PST4 EQU     /8D       LEVEL 4 INT ERROR TRAP
*************************************************
      END
// DUP
*DELETE             KBCP0
*STORE      WS  UA  KBCP0              KEYBOARD/CONSOLE PRINTER I/O SUBROUTINE
```

```
(SETQQ SPRINT (LAMBDA (P L N M) (PROG (F G H)
Q     (AND (LESSP N (DIFF (CHRCT P) 15))
           (PRINC P ,                ,) (GO Q))
R     (AND (LESSP N (DIFF (CHRCT P) 3)) (PRINC P ,    ,) (GO R))
S     (AND (LESSP N (CHRCT P)) (PRINC P , ,) (GO S))
      (AND (OR (ATOM L) (LESSP (PLUS M -1 (FLATSIZE L)) (CHRCT P)))
           (RETURN (PRIN1 P L)))
      (PRINC P ,(,)
      (SETQ F (EQ (CAR L) 'PROG))
      (ERRSET (AND
           (NOT (ATOM (CDR L)))
           (OR F (SETQ N (MAXPAN (CDR L) (DIFF (CHRCT P) (FLATSIZE
                (CAR L)) 1))))
           (OR (ATOM (CAR L)) (NOT (LESSP (MAXPAN (CDR L) (CHRCT P)) N)))
           (PROG NIL
                (ERRSET (SETQ G (LESSP (MAXPAN (LAST L) (PLUS (FLATSIZE
                     (LAST L)) (CHRCT P) (MINUS (FLATSIZE L)))) N)))
          A     (PRIN1 P (CAR L))
                (PRINC P , ,)
                (AND (CDR (SETQ L (CDR L))) G (GO A))) ))
      (SETQ N (CHRCT P))
      (SETQ H (MEMBER (CAR L) '(LAMBDA NLAMBDA MLAMBDA LABEL)))
B     (SPRINT P (CAR L)
           (COND ((SETQ G (AND F (CAR L)(ATOM (CAR L)))) (PLUS N 5))(N))
           (COND ((NULL (SETQ L (CDR L))) (ADD1 M))
                ((ATOM L) (PLUS 4 M (FLATSIZE L))) (0)))
      (COND ((ATOM L) (AND L (PRINC P , . ,) (PRIN1 P L))
           (RETURN (PRINC P ,),)) ))
      (COND (H (SETQQ H NIL) (PRINC P , ,))
           ((OR (LESSP (CHRCT P) N) (AND G (ATOM (CAR L)))) (PRINT P)))
      (GO B)  )))
/*/*/
(SETQQ MAXPAN (LAMBDA (L N) (PROG (G)
      (SETQQ G 0)
A     (SETQ G (PLUS G (PANMAX (CAR L) N (COND
           ((NULL (SETQ L (CDR L))) (ADD1 M))
           ((ATOM L) (PLUS M 4 (FLATSIZE L)))
           (0) ))))
      (AND (ATOM L) (RETURN G))
      (GO A) )))
/*/*/
(SETQQ PANMAX (LAMBDA (L N M) (COND
      ((LESSP (PLUS M -1 (FLATSIZE L)) N) 1)
      ((OR (LESSP N 3) (ATOM L)) (ERR '(50)))
      ((AND (NOT (ATOM (CDR L))) (ATOM (CAR L)) (SETQ N (DIFF N 1
           (FLATSIZE (CAR L)))) (SETQ L (CDR L)) NIL))
      ((MAXPAN L (SUB1 N)))  )))
/*/*/
(SETQQ GRIND (NLAMBDA (P . X)
      (PGSKP (SETQ P (EVAL P)))
      (MAPC '(LAMBDA (L) (SPRINT P (LIST 'SETQQ L (EVAL L)) (LINEL P) 0)
           (PRINT P CR)) X)
      (PGSKP P) ))
(TYP)
```

```
(SETQ T*COUNT 0)(SETQ T*PDL NIL)(SETQ T*FNS NIL)(SETQ T*DEV SYSPR)
/*/*/
(SETQQ T*ARGS (NLAMBDA (T*F T*C . T*A) (COND
((AND (SWITCH 15) (EVAL T*C))
   (PRINC T*DEV CR T*COUNT ,  , (LENGTH T*PDL) ,  ENTERING , T*F CR)
   (MAPC '(LAMBDA (X) (PRINC T*DEV ,     , X , = ,)
       (PRIN1 T*DEV (CDR X) CR)) T*A)))
(SETQ T*PDL (CONS T*COUNT T*PDL))
(SETQ T%COUNT (ADD1 T*COUNT)) ))
/*/*/
(SETQQ T*RESULT (NLAMBDA (T*F T*C T*R)
(SETQ T*R (EVAL T*R))
(COND((AND (SWITCH 15) (EVAL T*C))
   (PRINC T*DEV CR (CAR T*PDL) ,  , (SUB1 (LENGTH T*PDL))
         ,  EXITING , T*F CR ,     RESULT = ,)
   (PRIN1 T*DEV T*R CR) ))
(SETQ T*PDL (CDR T*PDL))
T*R ))
/*/*/
(SETQQ T*SUBR (NLAMBDA (ARGS T*F T*S T*A T*R)
(SETQ ARGS (CDR ARGS))
(COND ((AND (SWITCH 15) (EVAL T*A))
   (PRINC T*DEV CR T*COUNT ,  , (LENGTH T*PDL) ,  ENTERING , T*F CR)
   (MAPC '(LAMBDA (X) (PRINC T*DEV ,     ,)(PRIN1 T*DEV X CR)) ARGS)))
(SETQ T*PDL (CONS T*COUNT T*PDL))
(SETQ T*COUNT (ADD1 T*COUNT))
(SETQ T*S (APPLY T*S ARGS))
(COND ((AND (SWITCH 15) (EVAL T*R))
   (PRINC T*DEV CR (CAR T*PDL) ,  , (SUB1 (LENGTH T*PDL)) ,  EXITING ,
           T*F CR ,     ,) (PRIN1 T*DEV T*S CR)))
(SETQ T*PDL (CDR T*PDL))
T*S))
/*/*/
(SETQQ TRACE (NLAMBDA (F A R)        (COND
((MEMBER F T*FNS) NIL)
((EQ (CADR F) 'SUBR) (SET F (LIST (COND ((ZEROP (LSH (CDDR F) -14))
     'LAMBDA) ('NLAMBDA)) 'T*X (LIST 'T*SUBR 'T*X F (CDR F) A R))))
((SET F (CONS (CADR F) (CONS (CADDR F) (CONS (CONS 'T*ARGS (CONS
     F (CONS A ((LABEL Q (LAMBDA (X) (COND ((NULL X) NIL)
     ((ATOM X) (LIST X)) (T(CONS (CAR X) (Q (CDR X)))) )))(CADDR F)) )))
(CADDDR F)) )))     (RPLACA (LAST (CDR F)) (LIST 'T*RESULT F R
                    (LAST (CDR F))) )))
(SETQ T*FNS (CONS F T*FNS)) F))
/*/*/
(SETQQ UNTRACE (NLAMBDA (F) (COND
((NOT (MEMBER F T*FNS)) NIL)
((EQ 'T*SUBR (CAADDDR F)) (SET F (CADDADDDR F))
     (SETQ T*FNS (REMOVE F T*FNS 50))      F)
((SET F (CONS (CADR F) (CONS(CADDR F) (CADDDDR F))))
        (RPLACA (LAST (CDR F)) (CADDDR (LAST (CDR F)))
         (SETQ T*FNS (REMOVE F T*FNS 50)) F))        ))
/*/*/
(TOPL '(EVAL (READ 2)))
/*/*/
```

```
// XEQ TLISP
(TOPL'(EVAL(READ 2)))
(SETQQ DERIV (LAMBDA (V E) (D E)))
/*/*/
(SETQQ D (LAMBDA (E) (COND
     ((EQ E V) 1)
     ((ATOM E) 0)
     ((EQ (CAR E) 'PLUS) (*PLUS (D (CADR E)) (D (CADDR E))))
     ((EQ (CAR E) 'DIFF) (*DIFF (D (CADR E)) (D (CADDR E))))
     ((EQ (CAR E) 'MINUS) (*MINUS (D (CADR E))))
     ((EQ(CAR E) 'TIMES) (*PLUS (*TIMES (CADR E) (D (CADDR E)))
          (*TIMES (CADDR E) (D (CADR E)))))
     ((EQ (CAR E) 'EXPT) (*PLUS (*TIMES (D (CADR E)) (*TIMES (CADDR E)
          (*EXPT (CADR E) (*DIFF (CADDR E) 1)))) (*TIMES E (*TIMES
          (*LN (CADR E)) (D (CADDR E))))))
     ((EQ (CAR E) 'LN) (*TIMES (*EXPT (CADR E) -1) (D (CADR E))))
     ((EQ (CAR E) 'SIN) (*TIMES (D (CADR E)) (*COS (CADR E))))
     ((EQ (CAR E) 'COS) (*TIMES (D (CADR E)) (*MINUS (*SIN (CADR E)))))
     ((LIST 'DERIV V E)) )))
/*/*/
(SETQQ M (NLAMBDA (E P) (MM (EVAL E) P)))
/*/*/
(SETQQ MM (LAMBDA (E P) (COND
     ((EQ P '*E))
     ((EQ P '*N)(NUMBERP E))
     ((ATOM P)(EQUAL E P))
     ((ATOM E) NIL)
     ((MM (CAR E) (CAR P)) (MM (CDR E)(CDR P))) )))
/*/*/
(SETQQ *PLUS (LAMBDA X (COND
     ((M X (*N *N))(PLUS (CAR X) (CADR X)))
     ((EQUAL (CAR X)(CADR X)) (*TIMES 2 (CAR X)))
     ((M X ((MINUS *E) *E)) (*DIFF (CADR X)(CADAR X)))
     ((M X (*E (MINUS *E))) (*DIFF (CAR X) (CADADR X)))
     ((M X (*E *N)) (*PLUS (CADR X) (CAR X)))
     ((M X (*N *E)) (COND
          ((ZEROP (CAR X)) (CADR X))
          ((MINUSP (CAR X)) (*DIFF (CADR X) (MINUS (CAR X))))
          ((M (CADR X) (PLUS *N *E))
               (*PLUS (PLUS (CAR X) (CADADR X)) (CADDADR X)))
          ((M (CADR X) (DIFF *N *E))
               (*DIFF (PLUS (CAR X) (CADADR X)) (CADDADR X)))
          ((CONS 'PLUS X)) ))
     ((M X ((LN *E) (LN *E)))(*LN (*TIMES (CADAR X) (CADADR X))))
     ((CONS 'PLUS X)) )))
/*/*/
(SETQQ *DIFF (LAMBDA X (COND
     ((M X (*N *N)) (DIFF (CAR X)(CADR X)))
     ((EQUAL (CAR X) (CADR X)) 0)
     ((M X ((MINUS *E) *E)) (*MINUS (*PLUS (CAR X)(CADR X))))
     ((M X (*E (MINUS *E))) (*PLUS (CAR X) (CADR X)))
     ((AND (M X (*N *E)) (ZEROP (CAR X))) (*MINUS (CADR X)))
     ((M X (*E *N)) (COND
          ((ZEROP (CADR X))(CAR X))
          ((MINUSP (CADR X)) (*PLUS (CAR X) (MINUS (CADR X))))
          ((CONS 'DIFF X)) ))
     ((M X (*E (DIFF *E *E))) (*PLUS (CAR X) (*DIFF (CADDADR X) (CADADR
          X))))
     ((CONS 'DIFF X)) )))
/*/*/
(SETQQ *MINUS (LAMBDA (X) (COND
     ((NUMBERP X)(MINUS X))
     ((M X (MINUS *E)) (CADR X))
     ((M X (DIFF *E *E)) (*DIFF (CADDR X) (CADR X)))
     ((LIST 'MINUS X)) )))
/*/*/
(SETQQ *TIMES (LAMBDA X (COND
     ((M X (*N *N)) (TIMES (CAR X)(CADR X)))
     ((EQUAL (CAR X) (CADR X)) (*EXPT (CAR X) 2))
     ((M X ((MINUS *E) (MINUS *E))) (*TIMES (CADAR X) (CADADR X)))
     ((M X ((MINUS *E) *E))(*MINUS (*TIMES (CADAR X) (CADR X))))
     ((M X (*E (MINUS *E))) (*MINUS (*TIMES (CAR X) (CADADR X))))
```

```
      ((M X (*E *N)) (*TIMES (CADR X) (CAR X)))
      ((M X (*N *E)) (COND
           ((ZEROP (CAR X)) 0)
           ((MINUSP (CAR X)) (*MINUS (*TIMES (MINUS (CAR X)) (CADR X))))
           ((ZEROP (SUB1 (CAR X)))(CADR X))
           ((M (CADR X) (TIMES *N *E))
                (*TIMES (TIMES (CAR X) (CADADR X)) (CADDADR X)))
           ((CONS 'TIMES X)) ))
      ((CONS 'TIMES X)) )))
/*/*/
(SETQQ *EXPT (LAMBDA X (COND
      ((M X (0 0)) (CONS 'EXPT X))
      ((M X (*E *N)) (COND
           ((ZEROP (CADR X)) 1)
           ((ZEROP (SUB1 (CADR X))) (CAR X))
           ((CONS 'EXPT X))))
      ((M X (*N *E)) (COND
           ((ZEROP (CAR X)) 0)
           ((ZEROP (SUB1 (CAR X))) 1)
           ((AND (NUMBERP (CADR X)) (ZEROP (ADD1 (CAR X)))) (COND
                ((ZEROP (REMAINDER (CADR X) 2)) 1)
                (-1) ))
           ((CONS 'EXPT X)) ))
      ((M X (E (LN *E))) (CADADR 0))
      ((M X ((EXPT *E *E) *E)) (*EXPT (CADAR X) (*TIMES (CADDAR X)
           (CADR X))))
      ((CONS 'EXPT X)) )))
/*/*/
(SETQQ *LN (LAMBDA (X) (COND
      ((M X 1) 0)
           ((M X (EXPT *E *E)) (*TIMES (CADDAR X) (*LN (CADAR X))))
      ((M X E) 1)
      ((LIST 'LN X)) )))
/*/*/
(SETQQ *SIN (LAMBDA (X) (COND
      ((M X(MINUS *E)) (*MINUS (*SIN (CADR X))))
      ((AND (NUMBERP X)(ZEROP X)) 0)
      ((LIST 'SIN X)) )))
/*/*/
(SETQQ *COS (LAMBDA (X)(COND
      ((M X (MINUS *E)) (*COS (CADR X)))
      ((AND (NUMBERP X) (ZEROP X)) 1)
      ((LIST 'COS X)) )))
/*/*/
(SETQQ OPS (
      (PLUS + . 2)
      (DIFF - . 1)
      (MINUS - . 0)
      (TIMES * . 3)
      (EXPT ** . 4)
      (SIN SIN . 5)
      (COS COS . 5)
      (LN LN . 5)
 ))
/*/*/
(SETQQ POLINF (LAMBDA (X N)(COND
  ((ATOM X)(LIST X))
  ((LESSP (CDDR (ASSOC (CAR X) OPS)) N) (LIST (POLINF X 0))  )
  ((NULL(CDDR X))(LIST(CADR(ASSOC(CAR X)OPS))(POLINF(CADR X)0)))
  ((APPEND(POLINF(CADR X)(CDDR(ASSOC(CAR X)OPS)))(LIST(CADR(ASSOC(CAR X)
          OPS)))(POLINF(CADDR X)(CDDR(ASSOC(CAR X)OPS)))))
  )))
/*/*/
(SETQQ IPOPS (
      (+ (PLUS 4 5) NIL)
      (- (DIFF 4 5) (MINUS 4 5))
      (* (TIMES 3 4) NIL)
      (** (EXPT 3 3) NIL)
      (SIN NIL (SIN 1 2))
      (COS NIL (COS 1 2))
      (LN NIL (LN 1 2))
      ))
```

```
/*/*/
(SETQQ INFPOL (LAMBDA (X F A Q) (PROG (N P)
     (AND X (NOT (SETQ N (ASSOC (CAR X) IPOPS))) Q (ERR NIL))
     (OR (SETQ P (COND ((NOT N) NIL) (Q (CADR N)) ((CADDR N)))) (NOT N)
          (ERR NIL))
     (RETURN (COND
          ((AND (NULL X) (NULL (CDR F))) (CAR A))
          ((OR (NULL X) (AND N (LESSP (CADDAR F) (CADDR P))))
           (INFPOL X (CDR F) (COND
                    ((CAAR F) (CONS (LIST (CADAR F) (CADR A) (CAR A))
                              (CDDR A)))
                    ((CONS (LIST (CADAR F) (CAR A)) (CDR A))))
              T))
          (N (INFPOL (CDR X) (CONS (CONS Q P) F) A NIL))
          ((ATOM (CAR X)) (INFPOL (CDR X) F (CONS (CAR X) A) T)).
          ((INFPOL (CDR X) F (CONS (INFPOL (CAR X) '((NIL NIL 100 0))
              NIL NIL) A) T)))) )))
/*/*/
(SETQQ D/DX (NLAMBDA (X) (POLINF (DERIV 'X (INFPOL X '((NIL NIL 100 0))
          NIL NIL)) 0)))
/*/*/
(PROG2 (PGSKP 3)(TEND))
(D/DX (X + 2))
(D/DX (X ** 2 - 3 * X + 15))
(D/DX (E ** X - E ** (- X)))
(D/DX ((SIN X) * (E ** X)))
(D/DX (X ** X))
(D/DX (E ** (- X ** 2)))
(QUIT)
```