```
(PRECOMP (SECTION SYS OCTAL)
  (LAP (PATCH (ORG)
    (ENTRY PDOK (LABEL PDOK))
    (ENTRY PDOK1 (LABEL PDOK1))
    GO (BUS (D. 1))
    (BUS (LABEL A))
    RECOV (BUS (D. 1))
    (BUS RECOV I)
    A (LDI 4 (R 7Q6))
    (LDX (INTCNT . SYS) 0 6)
    (LDX (DDSW . IC) 0 8)
    (LDX (INTTY . IO) 0 4)
    (BXH 2 (8 D) 0)
    (LDX BPO 0 8)
    (ARGS)
    (CALL START)
    (BUC (LABEL GO))
    (1Q6)
    (DITTO 4)
    PDOK1 (BXL (LABEL PDGONE) 8)
    PDOK (LDX (Z. 2Q1) 0 4)
    (BXL (LABEL P) 8)
    (XEC (LABEL P))
    (BUC 0 4)
    P (ATX -1 (4 L) 8)
    (LDX -1 (4 L) 4)
    (STX (PDADD . GC) 0 4) PDGONE (LDX PDGONE 0 4) (BUC 3 4) (END))
   (((DDSW . IC) OWN INTEGER VALUE)
    ((FREEZE . SYS) FLUID BOOLEAN VALUE)
    ((INTCNT . SYS) OWN INTEGER VALUE)
    ((FMCALL . SYS) OWN (FUNCTIONAL NOVALUE) VALUE)
    ((INTERRUPT . SYS) FUNCTION (FUNCTIONAL OCTAL) VALUE)
    ((INTTY . IO) FUNCTION (FUNCTIONAL NOVALUE) VALUE)
    (FPO OWN OCTAL VALUE)
    (FPP OWN OCTAL VALUE)
    (CHO OWN OCTAL VALUE)
    (TRO OWN OCTAL VALUE)
    (TRP OWN OCTAL VALUE)
    (BPO OWN OCTAL VALUE)
    (BPP OWN OCTAL VALUE)
    (ARO OWN OCTAL VALUE)
    (ARP OWN OCTAL VALUE)
    (LSP OWN OCTAL VALUE)
    (LSO OWN OCTAL VALUE)
    (TRL OWN OCTAL VALUE)
    (OBLIST OWN (ARRAY SYMBOL) VALUE)
    (OBLSIZ OWN INTEGER VALUE)
    (PDOUT OWN SYMBOL VALUE)
    ((PDADD . GC) OWN OCTAL VALUE)
    ((PDBUF . GC) OWN OCTAL VALUE)
    (START FUNCTION (FUNCTIONAL NOVALUE) VALUE)
    (RECOV FUNCTION (FUNCTIONAL NOVALUE) VALUE)
    (PDGONE FUNCTION (FUNCTIONAL NOVALUE) VALUE)
    (FNTRAP FUNCTION (FUNCTIONAL NOVALUE) VALUE)
    (FMTRAP FUNCTION (FUNCTIONAL NOVALUE) VALUE)) SYS)
  (LAP (PATCH (ORG)
    (ENTRY RETURN (LABEL RETURN))
    (ENTRY ROUT (LABEL ROUT))
    (ENTRY FLBIND (LABEL FLBIND))
    (ENTRY FLREST (LABEL FLREST))
    (ENTRY STZENT (LABEL STZENT))
    (ENTRY ONENT (LABEL ONENT))
    (ENTRY I2OENT (LABEL I2OENT))
   (ENTRY STBENT (LABEL STBENT))
   (ENTRY B48. (LABEL B48.))
   (ENTRY INTER (LABEL INTER))
   RETURN (LDX 0 8 4)
   (ATX -1 (4 L) 8)
   INTER (BPX 1 (6 D) 1)
   (LDX (FREEZE . SYS) I 3)
   (LDX 100 R 6)
   (BXE 1 (3 D) 1)
   (LDX -1 (4 L) 4)
   (STX (D. 1) L 4)
   (BPX (D. 1) 8)
   (LDX (INTERRUPT . SYS) 0 4)
   (BUC 2 4)
   ROUT (LDX 0 8 4)
   (ATX -1 (4 L) 8)
   (BUC 0 4)
   FLBIND (LDX (Z. 2Q1) 0 4)
   (ATX (Z. 8) 0 3)
   FLB1 (BAX (D. 1) 3 -1)
   (LDA 0 (4 L7.123 S))
   (ADD (Z. 8))
   (TST 0 4 6Q1)
   (LDA 0 A)
   (ECH 0 (4 I))
   (STF 0 3)
   (TST 0 4 51Q)
   (BAX (LABEL FLB1) 4 1)
   (BUC 1 4)
   FLREST (STP (LABEL FLR2) S567.7)
   (LDX -1 (7 L) 3)
   (ATX (Z. 8) 0 3)
   FLR1 (BAX (D. 1) 3 -1)
   (LDB 0 3)
   (STB 0 (7 I))
   (TST 0 7 51Q)
   (BAX (LABEL FLR1) 7 1)
   FLR2 (BUC)
   STZENT (STZ A.)
   (BUC 0 4)
   ONENT (LDA 1 (L567.7 R))
   (BUC 0 4)
   I20ENT (BNZM 0 4)
   (STZ A.)
   (BUC 0 4)
   STBENT (BOZP 0 4)
   (LDA 1 (L567.7 R)) (BUC 0 4) B48. (1071) (END)) NIL SYS)
(LAP (PATCH (ORG) (END))
 ((CONDERR FUNCTION (FUNCTIONAL NOVALUE) VALUE)
   (EQUAL. FUNCTION (FUNCTIONAL BOOLEAN SYMBOL SYMBOL) VALUE)
   (EQUALN. FUNCTION (FUNCTIONAL BOOLEAN SYMBOL SYMBOL) VALUE)
   (SYMABS FUNCTION (FUNCTIONAL SYMBOL SYMBOL) VALUE)
   (SYMSGN FUNCTION (FUNCTIONAL INTEGER SYMBOL) VALUE)
   (STIMS FUNCTION (FUNCTIONAL SYMBOL SYMBOL SYMBOL) VALUE)
   (STIMR FUNCTION (FUNCTIONAL REAL REAL SYMBOL) VALUE)
   (STIMI FUNCTION (FUNCTIONAL SYMBOL INTEGER SYMBOL) VALUE)
   (SPLUS FUNCTION (FUNCTIONAL SYMBOL SYMBOL SYMBOL) VALUE)
   (SPLUR FUNCTION (FUNCTIONAL REAL REAL SYMBOL) VALUE)
   (SPLUI FUNCTION (FUNCTIONAL SYMBOL INTEGER SYMBOL) VALUE)
   (SMINS FUNCTION (FUNCTIONAL SYMBOL SYMBOL SYMBOL) VALUE)
   (SMINI FUNCTION (FUNCTIONAL SYMBOL INTEGER SYMBOL) VALUE)
   (SMINR FUNCTION (FUNCTIONAL REAL REAL SYMBOL) VALUE)
   (MINSYM FUNCTION (FUNCTIONAL SYMBOL SYMBOL) VALUE)) SYS)
 (LAP (PATCH (ORG) (END))                                              0012700
  ((INT2OCT ROUTINE (FUNCTIONAL OCTAL INTEGER) VALUE)                  0012800
   (SYM2OCT FUNCTION (FUNCTIONAL OCTAL SYMBOL) VALUE)                  0012900
   (SYM2INT FUNCTION (FUNCTIONAL INTEGER SYMBOL) VALUE)                0013000
   (SYM2REAL FUNCTION (FUNCTIONAL REAL SYMBOL) VALUE)                  0013100
   (OCT2SYM FUNCTION (FUNCTIONAL SYMBOL OCTAL) VALUE)                  0013200
   (REAL2SYM FUNCTION (FUNCTIONAL SYMBOL REAL) VALUE)                  0013300
   (INT2SYM FUNCTION (FUNCTIONAL SYMBOL INTEGER) VALUE)                0013400
   (FORM2SYM FUNCTION (FUNCTIONAL SYMBOL FUNCTIONAL) VALUE)            0013500
   (SYM2FORM FUNCTION (FUNCTIONAL FUNCTIONAL SYMBOL) VALUE)            0013600
   (OCTROUND ROUTINE (FUNCTIONAL OCTAL REAL) VALUE)                    0013700
   (ROUND ROUTINE (FUNCTIONAL INTEGER REAL) VALUE)) LISP))             0013800
                                                                       0013900
****END OF FILE DETECTED
```

```
(APRIL.27.1200 (SECTION SYS OCTAL)
  MACRO1 (((LSHIFT (LAMBDA (L) (CONS (QUOTE SHIFT) (CDR L))))
    (RSHIFT (LAMBDA (L)
      (LIST (QUOTE SHIFT)
       (CADR L) (CONS (QUOTE MINUS) (CDDR L)))))))
  (ROUTINE FXRLB ((X OCTAL)))
  (DECLARE (OBLIST (ARRAY SYMBOL) OWN)
   (CHO OWN)
   (TRO OWN)
   (TRP OWN)
   (TRM OWN)
   (BPO OWN)
   (BPP OWN)
   (ARO OWN)
   (ARP OWN) (LSP OWN) (LSC OWN) (TRL OWN) (PDOUT SYMBOL OWN))
  (SECTION (GC SYS) OCTAL)
  INSTRUCTIONS (((MARKED (LAMBDA NIL (PROG NIL (ATTACH (LIST (QUOTE
         LDS) (CADR EXP) (QUOTE (I 40015Q2))))
       (SETQ VREG (QUOTE AC))
       (SETQ VTYPE (QUOTE BOOLEAN))
       (SETQ VCLASS (QUOTE ACTIVE)) (BLOTCH (QUOTE AC)))))
    (UNMARK (LAMBDA NIL (PROG NIL (ATTACH (LIST (QUOTE INS)
         (CADR EXP) (QUOTE (I 44Q5)))))))))
  MACRO1 (((LEFT (LAMBDA (X) (ITYBIT T 24 18 (CDR X))))
    (RIGHT (LAMBDA (X) (ITYBIT T 0 18 (CDR X))))
    (PREFIX (LAMBDA (X) (ITYBIT T 42 6 (CDR X))))
    (TAG (LAMBDA (X) (ITYBIT T 18 6 (CDR X))))
    (LEFTX (LAMBDA (X) (ITYBIT NIL 24 18 (CDR X))))
    (RIGHTX (LAMBDA (X) (ITYBIT NIL 0 18 (CDR X))))
    (ONEMOR (LAMBDA (X)
      (LIST (QUOTE I20.) (LIST (QUOTE PLUS) (CADR X) 1))))
    (ONELSS (LAMBDA (X)
      (LIST (QUOTE I20.) (LIST (QUOTE PLUS) (CADR X) -1))))
    (PLUS (LAMBDA (X)
      (LIST (QUOTE I20.)
       (CONS (QUOTE (PLUS . LISP)) (CDR X)))))))
  DEFINE (((ITYBIT (LAMBDA (W X Y Z)
      (CONS (QUOTE BIT)
       (CONS X (CONS Y (COND (W (CONS (CONS (QUOTE CORE) Z) NIL))
          (T Z)))))))))
  (DECLARE (GCC OWN)
   (A OWN)
   (B OWN)
   (C OWN)
   (D OWN)
   (X OWN)
   (Y OWN)
   (Z OWN)
   (BOOL BOOLEAN OWN)
   (XPDP OWN)
   (TEMP1 OWN)
   (TEMP2 OWN)
   (TEMP3 OWN)
   (TEMP4 OWN)
   (ARYORG OWN)
   (ARYCNT OWN)
   (BPSORG OWN)
   (FREREL OWN)
   (PDLREL OWN)
   (TRPCNT OWN)
   (BPMIN OWN)
   (PDADD OWN)
   (PDBUF OWN)
```

```
 (GCERR BOOLEAN OWN)
 (GC1 INTEGER OWN 200)
 (GC2 INTEGER OWN 200)
 (GC3 INTEGER OWN 2000)
 (GC4 REAL OWN 0.1999999999)
 (GC5 INTEGER OWN 500)
 (GC6 REAL OWN 0.5)
 (GC7 INTEGER OWN 20000) (GC8 INTEGER OWN 5000))
(ROUTINE ((ACPOOK . SYS) NOVALUE) ((X INTEGER)))
(ROUTINE ((MINR . LISP) REAL) ((A REAL) (B REAL)))
(ROUTINE ((MAXR . LISP) REAL) ((A REAL) (B REAL)))
(ROUTINE ((ENTIER . LISP) INTEGER) ((A REAL)))
(FUNCTION ((ERROR . LISP) SYMBOL) ((A SYMBOL)))
(FUNCTION ((RECLAIM . SYS) INTEGER)
 ((X INTEGER))
 (BLOCK ((S SYMBOL) (J INTEGER))
  (CODE (LDA (LABEL OFF)) (STF (ENTRY INTER)))
  (TRY S L (SET J (BLOCK NIL (IF (NULL PDOUT)
      (SET PDOUT (QUOTE G)))
     (SET GCC (PLUS 1 GCC))
     (SET (CORE (S20. OBLIST)) (WORDOR (CORE (S20. OBLIST)) 4Q7))
     (RETURN (DRIVER X))))))
  (CODE (LDA (LABEL ON)) (STF (ENTRY INTER)))
  (RETURN J)
  L (CODE (LDA (LABEL ON)) (STF (ENTRY INTER)))
  (EXIT S) OFF (CODE (BUC 1 4)) ON (CODE (BPX 1 (6 D) 1)))))
(FUNCTION (DRIVER INTEGER)
 (AA)
 (ORG NIL (BLOCK (ZZ)
   (CODE (STX XPDP 0 8))
   (SET BOOL FALSE)
   (SET ARYCNT 0)
   (SCNPDL MRKPDL)
   (MRKCHR)
   (MRKTSP)
   (PRUNVF)
   (PRUNCB)
   (PKTSP)
   (PKFREE)
   (SET ZZ (GPCALC AA))
   (RELARY)
   (FXTSP)
   (FXARY)
   (FXFREE)
   (RELBPS)
   (SET BOOL TRUE)
   (SCNPDL FIXIT1)
   (FXBPS)
   (SET TEMP1 BPO)
   (SET XPDP (PLUS XPDP -10))
   (IF (LS PDLREL 0)
    (BLOCK NIL (SET TEMP1 (PLUS TEMP1 PDLREL))
     (SET Z (PLUS (SET X XPDP) PDLREL))
     MO (SET (CORE Z) (CORE X))
     (SET Z (ONEMOR Z))
     (IF (NQ (SET X (ONEMOR X)) BPO) (GO MO))
     (CODE (ATX PDLREL 0 8))))
   (SET BPP (PACKIT 4Q15 BPO BPP TEMP1))
   (SET BPO TEMP1)
   A (SET TEMP1 (IF (LS ARYORG ARO) ARYORG ARO))
   (SET ARP (PACKIT 4Q7 ARO ARP TEMP1))
   (SET ARO TEMP1)
   (IF (GR ARYORG ARO)
    (BLOCK NIL (SET ARP (SET Z (PLUS (SET X ARP)
       (MINUS ARO) ARYORG)))
     (GO L)
     M (SET (CORE Z) (CORE X))
     L (SET Z (ONELESS Z))
     (IF (GQ (SET X (ONELESS X)) ARO) (GO M)) (SET ARO ARYORG)))
   (IF (GR PDLREL 0)
    (BLOCK NIL (SET BPP (SET Z (PLUS (SET X BPP) PDLREL)))
     (GO L1)
     M1 (SET (CORE Z) (CORE X))
     L1 (SET Z (ONELESS Z))
     (IF (GQ (SET X (ONELESS X)) XPDP) (GO M1))
     (SET BPC (PLUS BPC PDLREL)) (CODE (ATX PDLREL 0 8))))
   (IF (EQ PDOUT (QUOTE B)) (ADPDCK PDBUF))
   (SET PDOUT NIL)
   (IF GCERR (ERROR (QUOTE (OUT OF STORAGE)))
    (RETURN ZZ)))))
MACRO1 (((PLUS (LAMBDA (X)
  (CONS (QUOTE (PLUS . LISP)) (CDR X))))))
(FUNCTION (GPCALC INTEGER)
 ((AA OCTAL))
 (BLOCK ((A1IU INTEGER (PLUS BPO PDADD TRP (MINUS TRO)
     (MINUS XPDP)))
   (A2IU INTEGER (PLUS BPP BPMIN (MINUS BPO)))
   (A3IU INTEGER (PLUS LSO (BLOCK ((I INTEGER ARO)
        (ARYCNT INTEGER 0))
      (FOR I (RESET I (PLUS I (BIT 24 18 (CORE I))))
       (WHILE (NQ I ARP))
       (UNLESS (EQ (WORDAND (CORE I) 4Q7) 0))
       (SET ARYCNT (PLUS ARYCNT (BIT 24 18 (CORE I)))))
      (RETURN ARYCNT)) AA (MINUS LSP))) (ZZ INTEGER))
  (SET GCERR FALSE)
  BACK (BLOCK ((N INTEGER (DIFFERENCE PDBUF (SET ZZ (PLUS LSO
       (MINUS TRO) (MINUS A1IU) (MINUS A2IU) (MINUS A3IU))))))
   (IF (AND GCERR (GR N 0))
    (GO PP)
    (AND (NOT GCERR) (LS (MINUS N) GC8))
    (BLOCK ((BPKEEP INTEGER BPMIN)
      (OK BOOLEAN (UNLBPS (PLUS N GC8 ARO (MINUS BPP)
         (MINUS BPMIN)))))
     (SET A2IU (PLUS A2IU BPMIN (MINUS BPKEEP)))
     (IF (NOT OK) (SET GCERR TRUE) (SET GCERR FALSE)) (GO BACK))
    (SET GCERR FALSE)))
  (BLOCK ((PDEX INTEGER (PLUS BPC (MINUS TRO) (MINUS A1IU)))
    (BPEX INTEGER (PLUS ARO (MINUS BPO) (MINUS A2IU)))
    (LAEX INTEGER (PLUS LSO (MINUS ARO) (MINUS A3IU))))
   (IF (AND (GR BPEX GC1) (GR PDEX GC2) (GR LAEX GC3)) (GO PP))
   LL (BLOCK ((ZE INTEGER (DIFFERENCE ZZ PDBUF)) (ST INTEGER))
    (SET ST (MIN (TIMES GC4 ZE) GC5))
    (SET PDLREL (PLUS ST PDBUF))
    (SET ZE (DIFFERENCE ZE ST))
    (SET ARYORG (PLUS PDLREL TRO A1IU A2IU (TIMES GC6 ZE)))
    (SET PDLREL (DIFFERENCE PDLREL PDEX))) (GO P))
  PP (SET PDLREL 0)
  (SET ARYORG ARO)
  P (SET BPMIN 0)
  (SET GC7 A2IU)
  (SET BPSORG (PLUS BPC PDLREL 1)) (RETURN ZZ)))
MACRO1 (((PLUS (LAMBDA (X)
    (LIST (QUOTE I20.) (CONS (QUOTE (PLUS . LISP)) (CDR X)))))))
(FUNCTION (FIXIT1 NOVALUE) ((J OCTAL)) (ORG NIL (FIXIT J)))
(FUNCTION (SCNPDL NOVALUE)
 ((FN (FUNCTIONAL NOVALUE OCTAL)))
 (ORG NIL (BLOCK (CURONE CURLNK (CURPNT XPDP) NXTONE BITMAP COUNT)
   MORE (SET CURLNK (RIGHT (SET CURONE CURPNT)))
   (SET NXTONE (PLUS CURPNT (LEFT (ONELSS CURLNK))))
   TO (SET BITMAP (LSHIFT (CORE CURLNK)
     (SET COUNT (PLUS CURPNT (MINUS NXTONE)
       (LEFT (ONELSS CURLNK))))))
   (SET COUNT (PLUS 24 (MINUS COUNT)))
   XTO (IF (EQUAL (SET CURPNT (ONEMOR CURPNT)) NXTONE)
    (BLOCK NIL (SET CURLNK (RIGHT CURLNK)) (GO XT1))
    (LS BITMAP 0) (FN CURPNT))
   (SET BITMAP (LSHIFT BITMAP 1))
   (IF (NQ (SET COUNT (ONELSS COUNT)) 0) (GO XTO))
   (IF (EQ (TAG (SET CURLNK (RIGHT CURLNK))) 0) (GO TO))
   T1 (SET CURPNT NXTONE)
   XT1 (IF (AND BOOL (NQ (SET TEMP1 (RIGHT CURLNK)) 0))
    (SET (CORE CURONE)
     (PLUS (CORE CURONE)
      (IF (EQUAL (WORDAND 1Q15 (SET TEMP1 (CORE TEMP1))) 0)
       (RIGHTX TEMP1) (LEFTX TEMP1)) (MINUS CURLNK) -1)))
   (IF (LS CURPNT BPC) (GO MORE)))))
(FUNCTION (MRKPDL NOVALUE)
 (X)
 (BLOCK NIL (IF (GQ (SET X (CORE X)) 1Q6)
   (BLOCK NIL (MARKIT (LEFTX X))
    (IF (NQ (BIT 18 6 X) 0) (MARKIT (ONEMOR (RIGHTX X)))))
   (BLOCK NIL (IF (GQ X ARO)
     (GO FIX)
     (LS X TRC)
     (GO OUT)
     (LS X TRP)
     (IF (EQUAL (PREFIX (ONEMOR X)) 12Q) (SET X (ONEMOR X)))
     (GO OUT)) FIX (MARKIT X) OUT))))
(FUNCTION (MARKIT NOVALUE)
 ((X OCTAL))
 (BLOCK NIL MORE (IF (LS X ARO)
   (IF (AND (LQ TRC X) (LS X TRP) (MARKEM X))
    (BLOCK NIL (SET X (MRKTRP X)) (GO MORE)))
   (AND (LS X LSC) (MARKEM X))
   (IF (LS X LSP)
    (MRKARY X)
    (BLOCK NIL (MARKIT (BIT 24 18 (CORE X)))
     (SET X (BIT 0 18 (CORE X))) (GO MORE)))) OUT))
(FUNCTION MRKTRP (X)
 (BLOCK NIL (SET TEMP1 (CORE (ONELSS X)))
  (CASE (PLUS (PREFIX X) -6)
   (GO T7) (GO T10) (GO T11) (GO T12) (GO T13))
  T7 (IF (NQ (WORDAND (CORE X) 4Q6) 0)
   (SET (CORE TEMP1) (WORDOR (CORE TEMP1) 4Q7)))
  (RETURN (RIGHT X))
  T10 (RETURN (RIGHTX TEMP1))
  T12 (CASE (ONEMOR (RSHIFT (SET TEMP2 (CHEKUP (CORE (ONEMOR X))))
     3)) (GO X0) (GO X11) (GO XX) (GO X11) (GO X4) (GO X11))
  X0 (IF (EQUAL TEMP2 0) (LABEL XX (MARKIT TEMP1)))
  (GO T13)
  X4 (IF (EQUAL (BIT 42 6 TEMP1) 0)
   (MARKIT (LEFTX TEMP1))
   (BLOCK NIL (IF (EQUAL (WORDAND TEMP1 1Q15) 0)
     (MARKIT (LEFTX TEMP1))) (GO T13)))
  X5 (IF (MARKEM (SET TEMP1 (ONEMOR (RIGHT (ONELSS X)))))
   (MARKIT (MRKTRP TEMP1)))
  (GO T13)
  T11 (IF (EQUAL (BIT 42 6 (SET TEMP2 (CORE (ONEMOR X)))) 1)
   (MARKEM (LEFTX TEMP2))
```

```
    (EQ (BIT 42 6 TEMP2) 2) (MARKIT (LEFTX TEMP2)))
   X11 (IF (NQ (SET TEMP2 (LEFTX TEMP1)) 0)
    (IF (MARKEM TEMP2) (MRKARY TEMP2))
    (LS TEMP1 TRP)
    (IF (MARKEM (SET TEMP1 (ONEMOR TEMP1)))
     (MARKIT (MRKTRP TEMP1)))) T13 (RETURN (LEFT X))))
(FUNCTION CHEKUP (X)
 (BLOCK NIL (IF (GR (SET TEMP2 (BIT 42 6 X)) 2) (RETURN TEMP2))
  (CASE (ONEMOR TEMP2) (GO T0) (GO T1) (GO T2))
  T0 (RETURN (BIT 18 6 X))
  T1 (SET (CORE TEMP2) (WORDOR (CORE (SET TEMP2 (LEFTX X))) 4Q7))
  (RETURN (PREFIX (ONEMOR TEMP2)))
  T2 (BLOCK ((Y TEMP1))
   (MARKIT (MRKTRP (LEFTX X)))
   (SET TEMP1 Y) (RETURN (CHEKUP (CORE (ONEMOR (LEFTX X))))))))
(FUNCTION (MRKARY NOVALUE)
 (X)
 (BLOCK (Y Z)
  (SET ARYCNT (PLUS ARYCNT (LEFT X)))
  (SET Y (ONEMOR X))
  (SET Z (PLUS X (LEFT X)))
  (CASE (ONEMOR (WORDAND (PREFIX X) 7Q))
   (GO T0) (GO TX) (GO TX) (GO TX) (GO TX) (GO T5) (GO TX))
  T0 (FOR Y (RESET Y (PLUS Y 1))
   (WHILE (NQ Y Z)) (MARKIT (RIGHT Y)))
  (GO TX)
  T5 (FOR Y (RESET Y (PLUS Y 1))
   (WHILE (NQ Y Z))
   (BLOCK NIL (MARKIT (LEFT Y)) (MARKIT (ONEMOR (RIGHT Y))))) TX))
(ROUTINE (MARKEM BOOLEAN)
 ((X OCTAL))
 (IF (NOT (MARKED X))
  (O2B. (SET (CORE X) (WORDOR (CORE X) 4Q7))) FALSE))
(FUNCTION (MRKCHR NOVALUE)
 NIL (BLOCK ((X CHO))
  (FOR X (RESET X (PLUS X 1))
   (WHILE (NQ X TRO)) (MARKIT (RIGHT X)))))
(FUNCTION (MRKTSP NOVALUE)
 NIL (BLOCK ((X (ONEMOR TRO)))
  (FOR X (RESET X (PLUS X 3))
   (WHILE (LS X TRP))
   (IF (NOT (MARKED X))
    (IF (NQ (RIGHT X) 0)
     (LABEL GETUM (BLOCK NIL (SET (CORE X) (WORDOR (CORE X) 4Q7))
       (MARKIT (MRKTRP X))))
     (BLOCK NIL (CASE (PLUS (PREFIX X) -6)
       (GO T7) (GO TS) (GO TX) (GO TX) (GO TS))
      T7 (IF (NQ (LEFT (ONEMOR X)) 0) (GO GETUM))
      (GO TS)
      TX (IF (NQ (WORDAND (CORE X) 1Q7) 0) (GO GETUM)) TS))))))
(FUNCTION (PRUNVF NOVALUE)
 NIL (BLOCK NIL (FOR A (RESET CHO (PLUS A 1))
   (WHILE (NQ A TRO)) (IF (NQ A (LEFT A)) (PRUNIT A)))
  (FOR A (RESET (PLUS TRO 1) (PLUS A 3))
   (WHILE (LS A TRP))
   (IF (EQUAL (PREFIX A) 7Q)
    (IF (MARKED A)
     (IF (NQ A (LEFT A)) (PRUNIT A))
     (IF (NOT (OR (EQUAL A (LEFT A)) (EQUAL A (PRUNIT A))))
      (MARKIT A)))))))
(ROUTINE PRUNIT (X)
 (BLOCK NIL (SET TEMP1 (LEFT X))
  (SET TEMP2 (CODE (LDA TEMP1 (R L567.7))))
```

```
    (SET TEMP3 TEMP1)                                                   0031600
    AGAIN (SET TEMP4 (ONEMOR TEMP3))                                    0031700
    (IF (MARKED TEMP3)                                                  0031800
     (SET TEMP2 TEMP4) (SET (RIGHT TEMP2) (RIGHT TEMP4)))               0031900
    (IF (NQ X (SET TEMP3 (RIGHT TEMP4))) (GO AGAIN))                    0032000
    (RETURN (SET (LEFT X) TEMP1))))                                     0032100
  (ROUTINE (PRUNOB NOVALUE)                                             0032200
   NIL (BLOCK NIL (SET A (ONEMOR (S2C. OBLIST)))                        0032300
    (SET B (PLUS A 125))                                                0032400
    (FOR A (RESET A (PLUS A 1))                                         0032500
     (WHILE (NQ A B))                                                   0032600
     (BLOCK NIL (SET Z (CORE (SET C A)))                                0032700
      MORE (IF (NQ (SET X Z) 0)                                         0032800
       (BLOCK NIL (SET Z (RIGHT (SET Y (ONEMOR X))))                    0032900
        (IF (MARKED X) (SET C Y) (SET (RIGHT C) Z)) (GO MORE)))))))     0033000
  (ROUTINE (PKTSP NOVALUE)                                              0033100
   NIL (BLOCK ((X OCTAL (PLUS TRP -2)))                                 0033200
    (FOR X (STEP X -3)                                                  0033300
     (WHILE (NOT (MARKED X)))                                           0033400
     (BLOCK NIL (SET TRP (PLUS TRP -3)) (ADPUOK -3)))                   0033500
    (SET TRPCNT 0)                                                      0033600
    (SET TRL 0)                                                         0033700
    (FOR X (STEP X -3 LS TRO)                                           0033800
     (IF (MARKED X)                                                     0033900
      (UNMARK X)                                                        0034000
      (BLOCK NIL (SET (CORE (PLUS X -1)) 0)                             0034100
       (SET (CORE X) 13Q14)                                             0034200
       (SET (CORE (PLUS X 1)) TRL)                                      0034300
       (SET TRL X) (SET TRPCNT (PLUS TRPCNT 1)))))))                    0034400
  (ROUTINE (PKFREE NOVALUE)                                             0034500
   NIL (BLOCK NIL (FOR X (RESET (ONELSS LSO) (PLUS X -1))               0034600
     (WHILE (GQ X LSP))                                                 0034700
     (BLOCK NIL (IF (NOT (MARKED X))                                    0034800
       (BLOCK NIL (FOR LSP (RESET LSP (PLUS LSP 1))                     0034900
         (WHILE (GQ X LSP))                                             0035000
         (IF (MARKED (LSP . SYS))                                       0035100
          (BLOCK NIL (UNMARK (LSP . SYS))                               0035200
           (SET (CORE X) (CORE LSP))                                    0035300
           (SET (CORE LSP) X) (SET LSP (ONEMOR LSP)) (GO MORE))))       0035400
        (GO OUT)) (UNMARK X)) MORE)) OUT))                              0035500
  (ROUTINE (RELBPS NOVALUE)                                             0035600
   NIL (ORG NIL (BLOCK NIL (SET X BPO)                                  0035700
     MORE (SET Y (LEFT X))                                              0035800
     (IF (NQ (WORDAND (CORE X) 4Q15) 0)                                 0035900
      (BLOCK NIL (IF (NQ (WORDAND (CORE (SET Z (RIGHT X))) 1Q15) 0)     0036000
        (SET (LEFT Z) BPSORG) (SET (RIGHT Z) BPSORG))                   0036100
       (SET BPSORG (PLUS BPSORG Y))))                                   0036200
     (IF (NQ (SET X (PLUS X Y)) BPP) (GO MORE)))))                      0036300
  (ROUTINE (RELARY NOVALUE)                                             0036400
   NIL (BLOCK NIL (SET X ARO)                                           0036500
    (SET Z ARYORG)                                                      0036600
    MORE (SET Y (LEFT X))                                               0036700
    (IF (MARKED X) (BLOCK NIL (SET (RIGHT X) Z) (SET Z (PLUS Z Y))))    0036800
    (IF (NQ (SET X (PLUS X Y)) ARP) (GO MORE))))                        0036900
  (ROUTINE (FIXIT NOVALUE)                                              0037000
   (X)                                                                  0037100
   (ORG NIL (IF (LS (SET TEMP1 (CORE X)) 1Q6)                           0037200
     (SET (CORE X) (RELOC TEMP1))                                       0037300
     (NQ (BIT 18 6 TEMP1) 0)                                            0037400
     (SET (LEFT X) (RELOC (LEFTX TEMP1)))                               0037500
     (BLOCK NIL (SET TEMP1 (PLUS (RIGHTX TEMP1)                         0037600
        (MINUS (SET TEMP2 (LEFTX TEMP1)))))                             0037700
      (SET (LEFTX TEMP2) (SET TEMP2 (RELOC TEMP2)))                     0037800
     (SET (CORE X) (PLUS TEMP1 TEMP2)))))
  (ROUTINE RELOC (X)
   (ORG NIL (IF (OR (LS X TRP) (GQ X LSO))
     X (GQ X ARP)
     (PLUS (IF (GQ X LSP) X (CORE X)) FREREL)
     (GQ X ARO) (RIGHT X) (LS X BPO) (PLUS X POLREL) X)))
  (ROUTINE (FXTSP NOVALUE)
   NIL (BLOCK NIL (FOR X (RESET CHC (PLUS X 1))
     (WHILE (NQ X TRO)) (SET (RIGHT X) (RELOC (RIGHT X))))
    (FOR X (RESET (PLUS TRC 1) (PLUS X 3))
     (WHILE (LS X TRP))
     (BLOCK NIL (SET Y (ONELSS X))
      (CASE (PLUS (PREFIX X) -6)
       (GO T7) (GO T10) (GO T11) (GO T12) (GO T13))
      T7 (IF (NQ (WORDAND (CORE X) 4Q6) 0)
       (SET (RIGHT Y) (RIGHT (RIGHT Y))))
      (SET Y X)
      T10 (SET (RIGHT Y) (RELOC (RIGHT Y)))
      (GO T13)
      T12 (IF (EQUAL (SET TEMP1 (FXTYPE (ONEMOR X))) 45Q)
       (BLOCK NIL (IF (EQUAL (WORDAND (SET TEMP1 (CORE Y)) 1Q15) 0)
         (SET (LEFT Y) (RELOC (LEFTX TEMP1)))) (GO T13))
       (GR TEMP1 4) (GO X12) (EQUAL TEMP1 0) (GO T10))
      (GO T13)
      T11 (IF (EQUAL (PREFIX (SET TEMP1 (ONEMOR X))) 1)
       (SET (LEFT TEMP1) (RIGHT (LEFT TEMP1))))
      X12 (FIXIT Y) T13))))
  (ROUTINE FXTYPE (X)
   (BLOCK NIL (IF (GR (SET TEMP1 (PREFIX X)) 2) (GO TX))
    (CASE (ONEMOR TEMP1) (GO T0) (GO T1) (GO T2))
    T2 (RETURN (FXTYPE (ONEMOR (LEFT X))))
    T0 (RETURN (TAG X))
    T1 (SET TEMP1 (PREFIX (ONEMOR (SET TEMP2 (LEFT X)))))
    (SET (LEFT X) (RIGHT TEMP2)) TX (RETURN TEMP1)))
  (ROUTINE (FXARY NOVALUE)
   NIL (BLOCK NIL (SET X ARO)
    MORE (SET Y (LEFT X))
    (IF (MARKED X)
     (BLOCK NIL (CASE (ONEMOR (WORDAND (PREFIX X) 7))
       (GO OK) (GO TX) (GO TX) (GO TX) (GO TX) (GO OK) (GO TX))
      OK (SET B (PLUS X Y))
      (FOR A (RESET (PLUS X 1) (PLUS A 1))
       (WHILE (NQ A B)) (FIXIT A)) TX))
    (IF (NQ (SET X (PLUS X Y)) ARP) (GO MORE))))
  (ROUTINE (FXFREE NOVALUE)
   NIL (BLOCK NIL (FOR X (RESET LSP (PLUS X 1))
     (WHILE (NQ X LSO))
     (BLOCK NIL (SET (LEFT X) (RELOC (LEFT X)))
      (SET (RIGHT X) (RELOC (RIGHT X)))))))
  (ROUTINE (FXBPS NOVALUE)
   NIL (ORG NIL (BLOCK NIL (SET TEMP1 BPO)
     MORE (SET TEMP2 (LEFT TEMP1))
     (IF (NQ (WORDAND (CORE TEMP1) 4Q15) 0) (FXFN TEMP1 FALSE))
     (IF (NQ (SET TEMP1 (PLUS TEMP1 TEMP2)) BPP) (GO MORE)))))
  (ROUTINE (FXFN NOVALUE)
   ((Y OCTAL) (BOOL BOOLEAN))
   (ORG NIL (BLOCK NIL (SET Z (PLUS Y (LEFT Y) -1))
     (SET X (RIGHT Y))
     (SET C (PLUS (IF (NQ (WORDAND (CORE X) 1Q15) 0)
        (LEFT X) (RIGHT X)) (MINUS Y) -1))
     MORE (SET B (CORE Z))
     (FOR A (RESET 0 (PLUS A 1))
      (WHILE (NQ A 24))
      (BLOCK NIL (IF (GQ Y Z)
        (GO OUT)
        (NQ (WORDAND B 4Q15) 0)
        (IF (LS (SET D (LEFT Y)) TRP)
         (IF BOOL (FXRUB D))
         (IF (NOT BOOL) (SET (LEFT Y) (PLUS C D)))))
       (IF (NQ (WORDAND B 2Q15) 0)
        (IF (LS (SET D (RIGHT Y)) TRP)
         (IF BOOL (FXRUB D))
         (IF (NOT BOOL) (SET (RIGHT Y) (PLUS C D)))))
       (SET Y (ONEMOR Y)) (SET B (LSHIFT B 2))))
     (SET Z (ONELSS Z)) (GO MORE) OUT)))
  (ROUTINE PACKIT (W X Y Z)
   (ORG NIL (BLOCK NIL (SET BOOL (NQ X Z))
     AGAIN (SET A (PLUS X (LEFT X)))
     (IF (NQ (WORDAND (CORE X) W) 0)
      (IF BOOL (BLOCK NIL (UNMARK X)
        MORE (SET (CORE Z) (CORE X))
        (SET Z (ONEMOR Z)) (IF (NQ (SET X (ONEMOR X)) A) (GO MORE)))
       (BLOCK NIL (UNMARK X) (SET X (SET Z A))))
      (SET BOOL (D2B. (SET X A))))
     (IF (NQ X Y) (GO AGAIN)) (RETURN Z)))))
****END OF FILE DETECTED
```

```
(INDEXD (SECTION LISP SYMBOL)
  (FUNCTION (ERROR SYMBOL) ((M SYMBOL)))
  (FUNCTION (EXIT SYMBOL) ((M SYMBOL)))
  (FUNCTION (LAP SYMBOL) ((X SYMBOL) (Y SYMBOL) (Z SYMBOL)))
  (DECLARE TTY. DISC. TAPE. CORE. CRT. (SKIPR. INTEGER OWN)
   (SKIPF. INTEGER OWN)
   (WEOF. INTEGER OWN)
   (WEOT. INTEGER OWN)
   (REWIND. INTEGER OWN)
   (BACKR. INTEGER OWN) (BACKF. INTEGER OWN) (KEY. INTEGER OWN))
  (FUNCTION (PRETTYP SYMBOL) ((S SYMBOL)))
  (FUNCTION (OPEN SYMBOL) ((FN SYMBOL) (DL SYMBOL)))
  (FUNCTION (SHUT SYMBOL) ((FN SYMBOL) (DL SYMBOL)))
  (FUNCTION (POSITION SYMBOL) ((FN SYMBOL) (DL INTEGER)))
  (FUNCTION (INPUT SYMBOL) ((X SYMBOL)))
  (FUNCTION (OUTPUT SYMBOL) ((X SYMBOL)))
  (FUNCTION (READ SYMBOL) NIL)
  (FUNCTION (PRINT SYMBOL) ((X SYMBOL)))
  (FUNCTION (PRIN SYMBOL) ((X SYMBOL)))
  (FUNCTION (PRIN0 SYMBOL) ((X SYMBOL)))
  (FUNCTION (PRINATOM SYMBOL) ((X SYMBOL)))
  (FUNCTION (PRINTOKEN SYMBOL) ((X SYMBOL)) (PRINATOM X))
  (FUNCTION (PRINSTRING SYMBOL) ((X SYMBOL)))
  (FUNCTION (SYMPRINT SYMBOL) ((X SYMBOL)))
  (FUNCTION (SYMPRIN SYMBOL) ((X SYMBOL)))
  (FUNCTION (PRINCH SYMBOL) ((X SYMBOL)))
  (FUNCTION (READCH SYMBOL) NIL)
  (FUNCTION (PRINWORD OCTAL) ((X OCTAL)))
  (FUNCTION (READWORD OCTAL) NIL)
  (FUNCTION (ENDIN NOVALUE) NIL)
  (FUNCTION (ENDINR NOVALUE) NIL)
  (FUNCTION (ENDOUT NOVALUE) NIL)
  (FUNCTION (ENDOUTR NOVALUE) NIL)
  (FUNCTION (ENDINP NOVALUE) NIL)
  (FUNCTION (ENDOUTP NOVALUE) NIL)
  (FUNCTION (NOP NOVALUE) NIL)
  (FUNCTION (NILF SYMBOL) NIL)
  (ROUTINE (CLEAR NOVALUE) ((FN SYMBOL)))
  (FUNCTION (ARREAD SYMBOL) NIL)
  (ROUTINE GETCHAR ((A (ARRAY OCTAL)) (N INTEGER)))
  (SECTION SYS SYMBOL)
  (FUNCTION MESSAGE (A))
  (FUNCTION (FNTRAP NOVALUE) NIL)
  (FUNCTION (FMTRAP NOVALUE) NIL)
  (FUNCTION (RECLAIM INTEGER) ((I INTEGER)))
  (FUNCTION ((FVLIS1 . COMPIL) SYMBOL) (ARG))
  (SECTION IO SYMBOL)
  (DECLARE (XXSAVE SYMBOL FLUID LOC))
  (DECLARE (XXFUNC (FUNCTIONAL SYMBOL) FLUID LOC)))
(MACRO1 (SECTION SYS SYMBOL)
  DEFINE (((WDPART (LAMBDA (A B S M)
      (LIST (QUOTE BIT)
       A B (COND (M (CADR S))
        (T (CONS (QUOTE CORE) (CDR S)))))))))
  MACRO1 (((ARSIZE (LAMBDA (S) (CONS (QUOTE LEFTAD) (CDR S))))
    (PREFIX (LAMBDA (S) (WDPART 42 6 S NIL)))
    (PREFIM (LAMBDA (S) (WDPART 42 6 S T)))
    (TAG (LAMBDA (S) (WDPART 18 6 S NIL)))
    (TAGIM (LAMBDA (S) (WDPART 18 6 S T)))
    (LEFTAD (LAMBDA (S) (WDPART 24 18 S NIL)))
    (LEFTIM (LAMBDA (S) (WDPART 24 18 S T)))
    (RGHTAD (LAMBDA (S) (WDPART 0 18 S NIL)))
    (RGHTIM (LAMBDA (S) (WDPART 0 18 S T)))
    (WORD1 (LAMBDA (S)
      (LIST (QUOTE CORE)
       (LIST (QUOTE I20.) (LIST (QUOTE PLUS) (CADR S) -1)))))
    (WORD2 (LAMBDA (S) (CONS (QUOTE CORE) (CDR S))))
    (WORD3 (LAMBDA (S)
      (LIST (QUOTE CORE)
       (LIST (QUOTE I20.) (LIST (QUOTE PLUS) (CADR S) 1)))))
    (LINK (LAMBDA (S)
      (LIST (QUOTE RGHTIM) (CONS (QUOTE WORD3) (CDR S)))))
    (PNAME (LAMBDA (S)
      (LIST (QUOTE RGHTIM) (CONS (QUOTE WORD1) (CDR S)))))
    (CORES (LAMBDA (S)
      (LIST (QUOTE CORE) (CONS (QUOTE S20.) (CDR S)))))
    (CHAIN (LAMBDA (S)
      (LIST (QUOTE LEFTIM) (CONS (QUOTE WORD2) (CDR S)))))
    (CHAINS (LAMBDA (S)
      (LIST (QUOTE LEFTIM) (CONS (QUOTE CORES) (CDR S)))))
    (VCOUNT (LAMBDA (S) (WDPART 0 18 S NIL)))
    (TCODE (LAMBDA (S)
      (WDPART 18 30 (LIST (CAR S) (CONS (QUOTE WORD3) (CDR S))) T)))
    (TCODES (LAMBDA (S)
      (LIST (QUOTE TCODE) (CONS (QUOTE S20.) (CDR S)))))
    (XMFLAG (LAMBDA (S) (WDPART 22 1 S NIL))))))
(BOUNDS (SECTION SYS OCTAL)
  (DECLARE (TRL OCTAL OWN)
   (FPO OCTAL OWN)
   (FPP OCTAL OWN)
   (CHO OCTAL OWN)
   (TRO OCTAL OWN)
   (TRP OCTAL OWN)
   (TRM OCTAL OWN)
   (BPO OCTAL OWN)
   (BPP OCTAL OWN)
   (ARO OCTAL OWN)
   (ARP OCTAL OWN)
   (LSP OCTAL OWN) (LSC OCTAL OWN) (DELTRM OCTAL OWN)))
(PREDS (SECTION (LISP SYS) SYMBOL)
  (ROUTINE (NORMSP BOOLEAN)
   ((S SYMBOL)) (EQ (BIT 21 1 (WORD2 (S20. S))) 0Q))
  (ROUTINE (ARRAYP BOOLEAN)
   ((S SYMBOL))
   (AND ((ARSPAC . SYS) S) (EQ (BIT 3 3 (PREFIX (S20. S))) 2)))
  (ROUTINE (NUMBP BOOLEAN)
   ((S SYMBOL))
   (OR (GQ (S20. S) 2Q5)
    (AND ((ARSPAC . SYS) S)
     (LS (PREFIX (S20. S)) 5) (GQ (PREFIX (S20. S)) 2))))
  (ROUTINE (OCTALP BOOLEAN)
   ((S SYMBOL))
   (OR (AND (GQ (S20. S) 2Q5) (LS (S20. S) 4Q5))
    (AND ((ARSPAC . SYS) S) (EQ (PREFIX (S20. S)) 2))))
  (ROUTINE (INTP BOOLEAN)
   ((S SYMBOL))
   (OR (GQ (S20. S) 4Q5)
    (AND ((ARSPAC . SYS) S) (EQ (PREFIX (S20. S)) 3))))
  (ROUTINE (REALP BOOLEAN)
   ((S SYMBOL)) (AND ((ARSPAC . SYS) S) (EQ (PREFIX (S20. S)) 4)))
  (ROUTINE (FORMALP BOOLEAN)
   ((S SYMBOL)) (AND ((ARSPAC . SYS) S) (EQ (PREFIX (S20. S)) 5)))
  (ROUTINE (OWNP BOOLEAN)
   ((S SYMBOL)) (AND ((TRSPAC . SYS) S) (EQ (PREFIX (S20. S)) 12Q)))
  (ROUTINE (FLUIDP BOOLEAN)
   ((S SYMBOL)) (AND ((TRSPAC . SYS) S) (EQ (PREFIX (S20. S)) 11Q)))
  (ROUTINE (FIXP BOOLEAN)
   ((S SYMBOL))
   (OR (GQ (S20. S) 2Q5)
    (AND ((ARSPAC . SYS) S)
     (OR (EQ (PREFIX (S20. S)) 2) (EQ (PREFIX (S20. S)) 3)))))
  (ROUTINE (IDP BOOLEAN)
   ((S SYMBOL))
   (OR (AND ((TRSPAC . SYS) S) (EQ (PREFIX (S20. S)) 7)) (CHARP S)))
  (ROUTINE (STRINGP BOOLEAN)
   ((S SYMBOL)) (AND ((ARSPAC . SYS) S) (EQ (PREFIX (S20. S)) 6)))
  (ROUTINE (LISTP BOOLEAN)
   ((X SYMBOL))
   (BLOCK NIL LOOP (IF (ATOM X) (RETURN (NULL X)))
    (SET X (CDR X)) (GO LOOP)))
  (ROUTINE (ATOM BOOLEAN)
   ((S SYMBOL)) (OR (LS (S20. S) LSP) (GQ (S20. S) LSO)))
  (ROUTINE (GENIDP BOOLEAN)
   ((S SYMBOL)) (AND (IDP S) (EQ (BIT 18 1 (WORD2 (S20. S))) 1)))
  (ROUTINE (CHARP BOOLEAN)
   ((S SYMBOL)) (AND (LS (S20. S) TRO) (GQ (S20. S) CHO)))
  (ROUTINE (BOOLP BOOLEAN) ((S SYMBOL)) (LS (S20. S) 2)))
(EQUALS (SECTION SYS SYMBOL)
  (FUNCTION (EQUAL. BOOLEAN)
   ((A SYMBOL) (B SYMBOL)) (EQL. A B TRUE))
  (FUNCTION (EQUALN. BOOLEAN)
   ((A SYMBOL) (B SYMBOL)) (EQL. A B NIL))
  (FUNCTION (EQL. BOOLEAN)
   ((A (ARRAY OCTAL)) (B (ARRAY OCTAL)) (FN BOOLEAN))
   (BLOCK NIL (IF (EQN A B) (LABEL LIVE (RETURN TRUE)))
    (BLOCK ((BA BOOLEAN (ATOM B)))
     (IF (ATOM A) (IF BA (GO AT) (GO DIE)) BA (GO DIE)))
    (RETURN (AND (EQL. (CAR A) (CAR B) FN)
      (EQL. (CDR A) (CDR B) FN)))
    AT (IF (OR (LS (S20. A) TRP) (LS (S20. B) TRP)) (GO DIE))
    (IF (AND FN (NUMBP A) (NUMBP B))
     (BLOCK ((BR BOOLEAN (REALP B)))
      (IF (REALP A)
       (IF (NOT BR) (SET B (FLOAT B)))
       BR (SET A (FLOAT A))
       (BLOCK ((BS BOOLEAN (SPACEP. B)))
        (IF (SPACEP. A) (IF BS (GO AR) (GO DIE)) BS (GO DIE))
        (RETURN (EQ (WORDXOR (S20. A) (S20. B)) 4Q5)))) (GO AR)))
    (IF (NOT (AND (SPACEP. A) (SPACEP. B)))
     (LABEL DIE (RETURN FALSE)))
    AR (BLOCK ((C OCTAL (BIT 24 24 (CORES A)))
      (D OCTAL (BIT 24 24 (CORES B))))
     (IF (OR (AND (NOT FN) (NQ C D))
       (NQ (WORDAND (WORDXOR C D) 20777777Q) 0)) (GO DIE))
     (BLOCK ((U OCTAL (BIT 18 3 C))
       (V OCTAL (BIT 18 3 D)) (FLAG BOOLEAN NIL))
      (SET C (PLUS (BIT 0 18 C) -1))
      TNQ (CASE (PLUS U 1)
       (CASE (PLUS V 1)
        (GO SS) (GO CA) (GO SI) (GO SI) (GO SR) (GO DIE))
       (IF (EQ V 1) (GO CA) (GO XCH))
       (CASE (PLUS V -1) (GO CA) (GO OI) (GO OR) (GO XCH))
       (CASE (PLUS V -2) (GO II) (GO OR) (GO XCH))
       (IF (EQ V 4Q) (GO II) (GO XCH))
       (IF (EQ V 5Q) (GO FF) (GO XCH))
       (IF (EQ V 6Q) (GO CA) (GO DIE)))
      XCH (IF (NOT (SET FLAG (NOT FLAG))) (GO DIE))
      (CODE (LDA U) (ECH V) (STF U) (LDA A) (ECH B) (STF A))
      (GO TNQ))
     CA (FOR C (STEP C -1 EQ 0) (IF (NQ (A C) (B C)) (GO DIE)))
     (GO LIVE)
     SS (FOR C (STEP C -1 EQ 0)
      (IF (NOT (EQL. (O2S. (A C)) (O2S. (B C)) FN)) (GO DIE)))
     (GO LIVE)
     OI (IF (LS C 1) (GO LIVE))
     II (FOR C (STEP C -1 EQ 0)
      (IF (NOT (EQ (O2I. (A C)) (O2I. (B C)))) (GO DIE)))
     (GO LIVE)
     FF (FOR C (STEP C -1 EQ 0)
      (IF (NOT (EQ (BIT 0 24 (A C)) (BIT 0 24 (B C)))) (GO DIE)))
     (GO LIVE)
     SI (FOR C (STEP C -1 EQ 0)
      (IF (NOT (OR (AND (FIXP (O2S. (A C)))
          (EQ (SYM2INT (O2S. (A C))) (B C)))
         (AND (REALP (O2S. (A C)))
          (EQ (CORE (PLUS (A C) 1)) (FLOAT (B C)))))) (GO DIE)))
     (GO LIVE)
     SR (FOR C (STEP C -1 EQ 0)
      (IF (NOT (OR (AND (REALP (O2S. (A C)))
          (EQ (O2R. (CORE (PLUS (A C) 1))) (O2R. (B C))))
         (AND (FIXP (O2S. (A C))) (EQ (O2S. (A C)) (O2R. (B C))))))
       (GO DIE)))
     (GO LIVE)
     OR (FOR C (STEP C -1 EQ 0)
      (IF (NOT (EQ (FLOAT (A C)) (O2R. (B C)))) (GO DIE)))
     (GO LIVE)))))
(ARRAYS (SECTION SYS SYMBOL)
  (FUNCTION ((COPYARRAY . LISP) SYMBOL)
   ((A (ARRAY OCTAL)))
   (BLOCK NIL (IF (OR (LS (S2O. A) ARO) (GQ (S2O. A) ARP))
     (RETURN NIL))
    (BLOCK ((I INTEGER (ARSIZE (S2O. A))))
     (BLOCK ((B (ARRAY OCTAL) (GETARRAY I)))
      (FOR I (STEP (PLUS I -1) -1 LS 0) (SET (B I) (A I)))
      (SET (RGHTAD (S2O. B)) (S2O. B)) (RETURN B)))))
  (FUNCTION ((CREATE . LISP) SYMBOL)
   ((N INTEGER) (TYPE SYMBOL) (VALUE SYMBOL))
   (BLOCK ((S (ARRAY OCTAL) (GETARRAY (PLUS N 1)))
     (P OCTAL 0)
     (V OCTAL (IF VALUE (CONVRT TYPE VALUE) (DFINIT TYPE)))
     (X SYMBOL (FINDN TYPE (QUOTE ((SYMBOL . 2Q1)
         (OCTAL . 22Q)
         (INTEGER . 23Q)
         (REAL . 24Q) (FUNCTIONAL . 25Q) (BOOLEAN . 21Q))))))
    (IF X (SET P (CDR X)) (ERRMSG TYPE TYPMSG))
    (SET (PREFIX (S2O. S)) P)
    (IF (LS N 1) (GO R))
    (FOR N (STEP N -1 LS 1) (SET (S N) V)) R (RETURN S)))
  (ROUTINE (ARSPAC BOOLEAN)
   ((S SYMBOL)) (AND (GQ (S2O. S) ARO) (LS (S2O. S) ARP)))
  (FUNCTION (GETARRAY SYMBOL)
   ((SIZE INTEGER))
   (BLOCK ((X OCTAL ARP))
    (IF (LS SIZE 1)
     (RETURN NIL) (LS (SET ARP (I2O. (PLUS ARP SIZE))) LSP) (GO A))
    (SET ARP X)
    (RECLAIM SIZE)
    (SET X ARP)
    (SET ARP (I2O. (PLUS ARP SIZE)))
    A (SET (CORE X) (WORDOR X (SHIFT SIZE 24) 22Q14))
    (RETURN (O2S. X))))
  (ROUTINE ((TRUNC. . LISP) SYMBOL)
 ((S SYMBOL) (N INTEGER))
 (IF (ARRAYP S)
  (BLOCK ((Z INTEGER (ARSIZE (S2O. S)))
   (W OCTAL (I2O. (PLUS (S2O. S) (SET N (PLUS N 1))))))
   (BLOCK ((I INTEGER (PLUS Z (MINUS N))))
    (IF (NOT (GR Z N))
     (GO R) (NOT (LS (PLUS Z (S2O. S)) ARP)) (GO B))
    (SET (CORE W) (WORDOR W (SHIFT (I2O. I) 24) 22Q14))
    A (SET (ARSIZE (S2O. S)) N)
    R (RETURN S) B (SET ARP (PLUS ARP (MINUS I))) (GO A))) NIL))
(FUNCTION ((SCONCS . LISP) SYMBOL)
 ((A SYMBOL) (B SYMBOL)) (FCONC. A B NIL))
(FUNCTION ((NCONCS . LISP) SYMBOL)
 ((A SYMBOL) (B SYMBOL)) (FCONC. A B TRUE))
(FUNCTION (FCONC. SYMBOL)
 ((A (ARRAY OCTAL)) (B (ARRAY OCTAL)) (FLAG BOOLEAN))
 (BLOCK ((S (ARRAY OCTAL) A)
   (SA SYMBOL (STYPE A)) (SB SYMBOL (STYPE B)))
  (IF A (IF B (GO TEST) FLAG (RETURN A) (GO COPY))
   B (LABEL BS (SET S B)) (RETURN NIL))
  COPY (SET S (COPYARRAY S))
  R (RETURN S)
  TEST (IF (NQ SA SB)
   (ERRMSG (CONS SA SB)
    (QUOTE (DIFFERENT STRUCTURES. SCONCS OR NCONCS))))
  (BLOCK ((C INTEGER (ARSIZE (S2O. A)))
    (D INTEGER (ARSIZE (S2O. B))))
   (BLOCK ((Z INTEGER (PLUS C D -1)))
    (IF (NOT (OR (STRINGP A) (ARRAYP A)))
     (ERRMSG SA (QUOTE (NOT STRING OR ARRAY. SCONCS OR NCONCS)))
     (EQ C 1)
     (GO BS) (NQ D 1) (GO MERGE) FLAG (RETURN A) (GO COPY))
    MERGE (BLOCK ((I INTEGER (PLUS D -1))
      (E INTEGER)
      (G INTEGER (SHIFT (TAG (S2O. A)) 3))
      (H INTEGER (SHIFT (TAG (S2O. B)) 3))
      (J OCTAL (I2O. (PLUS (TAG (S2O. A)) (TAG (S2O. B))))))
     (IF (EQ G 0) (GO Z0) (NOT (GR (PLUS G H) WDSIZE)) (GO Z1))
     (SET J (I2O. (PLUS J (MINUS (SHIFT WDSIZE -3)))))
     (GO Z0)
     Z1 (SET Z (PLUS Z -1))
     Z0 (IF (AND FLAG (EQ ARP (PLUS (S2O. A) C))) (GO ES))
     (SET FLAG NIL)
     (SET S (GETARRAY Z))
     (GO PR)
     ES (IF (NOT (LS (SET E (PLUS (S2O. A) Z)) LSP)) (GO AR))
     (SET ARP E)
     SE (SET S A)
     (GO PR)
     AR (RECLAIM I)
     (SET ARP (PLUS (S2O. A) Z))
     (GO SE)
     PR (SET (CORES S) (WORDOR (SHIFT (I2O. Z) 24) (S2O. S)))
     (SET (PREFIX (S2O. S)) (PREFIX (S2O. B)))
     (SET (TAG (S2O. S)) J)
     (IF (NOT FLAG)
      (FOR E (STEP (PLUS C -1) -1 LS 1) (SET (S E) (A E))))
     (IF (OR (EQ G 0) (NOT (LS G WDSIZE))) (GO FILL))
     (SET H (MINUS G))
     (SET G (DIFFERENCE WDSIZE G))
     (FOR E (STEP (PLUS Z -1) -1 LS C)
      (BLOCK NIL (SET I (PLUS I -1)) (SET (S E) (SHIFT (B I) G))))
     (SET E (PLUS Z -1))
```

```
      (FOR I (STEP (PLUS D -1) -1 LS 1)
       (BLOCK NIL (SET (S E) (WORDOR (S E) (SHIFT (B I) H)))
        (SET E (PLUS E -1))))
      (GO R)
      FILL (SET E Z)
      (FOR I (STEP I -1 LS 1) (SET (S (SET E (PLUS E -1))) (B I)))
      (GO R)))))))
(LISTS (SECTION SYS SYMBOL)
  (FUNCTION CONS2 (A B)
   (BLOCK ((S SYMBOL (C2S. (SET LSP (I2O. (PLUS LSP -1))))))
    (SET (CORE (S2C. S)) (S2O. B))
    (SET (CAR S) A) (IF (NOT (LS ARP LSP)) (RECLAIM 1)) (RETURN S)))
  (FUNCTION CONS3 (A B C) (CONS A (CONS B C)))
  (FUNCTION CONS4 (A B C D) (CONS A (CONS B (CONS C D))))
  (FUNCTION (LIST1 SYMBOL) ((X SYMBOL)) (CONS X NIL))
  (FUNCTION (LIST2 SYMBOL) ((X SYMBOL) (Y SYMBOL)) (CONS X Y NIL))
  (FUNCTION (LIST3 SYMBOL)
   ((X SYMBOL) (Y SYMBOL) (Z SYMBOL)) (CONS X Y Z NIL))
  (FUNCTION (LIST4 SYMBOL)
   ((X SYMBOL) (Y SYMBOL) (Z SYMBOL) (W SYMBOL)) (CONS X Y Z W NIL))
  (SECTION (LISP SYS FSM) SYMBOL)
  (FUNCTION LASTN ((I INTEGER) X)
   (BLOCK ((Y (FIRSTN I (SET X (DREVERSE X)))))
    (SET X (DREVERSE X)) (RETURN (DREVERSE Y))))
  (FUNCTION FIRSTN ((I INTEGER) X)
   (IF (OR (EQ I 0) (NULL X))
    NIL (CONS (CAR X) (FIRSTN (DIFFERENCE I 1) (CDR X)))))
  (ROUTINE DREVERSE (L)
   (BLOCK (M)
    Z (IF (NULL L) (RETURN M))
    (BLOCK ((N (CDR L))) (SET (CDR L) M) (SET M L) (SET L N))
    (GO Z)))
  (FUNCTION ((EXPLODE . LISP) SYMBOL)
   ((S (ARRAY OCTAL)))
   (BLOCK ((N INTEGER (STRINGL (IF (STRINGP S)
          S (SET S (TOSTRG S))))) (R SYMBOL))
    (IF (LS N 1) (RETURN NIL))
    (FOR N (STEP N -1 LS 1) (SET R (CONS (GETCHAR S N) R)))
    (RETURN R)))
  (FUNCTION ((COMPRESS . LISP) SYMBOL)
   ((L SYMBOL))
   (BLOCK NIL (FOR FSCHAR (IN L) (MAKEST))
    (SET FSCHAR NIL) (RETURN (MAKEST))))
  (FUNCTION (REVERSE SYMBOL)
   ((L SYMBOL))
   (BLOCK ((X SYMBOL) (M SYMBOL))
    (FOR X (IN L) (SET M (CONS X M))) (RETURN M)))
  (FUNCTION (LAST SYMBOL)
   ((L SYMBOL))
   (BLOCK NIL A (IF (ATOM L)
     (RETURN L) (NULL (CDR L)) (RETURN (CAR L)))
    (SET L (CDR L)) (GO A)))
  (FUNCTION (FIND SYMBOL)
   ((A SYMBOL) (B SYMBOL))
   (BLOCK ((C SYMBOL))
    (FOR C (IN B)
     (IF (AND (NOT (ATOM C)) (EQUALN A (CAR C))) (RETURN C)))
    (RETURN NIL)))
  (ROUTINE (FINDN SYMBOL)
   ((A SYMBOL) (B SYMBOL))
   (BLOCK ((C SYMBOL))
    (FOR C (IN B)
     (IF (AND (NOT (ATOM C)) (EQN A (CAR C))) (RETURN C)))
    (RETURN NIL)))
  (FUNCTION (MEMBER BOOLEAN)
   ((A SYMBOL) (B SYMBOL))
   (BLOCK NIL L (IF (ATOM B)
     (RETURN NIL) (EQUALN A (CAR B)) (RETURN TRUE))
    (SET B (CDR B)) (GO L)))
  (FUNCTION (MAPFN SYMBOL)
   ((L SYMBOL) (FN (FUNCTIONAL SYMBOL SYMBOL)))
   (BLOCK ((X SYMBOL (LIST NIL)))
    (BLOCK ((Y SYMBOL X) (Z SYMBOL))
     A (IF (NULL (SET Z (FN L))) (GO D) (ATOM Z) (GO E))
     (SET (CDR Y) Z)
     B (IF (ATOM (CDR Z)) (GO C))
     (SET Z (CDR Z))
     (GO B)
     C (SET Y Z)
     D (IF (ATOM L) (LABEL E (RETURN (CDR X))))
     (SET L (CDR L)) (GO A))))
  (FUNCTION (MAPCAR SYMBOL)
   ((L SYMBOL) (FN (FUNCTIONAL SYMBOL SYMBOL)))
   (MAPFN L (FUNARG SYMBOL ((J SYMBOL))
     (IF (ATOM J) NIL (LIST (FN (CAR J)))))))
  (FUNCTION (MAPLIST SYMBOL)
   ((L SYMBOL) (FN (FUNCTIONAL SYMBOL SYMBOL)))
   (MAPFN L (FUNARG SYMBOL ((J SYMBOL))
     (IF (ATOM J) NIL (LIST (FN J))))))
  (FUNCTION (MAP NOVALUE)
   ((L SYMBOL) (FN (FUNCTIONAL SYMBOL SYMBOL)))
   (BLOCK ((J SYMBOL)) (FOR J (ON L) (FN J))))
  (FUNCTION (APPEND SYMBOL)
   ((A SYMBOL) BBB8)
   (MAPFN A (FUNARG SYMBOL ((J SYMBOL))
     (IF (ATOM J) BBB8 (LIST (CAR J))))))
  (FUNCTION (NCONC SYMBOL)
   (AAA8 BBB8)
   (MAPFN (QUOTE (NIL))
    (FUNARG SYMBOL ((J SYMBOL)) (IF (NULL J) BBB8 AAA8))))
  (FUNCTION (DELETEL SYMBOL)
   (AAA8 (B SYMBOL))
   (MAPFN B (FUNARG SYMBOL ((J SYMBOL))
     (IF (OR (ATOM J) (MEMBER (CAR J) AAA8)) NIL (LIST (CAR J))))))
  (FUNCTION (LENGTH INTEGER)
   ((L SYMBOL))
   (BLOCK ((N INTEGER))
    (FOR L (ON L) (SET N (PLUS N 1))) (RETURN N)))
  (ROUTINE (NOFF SYMBOL)
   ((N INTEGER) (L SYMBOL))
   (BLOCK NIL (FOR L (ON L) (WHILE (GR N 0)) (SET N (PLUS N -1)))
    (RETURN L)))
  (ROUTINE (MAX INTEGER)
   ((I INTEGER) (J INTEGER)) (IF (GR I J) I J))
  (ROUTINE (MIN INTEGER)
   ((I INTEGER) (J INTEGER)) (IF (LS I J) I J))
  (ROUTINE (MINR REAL) ((A REAL) (B REAL)) (IF (LS A B) A B))
  (ROUTINE (MAXR REAL) ((A REAL) (B REAL)) (IF (GR A B) A B))
  (FUNCTION (SUBST SYMBOL)
   ((X SYMBOL) (Y SYMBOL) (Z SYMBOL))
   (IF (EQUALN Y Z)
    X (ATOM Z) Z (CONS (SUBST X Y (CAR Z)) (SUBST X Y (CDR Z))))))
(STRING (SECTION (LISP SYS FSM) SYMBOL)
  (FUNCTION (STRINGL INTEGER)
   ((S SYMBOL))
   (IF (STRINGP S)
  (PLUS (TAG (S2O. S)) (TIMES (ARSIZE (S2O. S)) 6) -12)
  (ERRMSG S (QUOTE (NOT A STRING)))))
(SECTION (FSM SYS) SYMBOL)
(FUNCTION (MAKEST SYMBOL)
 NIL (BLOCK ((ST SYMBOL)
   (I OCTAL (I2O. (PLUS 1 (TAG (S2O. BASEST))))))
  (IF (NOT FSCHAR) (GO START))
  (BLOCK ((OCT OCTAL (CH2OCT FSCHAR)))
   (CASE I (GO S0)  (GO S1) (GO S2) (GO S3) (GO S4) (GO S5) (GO S6))
   S0 (SET (BASEST 1) (SHIFT OCT 40))
   (GO R)
   S1 (SET OCT (SHIFT OCT 32))
   S5 (SET (BASEST 1) (WORDOR OCT (BASEST 1)))
   R (SET (TAG (S2O. BASEST)) I)
   (RETURN NIL)
   S2 (SET OCT (SHIFT OCT 24))
   (GO S5)
   S3 (SET OCT (SHIFT OCT 16))
   (GO S5)
   S4 (SET OCT (SHIFT OCT 8))
   (GO S5)
   S6 (SET WORKST (NCONCS WORKST BASEST))
   (SET I 1Q) (GO S0))
  START (SET ST (NCONCS WORKST BASEST))
  (SET WORKST NIL) (SET (TAG (S2O. BASEST)) 0Q) (RETURN ST)))
(FUNCTION ((TOSTRG . LISP) SYMBOL)
 ((S SYMBOL))
 (BLOCK NIL (IF (STRINGP S)
   (RETURN (COPYARRAY S))
   (CHARP S)
   (GO CH)
   (IDP S)
   (GO ID)
   (NUMBP S)
   (RETURN (NUMSTR S))
   (NOT S)
   (GO NL)
   (BOOLP S)
   (GO TR) (FORMALP S) (BLOCK NIL (SET S (QUOTE F.....)) (GO ID)))
  (ERRMSG S (QUOTE (NOT A TOKEN)))
  CH (SET FSCHAR NIL)
  (MAKEST)
  (SET FSCHAR S)
  (MAKEST)
  R (SET FSCHAR NIL)
  (RETURN (MAKEST))
  ID (IF (AND (GENIDP S) (EQ (WORD1 (S2O. S)) 0Q))
   (GENPNAME S))
  IDS (IF (EQ (BIT 2 1 (TAG (S2O. S))) 0Q)
   (BLOCK ((A (ARRAY OCTAL) (GETARRAY 2))
      (B OCTAL (TAGIM (WORD3 (S2O. S)))))
    (SET (PREFIX (S2O. A)) 6Q)
    (IF (EQ B 0Q) (RETURN (TRUNC. A 0)))
    (SET (A 1) (WORD1 (S2O. S)))
    (SET (TAG (S2O. A)) B) (RETURN A)))
  (RETURN (COPYARRAY (C2S. (PNAME (S2O. S)))))
  TR (SET S TRUE.) (GO IDS) NL (SET S NIL.) (GO IDS)))
(SECTION (SYS FSM LISP) SYMBOL)
(FUNCTION (NUMSTR SYMBOL)
 ((X SYMBOL))
 (BLOCK ((I INTEGER))
  (IF (INTP X)
   (BLOCK NIL (SET I X)
 A (IF (LS I 0)
  (BLOCK NIL (SET FSCHAR (QUOTE '-))
   (MAKEST) (SET I (MINUS I)))))
(REALP X)
(BLOCK ((X REAL X))
 (IF (EQ X 0.0) (BLOCK NIL (STMAKE (QUOTE ('0 '. '0))) (GO E)))
 (IF (LS X 0.0)
  (BLOCK NIL (SET FSCHAR (QUOTE '-))
   (MAKEST) (SET X (MINUS X))))
 (BLOCK ((J INTEGER (PLUS (BIT 36 11 (R20. X)) -1025))
   (M REAL))
  (SET I (ENTIER (TIMES 0.3010299954 J)))
  C (SET M (EXPT 10.0 (ABS I)))
  (IF (GQ (SET M (IF (LS I 0)
      (TIMES X M) (GR I 0) (QUOTIENT X M) X)) 10.0)
   (SET I (PLUS I 1)) (LS M 1.0) (SET I (PLUS I -1)) (GO D))
  (GO C)
  D (BLOCK ((K SYMBOL NIL) (CARRY BOOLEAN FALSE))
   (SET J 0)
   (IF (LS I 0) (GO G))
   (IF (GR I 11) (SET X (QUOTIENT X (EXPT 10.0 (PLUS I -11)))))
   (BLOCK ((XX INTEGER (ENTIER X)))
    (SET M (TIMES 10.0 (DIFFERENCE X XX)))
    LOOP (IF (EQ XX 0) (GO GG))
    (SET K (CONS (OCT2CH (WORDOR 6Q1 (REMAINDER XX 10))) K))
    (SET J (PLUS J 1))
    (SET XX (IQUOTIENT XX 10))
    (GO LOOP) GG (SET K (DREVERSE K)))
   G (FOR J (STEP J 1 EQ 12)
    (BLOCK ((XX INTEGER (ENTIER M)))
     (SET K (CONS (OCT2CH (WORDOR 6Q1 XX)) K))
     (SET M (TIMES 10.0 (DIFFERENCE M XX)))))
   (IF (GR M 5.0) (SET CARRY TRUE))
   (BLOCK ((KK SYMBOL NIL) (EL SYMBOL))
    (FOR EL (IN K)
     (UNLESS (AND (EQN (IF (NOT CARRY)
        EL (SET EL (IF (EQN EL (QUOTE '9))
          (QUOTE '0)
          (BLOCK NIL (SET CARRY FALSE)
           (RETURN (CHEAT INTEGER SYMBOL (PLUS 1 (CHEAT SYMBOL
              INTEGER EL)))))))) (QUOTE '0)) (NOT KK)))
     (SET KK (CONS EL KK))) (SET K KK))
   (IF (LQ (SET I (PLUS I 1)) 0) (GO S))
   (SET FSCHAR (CAR K))
   (MAKEST)
   (SET K (CDR K))
   (SET I (PLUS I -1))
   S (SET FSCHAR (QUOTE '.))
   (MAKEST)
   U (IF (NULL K)
    (IF (EQ I 0)
     (GO E) (BLOCK NIL (SET FSCHAR (QUOTE E)) (MAKEST) (GO A)))
    (BLOCK NIL (SET FSCHAR (CAR K))
     (MAKEST) (SET K (CDR K)) (GO U))))))
(OCTALP X)
(BLOCK ((Q OCTAL X))
 (FOR I (STEP 15 -1 EQ 0)
  (WHILE (EQ (BIT 45 3 Q) 0Q)) (SET Q (SHIFT Q 3)))
 (FOR I (STEP I -1)
  (BLOCK NIL (SET FSCHAR (OCT2CH (WORDOR 6Q1 (BIT 45 3 Q))))
   (MAKEST)
   (SET Q (SHIFT Q 3))
   (IF (AND (EQ Q 0) (OR (EQ I 0) (GR I 3))) (GO Q))))
```

```
     Q (SET FSCHAR (QUOTE Q)) (MAKEST) (IF (EQ I 0) (GO E))))
    I (BLOCK ((K SYMBOL NIL))
     X (SET K (CONS (OCT2CH (WORDOR 6Q1 (REMAINDER I 10))) K))
     (SET I (IQUOTIENT I 10)) (IF (GR I 0) (GO X)) (STMAKE K))
    E (SET FSCHAR NIL) (RETURN (MAKEST))))
  (FUNCTION (STMAKE NOVALUE)
   ((A SYMBOL)) (FOR FSCHAR (IN A) (MAKEST))))
(IDENTS (SECTION SYS SYMBOL)
  (DECLARE (GENNO INTEGER OWN 10000) (GENPFX SYMBOL OWN (QUOTE A)))
  (FUNCTION ((GETID . LISP) SYMBOL)
   ((S (ARRAY OCTAL)))
   (BLOCK ((B INTEGER (S20. (OBLIST (BUCKET S))))
     (W INTEGER (LEFTAD (S20. S))))
    (IF (GR W 2)
     (GO LONG)
     (EQ (TAG (S20. S)) 1)
     (RETURN (OCT2CH (BIT 40 8 (S 1))))
     (LS W 2) (SET W 0) (SET W (S 1)))
    SHORT (IF (EQ B 0)
     (GO NO)
     (AND (NQ (BIT 20 1 (WORD2 B)) 1) (EQ W (WORD1 B)))
     (RETURN (O2S. B)))
    (SET B (LINK B))
    (GO SHORT)
    LONG (SET W (BIT 24 24 (S 1)))
    TEST (IF (EQ B 0)
     (GO NO)
     (AND (NQ (BIT 20 1 (WORD2 B)) 0)
      (EQ (BIT 24 24 (WORD1 B)) W) (EQ S (O2S. (PNAME B))))
     (RETURN (O2S. B))) (SET B (LINK B)) (GO TEST) NO (RETURN NIL)))
  (FUNCTION (GENPNAME SYMBOL)
   ((S SYMBOL))
   (BLOCK ((P (ARRAY OCTAL)
      (SCONCS (TOSTRG GENPFX) (NUMSTR GENNO))))
    (IF (GQ GENNO 100000) (GO B))
    A (SET (WORD1 (S20. S)) (P 1))
    (SET (TAGIM (WORD3 (S20. S))) (TAGIM (CORE (S20. P))))
    (SET GENNO (I20. (PLUS GENNO 1)))
    (RETURN S)
    B (SET GENNO 0Q)
    (SET GENPFX (IF (EQ GENPFX (QUOTE Z))
      (QUOTE A) (CHEAT INTEGER SYMBOL (PLUS (S20. GENPFX) 1))))
    (GO A)))
  (FUNCTION (GENID SYMBOL)
   NIL (BLOCK ((S SYMBOL ((TRIPLE . SYS))))
    (SET (WORD1 (S20. S)) 0)
    (SET (WORD2 (S20. S)) 70000000106)
    (SET (CHAINS S) (S20. S)) (SET (WORD3 (S20. S)) 0) (RETURN S)))
  (DECLARE (OBLIST (ARRAY SYMBOL) OWN)
   (OBLSIZ INTEGER OWN 125) (BUCKNO INTEGER OWN))
  (ROUTINE (BUCKET INTEGER)
   ((S (ARRAY OCTAL)))
   (SET BUCKNO (IF (LS (LEFTAD (S20. S)) 2)
     1 (PLUS 1 (REMAINDER (ABS (S 1)) OBLSIZ)))))
  (SECTION (FSM SYS) SYMBOL)
  (DECLARE (BASEST (ARRAY OCTAL) OWN (QUOTE (*STRING AAAAA)))
   (WORKST (ARRAY OCTAL) OWN NIL)
   (FSMSYM SYMBOL OWN)
   (RMSG (ARRAY SYMBOL)
    OWN (QUOTE (*SYMBOL (*STRING '  IS '  ILLEGAL '  TOKEN '  SYNTAX)
      (*STRING '  IS '  ILLEGAL '  TOKEN '  IN '  DATUM)
      (*STRING '  FILE '  TERMINATOR '  INSIDE '  DATUM)
      (*STRING '. '  IS '  ILLEGAL '  AFTER '  LPAR)
     (*STRING '  FOUND '  INSTEAD '  OF '  ')
     '  IN '  DOTTED '  PAIR))))
 (XXCHAR SYMBOL OWN)
 ((GNLIST . SYS) SYMBOL NIL)
 (SPFLAG BOOLEAN OWN NIL) (FSCHAR SYMBOL OWN NIL))
 FUNCTION ((MAKID . FSM) SYMBOL)
 NIL (BLOCK ((S SYMBOL ((GETID . LISP) FSMSYM)))
  (IF S (RETURN S))
  (SET S (TRIPLE))
  (BLOCK ((A (ARRAY OCTAL) FSMSYM)
    (L SYMBOL ((OBLIST . SYS) (BUCKNO . SYS)))
    (W2 OCTAL (IF SPFLAG 1Q7 0Q)))
   (BLOCK ((N OCTAL (ARSIZE (S20. A))) (W3 OCTAL (S20. L)))
    (IF (NQ N 2) (GO LONG))
    (SET (WORD1 (S20. S)) (A 1))
    (SET (TAGIM W3) (TAG (S20. A)))
    R (SET (WORD3 (S20. S)) W3)
    (SET (PREFIM W2) 7Q)
    (SET (WORD2 (S20. S)) W2)
    (SET (CHAINS S) (S20. S))
    (SET ((OBLIST . SYS) (BUCKNO . SYS)) S)
    (RETURN S)
    LONG (IF (GR N 2) (GO L1))
    (SET W2 1Q7)
    (GO R)
    L1 (SET (WORD1 (S20. S))
     (WORDOR (S20. FSMSYM) (WORDAND 77777777Q8 (A 1))))
    (SET (BIT 20 1 W2) 1Q) (GO R)))))
(FUNCTION ((MAKEID . LISP) SYMBOL)
 ((A (ARRAY OCTAL)))
 (BLOCK NIL (SET SPFLAG (SPELLP A))
  (SET FSMSYM (COPYARRAY A)) (RETURN ((MAKID . FSM)))))
(FUNCTION (MGENID SYMBOL)
 NIL (BLOCK ((R SYMBOL (FIND FSMSYM GNLIST)))
  (IF R (RETURN (CDR R)))
  (SET GNLIST (CONS (CONS FSMSYM (SET R (GENID))) GNLIST))
  (RETURN R)))
(FUNCTION ((MAKIDB . FSM) SYMBOL)
 NIL (BLOCK ((A (ARRAY OCTAL) FSMSYM))
  (IF (EQ (LEFTAD (S20. A)) 2)
   (BLOCK ((W OCTAL (A 1)))
    (IF (EQ W (WORD1 (S20. TRUE.)))
     (RETURN TRUE)
     (OR (EQ W (WORD1 (S20. NIL.))) (EQ W (WORD1 (S20. FALSE.))))
     (RETURN NIL)))) (SET SPFLAG NIL) (RETURN ((MAKID . FSM)))))
(DECLARE (TRUE. SYMBOL OWN (QUOTE (*IDENTIFIER TRUE)))
 (FALSE. SYMBOL OWN (QUOTE (*IDENTIFIER FALSE)))
 (NIL. SYMBOL OWN (QUOTE (*IDENTIFIER 'N 'I 'L))))
(SECTION (IO FSM LISP SYS) SYMBOL)
(DECLARE (STSPEL SYMBOL OWN)
 (STSAVE SYMBOL OWN) (STR.CH INTEGER OWN))
(FUNCTION ((S.SUPL . IO) SYMBOL)
 NIL (IF (LS (STRINGL STSPEL) STR.CH)
  (OCT2CH 34Q)
  (BLOCK ((X SYMBOL (GETCHAR STSPEL STR.CH)))
   (SET STR.CH (PLUS STR.CH 1)) (RETURN X))))
 SECTION (FSM IO LISP SYS) SYMBOL)
(FUNCTION (TOKEN INTEGER) NIL)
(FUNCTION ((PARSE . LISP) INTEGER)
 ((A SYMBOL) (C INTEGER))
 (BLOCK ((TT (FUNCTIONAL SYMBOL) (XXFUNC . IO))
   (ZZ SYMBOL (XXSAVE . IO)))
  (SET (XXFUNC . IO) S.SUPL)
```

```
   (SET (XXSAVE . IC) STSAVE)
   (SET STSPEL A)
   (SET STR.CH C)
   (SET C (TOKEN))
   (SET STSAVE (XXSAVE . IC))
   (SET (XXFUNC . IC) TT) (SET (XXSAVE . IC) ZZ) (RETURN C)))
 (FUNCTION ((SPELLP . FSM) BOOLEAN)
  ((A (ARRAY OCTAL)))
  (BLOCK ((L INTEGER (STRINGL A)))
   (RETURN (OR (LS L 1)
     (AND (LS L 6)
      (OR (EQ (SET L (A 1)) (WORD1 (S2O. TRUE.)))
       (EQ L (WORD1 (S2O. NIL.))) (EQ L (WORD1 (S2O. FALSE.)))))
     (LS (BLOCK NIL (SET STSAVE NIL)
       (SET L (PARSE A 1))
       (IF (GR STR.CH (STRINGL STSPEL)) (RETURN L) (RETURN 0))) 12)
     (GR L 15))))) (SECTION LISP SYMBOL))
(ARITH (SECTION SYS SYMBOL)
  (FUNCTION (SYMSGN INTEGER) ((A SYMBOL)) (SIGN (SYM2REAL A)))
  (FUNCTION (SYMABS SYMBOL) ((A SYMBOL)) (TIMES A (SIGN A)))
  (FUNCTION (STIMS SYMBOL)
   ((A SYMBOL) (B SYMBOL))
   (IF (FIXP A) (TIMES (SYM2INT A) B) (TIMES (SYM2REAL A) B)))
  (FUNCTION (STIMR REAL)
   ((A REAL) (B SYMBOL)) (TIMES A (SYM2REAL B)))
  (FUNCTION (STIMI SYMBOL)
   ((A INTEGER) (B SYMBOL))
   (IF (FIXP B) (TIMES A (SYM2INT B)) (TIMES A (SYM2REAL B))))
  (FUNCTION (SPLUS SYMBOL)
   ((A SYMBOL) (B SYMBOL))
   (IF (FIXP A) (PLUS (SYM2INT A) B) (PLUS (SYM2REAL A) B)))
  (FUNCTION (SPLUR REAL)
   ((A REAL) (B SYMBOL)) (PLUS A (SYM2REAL B)))
  (FUNCTION (SPLUI SYMBOL)
   ((A INTEGER) (B SYMBOL))
   (IF (FIXP B) (PLUS A (SYM2INT B)) (PLUS A (SYM2REAL B))))
  (FUNCTION (SMINS SYMBOL)
   ((A SYMBOL) (B SYMBOL))
   (IF (FIXP B)
    (DIFFERENCE A (SYM2INT B)) (DIFFERENCE A (SYM2REAL B))))
  (FUNCTION (SMINI SYMBOL)
   ((A INTEGER) (B SYMBOL))
   (IF (FIXP B)
    (DIFFERENCE A (SYM2INT B)) (DIFFERENCE A (SYM2REAL B))))
  (FUNCTION (SMINR REAL)
   ((A REAL) (B SYMBOL)) (DIFFERENCE A (SYM2REAL B)))
  (FUNCTION (MINSYM SYMBOL)
   ((A SYMBOL))
   (IF (FIXP A) (MINUS (SYM2INT A)) (MINUS (SYM2REAL A))))
  (SECTION (LISP SYS) SYMBOL)
  (ROUTINE ((REMAINDER . LISP) INTEGER)
   ((A INTEGER) (B INTEGER))
   (DIFFERENCE A (TIMES B (IQUOTIENT A B))))
  (ROUTINE (OCTROUND OCTAL) ((A REAL)) (ROUND A))
  (ROUTINE (ROUND INTEGER) ((N REAL)) (ENTIER (PLUS N 0.5)))
  (ROUTINE (ENTIER INTEGER)
   ((N REAL))
   (IF (GQ N 0)
    (SCALE (BIT 0 36 (R2O. N))
     (DIFFERENCE (BIT 36 12 (R2O. N)) 2044Q))
    (MINUS (ENTIER (DIFFERENCE (O2R. 2000777777777777Q) N)))))
  (FUNCTION (EXPT REAL)
   ((X REAL) (Y INTEGER))
   (IF (LS Y 0)
    (QUOTIENT 1.0 (EXPT X (MINUS Y)))
    (BLOCK ((R REAL 1.0))
     A (IF (EQ Y 0) (RETURN R))
     (SET Y (PLUS Y -1)) (SET R (TIMES R X)) (GO A)))))
  ITTER (SECTION SYS SYMBOL)
  (ROUTINE (BITTST BOOLEAN)
   ((X INTEGER) (Y INTEGER))
   (AND (GQ X 0) (GR Y 0) (LQ (PLUS X Y) WDSIZE)))
  (DECLARE (WDSIZE INTEGER OWN 48))
  (ROUTINE (BITS OCTAL)
   ((X INTEGER) (Y INTEGER) (Z OCTAL))
   (IF (BITTST X Y)
    (WORDAND (INVERT (SHIFT (INVERT 0Q) Y))
     (SHIFT Z (MINUS X))) 0Q))
  (ROUTINE (BITSET OCTAL)
   ((X INTEGER) (Y INTEGER) (Z OCTAL LOC) (W OCTAL))
   (IF (BITTST X Y)
    (SET Z (BLOCK ((MASK OCTAL (INVERT 0Q)))
      (SET Y (PLUS WDSIZE (MINUS Y)))
      (SET MASK (SHIFT (SHIFT (SHIFT MASK (MINUS X)) Y)
        (PLUS X (MINUS Y))))
      (RETURN (WORDOR (WORDAND (SHIFT W X) MASK)
        (WORDAND Z (INVERT MASK)))))) W)))
(CONVERTS (SECTION (LISP SYS) SYMBOL)
  (ROUTINE ((INT2OCT . LISP) OCTAL) ((A INTEGER)) A)
  (FUNCTION (SYM2OCT OCTAL)
   ((S (ARRAY OCTAL)))
   (BLOCK ((X INTEGER (DIFFERENCE (S2O. S) 4Q5)))
    (IF (GQ X 0)
     (GO A)
     (GQ (SET X (PLUS X 2Q5)) 0)
     (GO B)
     (OCTALP S)
     (RETURN (S 1)) (INTP S) (GO C) (REALP S) (GO D) (NUMERR S))
    A (SET X (DIFFERENCE X 2Q5))
    B (RETURN (IF (EQ X 0) 0Q (I2O. X)))
    C (SET X (S 1)) (GO B) D (SET X (ENTIER (O2R. (S 1)))) (GO B)))
  (FUNCTION (SYM2INT INTEGER) ((S SYMBOL)) (O2I. (SYM2OCT S)))
  (FUNCTION (SYM2REAL REAL)
   ((S (ARRAY OCTAL)))
   (IF (REALP S)
    (O2R. (S 1)) (FIXP S) (FLOAT (SYM2OCT S)) (NUMERR S)))
  (FUNCTION (NUMERR SYMBOL)
   ((S SYMBOL)) (ERROR (CONS S (QUOTE (NOT A NUMBER)))))
  (FUNCTION (OCT2SYM SYMBOL)
   ((X OCTAL))
   (IF (EQ (BIT 16 32 X) 0Q)
    (O2S. (PLUS X 2Q5))
    (BLOCK ((A (ARRAY OCTAL) (GETARRAY 2)))
     (SET (PREFIX (S2O. A)) 2) (SET (A 1) X) (RETURN A))))
  (FUNCTION (REAL2SYM SYMBOL)
   ((X REAL))
   (BLOCK ((A (ARRAY OCTAL) (GETARRAY 2)))
    (SET (PREFIX (S2O. A)) 4) (SET (A 1) (R2O. X)) (RETURN A)))
  (FUNCTION (INT2SYM SYMBOL)
   ((X INTEGER))
   (IF (AND (LS X 2Q5) (GQ X 7777777777577777Q))
    (O2S. (I2O. (PLUS X 6Q5)))
    (BLOCK ((A (ARRAY OCTAL) (GETARRAY 2)))
     (SET (PREFIX (S2O. A)) 3) (SET (A 1) (I2O. X)) (RETURN A))))
  (FUNCTION (CH2OCT OCTAL)
   ((S SYMBOL))
```

```
    (IF (CHARP S)                                                           0082000
     (DIFFERENCE (S2O. S) CHO)                                              0082100
     (ERRMSG S (QUOTE (IS NOT A CHARACTER)))))                              0082200
   (FUNCTION (OCT2CH SYMBOL)                                                0082300
    ((X OCTAL))                                                             0082400
    (IF (AND (LS (SET X (I2O. (PLUS X CHO))) TRO) (GQ X CHO))               0082500
     (O2S. X)                                                               0082600
     (ERRMSG (I2O. (DIFFERENCE X CHO))                                      0082700
      (QUOTE (IS NOT CHARACTER REPRESENTATION)))))                          0082800
   (FUNCTION (FORM2SYM SYMBOL)                                              0082900
    ((X (FUNCTIONAL NOVALUE)))                                              0083000
    (BLOCK ((A (ARRAY OCTAL) (GETARRAY 2)))                                 0083100
     (SET (PREFIX (S2O. A)) 5) (SET (A 1) (F2O. X)) (RETURN A)))            0083200
   (FUNCTION (SYM2FORM FUNCTIONAL)                                          0083300
    ((S (ARRAY OCTAL)))                                                     0083400
    (IF (FORMALP S)                                                         0083500
     (O2F. (S 1)) (ERRMSG S (QUOTE (IS NOT A FUNCTIONAL))))))               0083600
 (UTILITY (SECTION SYS SYMBOL)                                              0083700
   (FUNCTION (ERMSG1 SYMBOL) ((A SYMBOL)) (ERROR A))                        0083800
   (FUNCTION (ERRMSG SYMBOL)                                                0083900
    ((A SYMBOL) (B SYMBOL)) (ERROR (CONS A B)))                             0084000
   (FUNCTION (TRIPLE SYMBOL)                                                0084100
    NIL (BLOCK ((X OCTAL TRL))                                              0084200
     (IF (EQ X 0) (GO A))                                                   0084300
     (SET TRL (LINK TRL))                                                   0084400
     (RETURN (O2S. x))                                                      0084500
     A (SET TRP (I2O. (PLUS TRP 3)))                                        0084600
     (ADPDOK 3) (RETURN (O2S. (I2O. (PLUS TRP -2))))))                      0084700
   (ROUTINE (ADPDOK NOVALUE)                                                0084800
    ((C INTEGER))                                                           0084900
    (BLOCK NIL (SET (BIT 24 18 (CORENTRY PDOK1))                            0085000
       (PLUS (BIT 24 18 (CORENTRY PDOK1)) C))                               0085100
     (SET (BIT 24 18 (CORE (PLUS (ENTRY PDOK) 1)))                          0085200
       (PLUS (BIT 24 18 (CORE (PLUS (ENTRY PDOK) 1))) C))))                 0085300
   (ROUTINE (SPACEP. BOOLEAN)                                               0085400
    ((S SYMBOL)) (AND (GQ (S2O. S) TRO) (LS (S2O. S) LSO)))                 0085500
   (ROUTINE (TRSPAC BOOLEAN)                                                0085600
    ((S SYMBOL)) (AND (GQ (S2O. S) TRO) (LS (S2O. S) TRP))))                0085700
 (DECLARES (SECTION SYS SYMBOL)                                             0085800
   (DECLARE DECLU SQWKUN)                                                   0085900
   (FUNCTION MAKEFREE (N S STOR TYPE X)                                     0086000
    (BLOCK ((FR (GETFREE N S))                                              0086100
      (TES (FIND STOR KINDLIST))                                            0086200
      (WRD1 OCTAL) DCWRD1 (WRD3 OCTAL) DCWRD3)                              0086300
     (IF (NULL TES) (GO KINDER))                                            0086400
     (BLOCK ((TEST OCTAL (CDR TES)))                                        0086500
      (IF (EQ TEST 2Q)                                                      0086600
       (GO MS)                                                              0086700
       (NULL (TYPEP TYPE)) (GO TYPERR) (SET TYPE (STANTP TYPE)))            0086800
      (IF (NULL FR) (GO MF))                                                0086900
      (BLOCK ((FG (FVLIST FR)))                                             0087000
       (IF (NQ (CADR FG) TYPE)                                              0087100
        (GO REDEF)                                                          0087200
        (NQ (CADDR FG) X)                                                   0087300
        (GO REDEF)                                                          0087400
        (EQ TEST 0Q)                                                        0087500
        (GO R)                                                              0087600
        (NQ (CDR (FIND (CAR FG) KINDLIST)) TEST) (GO REDEF)))               0087700
     MX (SET (XMFLAG (S2O. FR)) (IF (NQ STOR (QUOTE FREE)) 1Q 0Q))          0087800
     R (RETURN FR)                                                          0087900
     REDEF (IF (NQ (VCOUNT (S2O. FR)) 0Q) (GO RDECER))                      0088000
     (MESSAGE (APPEND (QUOTE (NEW DECLARATION FOR))                         0088100
       (LIST (CONS N S))))                                                  0088200
   MF (IF (NQ TEST 0) (GO MM))
   (SET STOR (QUOTE FREE))
   (SET TEST 11Q)
   MM (SET WRD3 (MAKETYPE (LIST TYPE X)))
   (IF (EQ (BIT 24 6 WRD3) 1Q)
    (SET DCWRD3 (O2S. (BIT 6 18 WRD3))))
   (IF (GR TEST 12Q) (GO FN))
   (SET TES (FTYPER TYPE))
   (IF (AND (EQ TEST 12Q) (EQ X (QUOTE VALUE)))
    (SET WRD1 (DFINIT TES)) (SET DCWRD1 (CREATE 1 TES NIL)))
   (GO MV)
   FN (SET WRD1 (F2O. FNTRAP))
   (SET (BIT 42 6 WRD1) TEST)
   (SET TEST 12Q)
   MV (IF FR (GO FILL))
   (SET FR (TRIPLE))
   (SET (WORD2 (S2O. FR)) 0Q)
   (SET (CHAINS FR) (S2O. S))
   (SET (LINK (S2O. FR)) (CHAINS N))
   (SET (CHAINS N) (S2O. FR))
   FILL (IF DCWRD3 (SET (BIT 6 18 WRD3) (S2O. DCWRD3)))
   (IF DCWRD1 (BLOCK NIL (SET (BIT 0 18 WRD1)
      (PLUS 1 (S2O. DCWRD1)))
     (SET (BIT 24 18 WRD1) (S2O. DCWRD1))))
   (SET (WORD1 (S2O. FR)) WRD1)
   (SET (TCODES FR) WRD3)
   (SET (PREFIX (S2O. FR)) TEST)
   (GO MX)
   MS (BLOCK ((FG (GETFREE TYPE X)))
    (IF (NULL FG) (GO NOPDER))
    (IF (AND (EQN TYPE N) (EQN X S))
     (BLOCK NIL (SET (TCODES FR) (SYNTYPE FR)) (GO R)))
    (IF (NULL (SET TES (SYNGET FG FR))) (GO SYNER))
    (SET FG (FVLIST TES))
    (IF (SET FR (MAKEFREE N S (CAR FG) (CADR FG) (CADDR FG)))
     (SET (TCODES FR) (WORDOR 2Q8 (SHIFT (S2O. TES) 6))))
    (GO R)))
  KINDER (SET TES (CONS STOR (QUOTE (INVALID KIND))))
  (GO ERR)
  TYPERR (SET TES (CONS TYPE (QUOTE (INVALID TYPE))))
  (GO ERR)
  RDECER (SET TES (CONS (FVLIST FR)
    (APPEND (QUOTE (NOT CHANGED TO)) (LIST (LIST STOR TYPE X)))))
  (GO ERR)
  SYNER (SET TES (QUOTE (CIRCULAR SYNONYM)))
  (GO MNS)
  NOPDER (SET TES (QUOTE (NO PRIOR DECLARATION)))
  MNS (SET TES (APPEND (LIST STOR (CONS TYPE X)) TES))
  ERR (MESSAGE (CONS (CONS N S) TES))))
(DECLARE (KINDLIST SYMBOL OWN (QUOTE ((STET . 0Q)
    (MEANS . 2Q)
    (FREE . 11Q)
    (FLUID . 11Q)
    (OWN . 12Q)
    (FUNCTION . 21Q)
    (MACRO . 22Q) (INSTRUCTIONS . 23Q) (ROUTINE . 24Q)))))
(ROUTINE (SYNTYPE OCTAL) ((A SYMBOL)) (TCODES (SYNGET A NIL)))
(ROUTINE (SYNGET SYMBOL)
 ((A SYMBOL) (B SYMBOL))
 (BLOCK ((C OCTAL))
  L (IF (EQN A B) (RETURN NIL))
  (SET C (TCODES A))
  (IF (NQ (BIT 24 6 C) 2Q) (RETURN A))
```

```
    (SET A (O2S. (BIT 6 18 C))) (GO L)))
  (FUNCTION (MAKETYPE OCTAL)
   ((TYPE SYMBOL LEXICAL))
   (BLOCK ((J OCTAL LEXICAL 0Q))
    (BLOCK ((AA (ARRAY OCTAL) FREE)
      (NN INTEGER FREE 1)
      (RR INTEGER FREE 0)
      (PP INTEGER FREE 42)
      (WWL OCTAL FREE LOC J) (HERE BOOLEAN FREE TRUE))
     (TYPRDL TYPE)
     (RETURN (IF (EQ NN 1)
       (BIT 42 6 WWL)
       (LS NN 6)
       (BIT 18 30 WWL) (WORDOR 1Q8 (SHIFT (S2O. AA) 6)))))))
  (FUNCTION (TYPRDL NOVALUE)
   ((X SYMBOL LEXICAL))
   (IF (ATOM X)
    (BLOCK ((TC OCTAL LEXICAL 0Q))
     (IF (NUMBP X)
      (SET TC X)
      (SET X (FINDN X (QUOTE ((SYMBOL . 0Q)
          (BOOLEAN . 1Q)
          (OCTAL . 2Q)
          (INTEGER . 3Q)
          (REAL . 4Q)
          (FUNCTIONAL . 5Q)
          (ARRAY . 12Q1)
          (LOC . 11Q1) (NOVALUE . 37Q) (INDEF . 37Q)))))
      (SET TC (CDR X)) (GO R))
     (IF HERE (GO STUFF)
      (AND (EQ TC 77Q) (OR (EQ PP 0) (EQ NN 5))) (GO R))
     (IF (EQ (SET NN (PLUS NN 1)) 6)
      (SET AA (CREATE 1 (QUOTE OCTAL) WWL))
      (EQ PP 0Q)
      (BLOCK NIL (SET AA (NCONCS AA (CREATE 1 (QUOTE OCTAL) 0Q)))
       (SET PP 48)) (GO UPP))
     (SET RR (PLUS RR 1))
     (LOCSET WWL (AA RR))
     UPP (SET PP (PLUS PP -6))
     STUFF (SET HERE (IF (EQ (BIT 6 6 TC) 0Q) NIL TRUE))
     (SET WWL (WORDOR (SHIFT (BIT 0 6 TC) PP) WWL)) R)
    (BLOCK NIL (IF (NULL (CDR X))
      (GO ON)
      (TMODEP (CADR X))
      (SET X (REVERSE X))
      (EQ (CAR X) (QUOTE FUNCTIONAL))
      (SET X (APPEND (CONS 45Q (CDR X)) (QUOTE (77Q)))))
     ON (MAPCAR X TYPRDL)))))
(RETRIEVE (SECTION (LISP SYS) SYMBOL)
  (FUNCTION ALLDEC (X)
   (IF (NOT (IDP X))
    NIL (BLOCK ((Y (O2S. (BIT 24 18 (CORE (S2O. X))))) Z)
     (FOR Y (RESET Y (O2S. (BIT 0 18 (CORE (PLUS (S2O. Y) 1)))))
      (WHILE (NQ X Y)) (SET Z (CONS ((VARNAME . SYS) Y) Z)))
     (RETURN Z))))
  (FUNCTION FINDEC (N SN)
   (BLOCK ((T OCTAL (S2O. (GETFREE N SN))))
    (RETURN (IF (EQ T 0)
      NIL (LIST T (CORE (PLUS T -1))
       ((FVLIS1 . COMPIL) (O2S. T)))))))
  (SECTION SYS SYMBOL)
  (ROUTINE (GETFREE SYMBOL)
   ((N SYMBOL) (S SYMBOL))
   (BLOCK ((P OCTAL (CHAINS N)))
    L (IF (EQN (O2S. P) N)
     (RETURN NIL) (EQN (O2S. (CHAIN P)) S) (RETURN (O2S. P)))
    (SET P (LINK P)) (GO L)))
  (FUNCTION (FVLIST SYMBOL)
   ((S SYMBOL)) (IF S (CONS (FVKIND S) (GETYPE S)) NIL))
  (FUNCTION (FVKIND SYMBOL)
   ((S SYMBOL))
   (IF (EQ (BIT 42 6 (WORD3 (S2O. S))) 2Q)
    (QUOTE MEANS)
    (EQ (PREFIM (WORD2 (S2O. S))) 11Q)
    (IF (EQ (XMFLAG (S2O. S)) 0Q) (QUOTE FREE) (QUOTE FLUID))
    (EQ (WORDAND 1Q14 (WORD3 (S2O. S))) 0Q)
    (QUOTE OWN)
    (BLOCK ((X SYMBOL (FINDN (BIT 0 3 (PREFIM (WORD1 (S2O. S))))
        (QUOTE ((0Q . OWN)
          (1Q . FUNCTION)
          (2Q . MACRO) (3Q . INSTRUCTIONS) (4Q . ROUTINE))))))
     (IF X (RETURN (CDR X)) (RETURN (QUOTE UNKNOWN))))))
  (DECLARE (AA (ARRAY OCTAL))
   (NN INTEGER)
   (WWL OCTAL LOC) (RR INTEGER) (PP INTEGER) (WW OCTAL))
  (FUNCTION GETYPE (S)
   (IF (EQ (BIT 42 6 (WORD3 (S2O. S))) 2Q)
    (BLOCK ((Q SYMBOL (VARNAME (O2S. (BIT 24 18 (WORD3 (S2O. S))))))))
     (RETURN (LIST (CAR Q) (CDR Q))))
    (BLOCK ((WW OCTAL FREE (SYNTYPE S))
      (AA (ARRAY OCTAL) FREE)
      (PP INTEGER FREE 42) (RR INTEGER FREE 1))
     (IF (EQ (BIT 24 6 WW) 0Q)
      (SET PP 0) (EQ (BIT 24 6 WW) 1Q) (GO A) (SET PP 24))
     R (RETURN (RDTYPE))
     A (SET AA (O2S. (BIT 6 18 WW))) (SET WW (AA 1)) (GO R))))
  (FUNCTION RDTYPE NIL (DETYPE (RDTYPC)))
  (FUNCTION DETYPE ((TC OCTAL))
   (LIST (STANTP (DTYP TC))
    (IF (EQ (BIT 3 1 TC) 0Q) (QUOTE VALUE) (QUOTE LOC))))
  (FUNCTION DTYP ((TC OCTAL))
   (IF (EQ (BIT 5 1 TC) 0Q)
    (CDR (FINDN (WORDAND TC 27Q) STYPES))
    (BLOCK ((N OCTAL (RDTYPC)))
     (BLOCK ((J (LIST (IF (NQ N 37Q) (DTYP N) (QUOTE NOVALUE))
         (QUOTE FUNCTIONAL))))
      LOOP (SET N (RDTYPC))
      (IF (NQ N 77Q)
       (BLOCK NIL (SET J (CONS (IF (NQ N 37Q)
           (DETYPE N) (LIST (QUOTE INDEF) (RDTYPE))) J)) (GO LOOP)))
      (RETURN (REVERSE J))))))
  (ROUTINE (RDTYPC OCTAL)
   NIL (BLOCK ((TC OCTAL (BIT PP 6 WW)))
    (IF (NQ PP 0) (SET PP (PLUS PP -6)) AA (GO ARR) (GO B))
    RT (RETURN TC)
    B (SET WW (INVERT 0Q))
    R (SET PP 42)
    (GO RT)
    ARR (IF (LS (SET RR (PLUS RR 1)) (ARSIZE (S2O. AA)))
     (SET WW (AA RR)) (GO B)) (GO R))))
(TYPEQ (SECTION SYS SYMBOL)
  (FUNCTION (TYPEP BOOLEAN)
   ((J SYMBOL)) (OR (STYPEP J) (ATYPEP J) (FUNTYP J)))
  (FUNCTION (STYPEP BOOLEAN)
   ((J SYMBOL))
   (MEMBER J (QUOTE (BOOLEAN INTEGER OCTAL REAL SYMBOL))))
  (FUNCTION (ATYPEP BOOLEAN)
   ((J SYMBOL))
   (AND (NOT (ATOM J))
    (EQN (CAR J) (QUOTE ARRAY))
    (CDR J) (NULL (CDDR J)) (FTYPP (CADR J))))
  (FUNCTION (FUNTYP BOOLEAN)
   ((J SYMBOL))
   (AND (NOT (ATOM J))
    (EQN (CAR J) (QUOTE FUNCTIONAL))
    (SET J (CDR J))
    (VTYPEP (CAR J))
    (BLOCK NIL LOOP (IF (NULL (SET J (CDR J)))
      (RETURN TRUE)
      (AND (EQN (LENGTH J) 1) (INDEFP (CAR J)))
      (RETURN TRUE) (PTYPEP (CAR J)) (GO LOOP)))))
  (FUNCTION (FTYPP BOOLEAN)
   ((J SYMBOL)) (OR (STYPEP J) (EQN J (QUOTE FUNCTIONAL))))
  (FUNCTION (VTYPEP BOOLEAN)
   ((J SYMBOL)) (OR (FTYPP J) (EQN J (QUOTE NOVALUE))))
  (FUNCTION (INDEFP BOOLEAN) ((J SYMBOL)) NIL)
  (FUNCTION (PTYPEP BOOLEAN)
   ((J SYMBOL))
   (IF (ATOM J)
    (FTYPP J)
    (AND (FTYPP (CAR J))
     (OR (NULL (CDR J)) (AND (TMODEP (CADR J)) (NULL (CDDR J)))))))
  (FUNCTION (TMODEP BOOLEAN)
   ((J SYMBOL)) (MEMBER J (QUOTE (LOC VALUE))))
  (ROUTINE (FTYPER SYMBOL)
   ((TYPE SYMBOL))
   (IF (ATOM TYPE)
    TYPE (EQ (CAR TYPE) (QUOTE FUNCTIONAL))
    (QUOTE FUNCTIONAL) (QUOTE SYMBOL)))
  (FUNCTION STANTP (TYPE)
   (IF (NOT (FUNTYP TYPE))
    TYPE (CONS (CAR TYPE) (CADR TYPE) (MAPCAR (CDDR TYPE) TPFIX))))
  (FUNCTION TPFIX (A)
   (IF (ATOM A)
    (CONS A (QUOTE (VALUE)))
    (EQ (CAR A) (QUOTE INDEF))
    (LIST (QUOTE INDEF) (TPFIX (CADR A))) A)))
(TYPES (SECTION SYS SYMBOL)
  (ROUTINE (DFINIT OCTAL)
   ((TYPE SYMBOL))
   (IF (NQ TYPE (QUOTE FUNCTIONAL)) 0Q (F2O. FMTRAP)))
  (DECLARE (TYPMSG OWN (QUOTE (IS NOT LEGAL TYPE))))
  (FUNCTION (CONVRT OCTAL)
   ((TYPE SYMBOL) (VALUE SYMBOL))
   (IF (OR (EQ TYPE (QUOTE SYMBOL)) (EQ TYPE (QUOTE BOOLEAN)))
    (S2O. VALUE)
    (OR (EQ TYPE (QUOTE INTEGER)) (EQ TYPE (QUOTE OCTAL)))
    VALUE (EQ TYPE (QUOTE REAL))
    (R2O. VALUE)
    (EQ TYPE (QUOTE FUNCTIONAL)) (F2O. VALUE) (ERRMSG TYPE TYPMSG)))
  (DECLARE (STYPES SYMBOL OWN (QUOTE ((0Q . SYMBOL)
      (1Q . BOOLEAN)
      (2Q . OCTAL)
      (3Q . INTEGER)
      (4Q . REAL)
      (5Q . FUNCTIONAL)
      (6Q . STRING)
      (7Q . ID)
      (1Q1 . QUOTE)
      (11Q . FLUID)
      (12Q . OWN)
      (13Q . EMPTY)
      (2Q1 ARRAY SYMBOL)
      (21Q ARRAY BOOLEAN)
      (22Q ARRAY OCTAL)
      (23Q ARRAY INTEGER)
      (24Q ARRAY REAL) (25Q ARRAY FUNCTIONAL)))))
  (FUNCTION (STYPE SYMBOL)
   ((S SYMBOL))
   (BLOCK ((A SYMBOL (IF (SPACEP. S)
       (FINDN (PREFIX (S20. S)) STYPES) NIL)))
    (RETURN (IF A (CAR A) NIL)))))
(MAKEQUOTE (FUNCTION (MAKEQUOTE SYMBOL)
   ((S SYMBOL))
   (BLOCK ((TR SYMBOL (TRIPLE)))
    (SET (WORD3 (S20. TR)) 0Q)
    (SET (WORD2 (S20. TR)) 1000000000000001Q)
    (SET (WORD1 (S20. TR)) (S20. S)) (RETURN TR))))
(DEBUGER (SECTION (DEBUG SYS) SYMBOL)
  (FUNCTION ((ARTYPE . IO) SYMBOL) ((X SYMBOL)))
  (FUNCTION ((ARYCHK . DEBUG) INTEGER)
   ((A SYMBOL) (T SYMBOL) (S INTEGER))
   (IF (NOT (ARRAYP A))
    (S20. (ERROR (LIST A (QUOTE SUBSCRIPTED))))
    (NOT (EQN T ((ARTYPE . IO) A)))
    (S20. (ERROR (QUOTE (BAD TYPED ARRAY))))
    (OR (LQ S 0) (GQ S (BIT 24 18 (CORE (S20. A)))))
    (S20. (ERROR (QUOTE (OUT OF BOUNDS SUBSCRIPT)))) S))
  (FUNCTION ((ATMCHK . DEBUG) SYMBOL)
   ((X SYMBOL))
   (IF (ATOM X) (ERROR (CONS X (QUOTE (CAR OR CDR ED)))) X))
  (FUNCTION ((FUNCHK . DEBUG) NOVALUE)
   ((D SYMBOL) (F (FUNCTIONAL REAL)))
   (BLOCK ((E SYMBOL (CADR (FVLIST (C2S. (I20. (PLUS (BIT 0 18 (F2C.
           F)) 1))))))))
    (IF (NOT (OR (EQ (QUOTE NOVALUE) (CADR D))
       (EQN (CADR D) (CADR E))))
     (ERROR (QUOTE (FUNCTIONAL VALUE TYPE MISMATCH))))
    (IF (NOT (EQ (CDDR D) (CDDR E)))
     (ERROR (QUOTE (ARG OF FUNCTIONAL TYPE MISMATCH)))))))

****END OF FILE DETECTED
```

```
(SEC.LISP (SECTION LISP SYMBOL)
  (FUNCTION (ERROR SYMBOL) ((S SYMBOL)))
  (FUNCTION (PRETTYP SYMBOL) ((S SYMBOL)))
  (FUNCTION (FITATOM SYMBOL) ((S SYMBOL)))
  (FUNCTION (OPEN SYMBOL) ((FN SYMBOL) (DL SYMBOL)))
  (FUNCTION (SHUT SYMBOL) ((FN SYMBOL) (DL SYMBOL)))
  (FUNCTION (POSITION SYMBOL) ((F SYMBOL) (A INTEGER)))
  (FUNCTION (INPUT SYMBOL) ((X SYMBOL)))
  (FUNCTION (OUTPUT SYMBOL) ((X SYMBOL)))
  (FUNCTION (PRINT SYMBOL) ((X SYMBOL)))
  (FUNCTION (PRIN SYMBOL) ((X SYMBOL)))
  (FUNCTION (PRIN0 SYMBOL) ((X SYMBOL)))
  (FUNCTION (PRINATOM SYMBOL) ((A SYMBOL)))
  (FUNCTION (PRINSTRING SYMBOL) ((X SYMBOL)))
  (FUNCTION (SYMPRINT SYMBOL) ((X SYMBOL)))
  (FUNCTION (SYMPRIN SYMBOL) ((X SYMBOL)))
  (FUNCTION (PRINCH SYMBOL) ((X SYMBOL)))
  (FUNCTION (READCH SYMBOL) NIL)
  (FUNCTION READ NIL)
  (FUNCTION PRINARRAY (A))
  (FUNCTION (PRINWORD OCTAL) ((X OCTAL)))
  (FUNCTION (READWORD OCTAL) NIL)
  (FUNCTION (ENDIN NOVALUE) NIL)
  (FUNCTION (ENDINR NOVALUE) NIL)
  (FUNCTION (ENDOUT NOVALUE) NIL)
  (FUNCTION (ENDOUTR NOVALUE) NIL)
  (FUNCTION (ENDINP NOVALUE) NIL)
  (FUNCTION (ENDOUTP NOVALUE) NIL)
  (FUNCTION (NOP NOVALUE) NIL)
  (FUNCTION (NILF SYMBOL) NIL)
  (ROUTINE (CLEAR NOVALUE) ((FN SYMBOL)))
  (ROUTINE (GET SYMBOL) ((FN SYMBOL) (DL SYMBOL)))
  (ROUTINE (GETCHAR SYMBOL) ((A (ARRAY OCTAL)) (CC INTEGER))))
(ARSIZE (SECTION IO SYMBOL)
 MACRO1(((ARSIZE (LAMBDA (S) (LIST (QUOTE BIT) 24 18
    (CONS (QUOTE CORE) (CDR S))))))))
 (SEC.IO (SECTION (IO SYS) SYMBOL)
  (FUNCTION (CAR. SYMBOL) ((A SYMBOL)) (CAR A))
  (FUNCTION (LOGTTY NOVALUE) ((C INTEGER) (M SYMBOL)))
  (FUNCTION (B1 SYMBOL) ((S SYMBOL)))
  (FUNCTION (F1 SYMBOL) ((S SYMBOL)))
  (FUNCTION READ NIL)
  (FUNCTION (CHSUPL SYMBOL) NIL)
  (FUNCTION (TTYSPL SYMBOL) NIL)
  (FUNCTION (TOSTRG SYMBOL) ((A SYMBOL)))
  (FUNCTION (CVRTNM INTEGER) ((FN SYMBOL)))
  (FUNCTION (SEQNO NOVALUE) NIL)
  (FUNCTION (INTTY NOVALUE) NIL)
  (FUNCTION (INTAPE NOVALUE) NIL)
  (FUNCTION (INDISC NOVALUE) NIL)
  (FUNCTION (OUTTY NOVALUE) NIL)
  (FUNCTION (OUTAPE NOVALUE) NIL)
  (FUNCTION (OUTDISC NOVALUE) NIL)
  (FUNCTION (INTPAS NOVALUE) NIL)
  (FUNCTION (OUTPAS NOVALUE) NIL)
  (FUNCTION (INDCAS NOVALUE) NIL)
  (FUNCTION (OUTDCAS NOVALUE) NIL)
  (ROUTINE (MOVEI INTEGER) ((S INTEGER) (B OCTAL LOC)))
  (ROUTINE (MOVEO INTEGER) ((S INTEGER) (B OCTAL LOC)))
  (ROUTINE (MODIFY INTEGER) NIL)
  (FUNCTION (RDEC SYMBOL) ((NM INTEGER)))
  (ROUTINE (FDEC INTEGER)
   ((NM INTEGER)
    (UT INTEGER)
    (FM INTEGER)
    (SZ INTEGER) (RL INTEGER) (PK INTEGER) (PT INTEGER)))
  (ROUTINE (CVRTN1 INTEGER) ((S OCTAL LOC) (I INTEGER)))
  (ROUTINE (SETBUF NOVALUE) ((SINK OCTAL LOC) (K OCTAL)))
  (FUNCTION (T8X6 BOOLEAN)
   ((EOR OCTAL)
    (SOURCE OCTAL LOC) (SINK OCTAL LOC) (J INTEGER) (I INTEGER)))
  (ROUTINE (T8X12 NOVALUE) NIL)
  (ROUTINE (T12X8 INTEGER) NIL)
  (FUNCTION (TRANDC INTEGER) ((CH OCTAL) (J INTEGER)))
  (FUNCTION (TRANTP INTEGER) ((CH OCTAL) (J INTEGER)))
  (FUNCTION (T6X8 NOVALUE)
   ((COL INTEGER) (WHAT (FUNCTIONAL INTEGER OCTAL INTEGER))))
  (ROUTINE (T75 BOOLEAN) NIL)
  (ROUTINE (GETCHAR SYMBOL) ((A OCTAL LOC) (CC INTEGER)))
  (ROUTINE (SETCHAR SYMBOL) ((CH SYMBOL) (A OCTAL LOC) (CC INTEGER))))
(DSPCHR (LAP (FUNCTION ((NCP . LISP) NOVALUE)
    NIL (ORG)
    (BEGIN)
    (END)
    (RETURN)
    (ENTRY FIXBUF (LABEL BUF))
    (ENTRY DCALL (LABEL DEC))
    (ENTRY DNAME ((LABEL DEC) 2))
    (ENTRY DUNIT ((LABEL DEC) 3))
    (ENTRY DFORM ((LABEL DEC) 4))
    (ENTRY DSIZE ((LABEL DEC) 6))
    (ENTRY DREEL ((LABEL DEC) 8))
    (ENTRY DPROTK ((LABEL DEC) 9))
    (ENTRY DPOST ((LABEL DEC) 10))
    (ENTRY DSTAT ((LABEL DEC) 11))
    (ENTRY MCALL (LABEL MOOV))
    (ENTRY MNAME ((LABEL MOOV) 2))
    (ENTRY MINCUT ((LABEL MOOV) 3))
    (ENTRY MLOC ((LABEL MOOV) 5))
    (ENTRY MSECTR ((LABEL MOOV) 7))
    (ENTRY MWDSIN ((LABEL MOOV) 9))
    (ENTRY MSTAT ((LABEL MOOV) 10))
    (ENTRY MSIZE ((LABEL MOOV) 12))
    (ENTRY MPOST ((LABEL MOOV) 13))
    (ENTRY BELL (LABEL BELLS))
    (ENTRY TAPOS (LABEL POS))
    (ENTRY TNAME ((LABEL POS) 2))
    (ENTRY ACTION ((LABEL POS) 3))
    (ENTRY RESCUE (LABEL RES))
    (ENTRY DEFILE (LABEL DEF))
    (ENTRY DFNAME ((LABEL DEF) 2))
    (ENTRY DELETE (LABEL DEL))
    (ENTRY DLNAME ((LABEL DEL) 2))
    (ENTRY DLSTAT ((LABEL DEL) 3))
    (ENTRY REFILE (LABEL REF))
    (ENTRY RFNAME ((LABEL REF) 2))
    (ENTRY RWDSIN ((LABEL REF) 4))
    (ENTRY RFSTAT ((LABEL REF) 5))
    (ENTRY RFORM ((LABEL REF) 6))
    (ENTRY INSERT (LABEL INS))
    (ENTRY ISNAME ((LABEL INS) 2))
    (ENTRY INNAME ((LABEL INS) 4))
    (ENTRY INSTAT ((LABEL INS) 5))
    (ENTRY INSIZE ((LABEL INS) 7))
    (ENTRY MODFY (LABEL MOD))
    (ENTRY MDNAME ((LABEL MOD) 2))
```

```
(ENTRY MDSIZE ((LABEL MOD) 4))
(ENTRY MDSTAT ((LABEL MOD) 5))
(ENTRY DSPCHR 312Q)
(ENTRY IN (LABEL INK))
(ENTRY OUT (LABEL OUTK))
DEC (433162473131011 3Q)
(26314325600606Q)
(16363701171777 7Q)
(6445316360600C1Q1)
(264651446C60521Q)
(456444662462016Q1)
(1)
(512525436C600604Q)
(0)
(475146632542Q4)
(474662636O6Q5)
(634562632163Q4)
MOOV (433162473131011 4Q)
(444665256C600606Q)
(16363701171777 7Q)
(466463476463046Q1)
(234651253167O1Q2)
(0 ((LABEL BUF) 1))
(243162233167O1Q2)
(0)
(662462314560O1Q2)
(0)
(634562632163Q4)
(456444662462016Q1)
(0)
(474662636C6Q5)
BELLS (433162473131O105Q)
(444665256C600606Q)
(16363701171777 7Q)
(466463476463046Q1)
(234651253167016Q1)
(0 (LABEL BELL1))
BELL1 (700C700C3Q4)
POS (43316247313101O3Q)
(632147446465O606Q)
(0)
(464763314645Q4)
RES (43316247313101O2Q)
(512562236425O16Q1)
(0)
DEF (433162473131O1O2Q)
(242526314325O606Q)
(0)
DEL (433162473131O1O3Q)
(242543256325O606Q)
(0)
(634562632163Q4)
REF (433162473131O1O6Q)
(512526314325O606Q)
(0)
(662462314560O1Q2)
(0)
(634562632163Q4)
(264651446C6Q5)
INS (433162473131O1O7Q)
(314562255163O606Q)
(0)
(314545214425O606Q)
```

```
    (0)
    (634562632163Q4)
    (456444662462016Q1)
    (0)
    MOD (433162473131010 5Q)
    (444624312670 0606Q)
    (0)
    (456444662462 01Q2)
    (0)
    (634562632163Q4)
    BUF (6Q14 0. 0 520)
    (606060606060606Q1)
    (DITTO 519)
    INK (314547646360 04Q2) OUTK (466463476463 04Q2)) NIL LISP))
(CNVRTB (FUNCTION ((NILF . LISP) SYMBOL) NIL NIL)
  (SECTION IO SYMBOL)
  (DECLARE (CNVRTB (ARRAY OCTAL)
    OWN (QUOTE (*OCTAL 600100400006Q5 6001000100610001Q
     6001000200620002Q 7702000300630003Q 6001000400640004Q
     6001000500650005Q 6001000600660006Q 5201000700670007Q
     600100100070001Q1 6001001100710011Q 3201001200000012Q
     6001001300750013Q 6001001400470014Q 3202001500720015Q
     6001001600760016Q 6001001700430017Q 600100200053002Q1
     6001002101010021Q 6001002201020022Q 6001002301030023Q
     6001002401040024Q 6001002501050025Q 6001002601060026Q
     6001002701070027Q 600100300110003Q1 6001003101110031Q
     6001003200150032Q 6001003300560033Q 6001003400510034Q
     6001003500450035Q 6001003601340036Q 6001003701370037Q
     602200400055004Q1 6001004101120041Q 6001004201130042Q
     1704004301140043Q 5305004401150044Q 3521004501160045Q
     2001004601170046Q 1403004701200047Q 740500500121005Q1
     3405005101220051Q 5405005200070052Q 2012005300440053Q
     7323005400520054Q 4013005501350055Q 3316005600730056Q
     6111005701360057Q 1400600040006Q1 114006100570061Q
     214006201230062Q 314006301240063Q 414006401250064Q
     514006501260065Q 614006601270066Q 714006701300067Q
     101500700131007Q1 1115007101320071Q 1505007200770072Q
     5605007300540073Q 7611007400500074Q 1311007501330075Q
     1611007600740076Q 7201007700030077Q 600101000000014Q1
     2121010100000101Q 2221010200000102Q 2321010300000103Q
     2421010400000104Q 2517010500000105Q 2621010600000106Q
     2721010700000107Q 302101100000011Q1 3121011100000111Q
     4121011200000112Q 4221011300000113Q 4321011400000114Q
     4421011500000115Q 4521011600000116Q 4621011700000117Q
     472101200000012Q1 5020012100000121Q 5121012200000122Q
     6221012300000123Q 6321012400000124Q 6421012500000125Q
     6521012600000126Q 6621012700000127Q 672101300000013Q1
     7021013100000131Q 7121013200000132Q 7505013300000133Q
     3605013400000134Q 5505013500000135Q 5705013600000136Q
     3705013700000137Q 600101Q10 21010101Q8 22010102Q8 23010103Q8
     24010104Q8 25010105Q8 26010106Q8 27010107Q8 3001011Q9 31010111Q8
     41010112Q8 42010113Q8 43010114Q8 44010115Q8 45010116Q8
     46010117Q8 4701012Q9 50010121Q8 51010122Q8 62010123Q8 63010124Q8
     64010125Q8 65010126Q8 66010127Q8 6701013Q9 70010131Q8 71010132Q8
     75010133Q8 6001004Q9 55010135Q8 6001004Q9 60010177Q8)))))
(VARIABLES (SECTION SYS SYMBOL)
  (DECLARE (CHO OCTAL OWN)
   (XXDLIM SYMBOL OWN (QUOTE '.))
   (XXCHAR SYMBOL OWN)
   (GNMODE BOOLEAN FLUID NIL)
   (PRMODE BOOLEAN FLUID NIL) (OTTY SYMBOL OWN) (ITTY SYMBOL OWN))
  (SECTION LISP SYMBOL)
  (DECLARE (TTY. SYMBOL FLUID (QUOTE ((UNIT . TTY)
      (FORM . ASCII) (RECORD . 1) (HORIZONTAL 1 73 72))))                      0025300
    (TAPE. SYMBOL FLUID (QUOTE ((UNIT . TAPE)                                  0025400
      (FORM . BCD)                                                             0025500
      (RECORD . 30) (HORIZONTAL 1 73 80) (VERTICAL 1 51 50))))                 0025600
    (DISC. SYMBOL FLUID (QUOTE ((UNIT . DISC)                                  0025700
      (FORM . BCD)                                                             0025800
      (RECORD . 51) (HORIZONTAL 1 73 80) (VERTICAL 1 51 50))))                 0025900
    (CORE. SYMBOL FLUID (QUOTE ((UNIT . CORE)                                  0026000
      (FORM . ASCII) (RECORD . 1))))                                           0026100
    (CRT. SYMBOL FLUID (QUOTE ((UNIT . CRT)                                    0026200
      (FORM . BINARY) (RECORD . 680))))                                        0026300
    (SKIPR. INTEGER OWN 1)                                                     0026400
    (SKIPF. INTEGER OWN 2)                                                     0026500
    (WEOF. INTEGER OWN 3)                                                      0026600
    (WEOT. INTEGER OWN 4)                                                      0026700
    (REWIND. INTEGER OWN 5)                                                    0026800
    (BACKR. INTEGER OWN 6)                                                     0026900
    (BACKF. INTEGER OWN 7) (KEY. INTEGER OWN 8))                               0027000
   (SECTION (IO SYS) SYMBOL)                                                   0027100
   (DECLARE (DDSW INTEGER OWN)                                                 0027200
    (CURCOL INTEGER FLUID LOC)                                                 0027300
    (ICURCOL INTEGER FLUID LOC)                                                0027400
    (CURLINE INTEGER FLUID LOC)                                                0027500
    (ICURLINE INTEGER FLUID LOC)                                               0027600
    (SUMLINE INTEGER FLUID LOC)                                                0027700
    (ISUMLINE INTEGER FLUID LOC)                                               0027800
    (LMG INTEGER FLUID LOC)                                                    0027900
    (ILMG INTEGER FLUID LOC)                                                   0028000
    (RMG INTEGER FLUID LOC)                                                    0028100
    (IRMG INTEGER FLUID LOC)                                                   0028200
    (MAXCOL INTEGER FLUID LOC)                                                 0028300
    (IMAXCOL INTEGER FLUID LOC)                                                0028400
    (TOP INTEGER FLUID LOC)                                                    0028500
    (ITOP INTEGER FLUID LOC)                                                   0028600
    (BOT INTEGER FLUID LOC)                                                    0028700
    (IBOT INTEGER FLUID LOC)                                                   0028800
    (PAGE INTEGER FLUID LOC)                                                   0028900
    (IPAGE INTEGER FLUID LOC)                                                  0029000
    (RECORD INTEGER FLUID LOC)                                                 0029100
    (IRECORD INTEGER FLUID LOC)                                                0029200
    (SIZE INTEGER FLUID LOC)                                                   0029300
    (ISIZE INTEGER FLUID LOC)                                                  0029400
    (COUNT INTEGER FLUID LOC)                                                  0029500
    (ICOUNT INTEGER FLUID LOC)                                                 0029600
    (MAXSEC INTEGER FLUID LOC)                                                 0029700
    (IMAXSEC INTEGER FLUID LOC)                                                0029800
    (SECTOR INTEGER FLUID LOC)                                                 0029900
    (ISECTOR INTEGER FLUID LOC)                                                0030000
    (STATUS INTEGER FLUID LOC)                                                 0030100
    (ISTATUS INTEGER FLUID LOC)                                                0030200
    (TTYMAX INTEGER FLUID LOC)                                                 0030300
    (NAME INTEGER FLUID LOC)                                                   0030400
    (INAME INTEGER FLUID LOC)                                                  0030500
    (BUFLOC (ARRAY OCTAL) FLUID)                                               0030600
    (IBUFLOC (ARRAY OCTAL) FLUID)                                              0030700
    (LINELOC OCTAL FLUID LOC)                                                  0030800
    (ILINELOC OCTAL FLUID LOC)                                                 0030900
    (FIXLOC OCTAL FLUID LOC)                                                   0031000
    (CURFN SYMBOL FLUID)                                                       0031100
    (ICURFN SYMBOL FLUID)                                                      0031200
    (RMGO (FUNCTIONAL NOVALUE) FLUID LOC)                                      0031300
    (IRMGO (FUNCTIONAL NOVALUE) FLUID LOC)                                     0031400
    (BOTO (FUNCTIONAL NOVALUE) FLUID LOC)                                      0031500
   (IBOTO (FUNCTIONAL NOVALUE) FLUID LOC)
   (MOVE (FUNCTIONAL NOVALUE) FLUID LOC)
   (IMOVE (FUNCTIONAL NOVALUE) FLUID LOC)
   (XXFUNC (FUNCTIONAL SYMBOL) FLUID LOC)
   (KEY (FUNCTIONAL NOVALUE) FLUID LOC)
   (IKEY (FUNCTIONAL NOVALUE) FLUID LOC)
   (XXSAVE SYMBOL FLUID LOC)
   (FILES. SYMBOL FLUID NIL)
   (WPL INTEGER FLUID LOC)
   (IWPL INTEGER FLUID LOC)
   (CPW INTEGER FLUID 6)
   (SHORWD SYMBOL FLUID (QUOTE (*STRING A.WORD)))
   (MSG1 SYMBOL OWN (QUOTE (*STRING 'R 'E 'D 'U 'N 'D 'A 'N 'T '  'F
     'I 'L 'E '  'N 'A 'M 'E ' )))
   (MSG2 SYMBOL OWN (QUOTE (*STRING 'B 'A 'D '  'O 'P 'E 'N '  'A 'R
     'G '2 ' )))
   (MSG3 SYMBOL OWN (QUOTE (*STRING 'T 'S 'S '  'R 'E 'J 'E 'C 'T ' ))
   )
   (MSG4 SYMBOL OWN (QUOTE (*STRING 'F 'I 'L 'E '  'N 'O 'T '  'O 'P
     'E 'N 'E 'D ' )))
   (MSG5 SYMBOL OWN (QUOTE (*STRING 'N 'O 'T '  'B 'I 'N 'A 'R 'Y '
     'F 'I 'L 'E ' )))
   (MSG6 SYMBOL OWN (QUOTE (*STRING 'I 'L 'L 'E 'G 'A 'L '  'U 'N 'I
     'T ' )))
   (MSG7 SYMBOL OWN (QUOTE (*STRING 'T 'A 'P 'E '  'X 'F 'E 'R ' )))
   (MSG8 SYMBOL OWN (QUOTE (*STRING 'F 'I 'L 'E '  'L 'O 'C 'K 'E 'D
     ' ))))
  (SECTION (IO SYS) SYMBOL) (DECLARE (BUFIX (ARRAY OCTAL) FLUID)))
(D.CALLS (SECTION (IO LISP SYS) SYMBOL)
  (ROUTINE (MODIFY INTEGER)
   NIL (BLOCK NIL (SET (CORENTRY MDNAME) NAME)
    (SET (CORENTRY MDSIZE) (TIMES 512 (PLUS 9 MAXSEC)))
    (CODE (LDA (ENTRY MODFY) (RA R)) (BUC (ENTRY DSPCHR)))
    (RETURN (BIT 0 6 (CORENTRY MDSTAT)))))
  (DECLARE (TO (ARRAY OCTAL)
    OWN (QUOTE (*OCTAL 22Q 3Q1 23Q 21Q 22Q 22Q))))
  (FUNCTION (RDEC SYMBOL)
   ((NM INTEGER))
   (BLOCK NIL (SET (CORENTRY RFNAME) NM)
    (CODE (LDA (ENTRY REFILE) (RA R)) (BUC (ENTRY DSPCHR)))
    (RETURN (LIST (BIT 0 6 (CORENTRY RFSTAT))
      (TO (PLUS 1 (BIT 0 6 (CORENTRY RFORM)))) (CORENTRY RWDSIN)))))
  (ROUTINE (FDEC INTEGER)
   ((NM INTEGER)
    (UT INTEGER)
    (FM INTEGER)
    (SZ INTEGER) (RL INTEGER) (PK INTEGER) (PT INTEGER))
   (BLOCK NIL (SET (BIT 0 6 (CORENTRY DPOST)) PT)
    (SET (BIT 0 6 (CORENTRY DPROTK)) PK)
    (SET (CORENTRY DREEL) RL)
    (SET (CORENTRY DSIZE) SZ)
    (SET (BIT 0 6 (CORENTRY DFORM)) FM)
    (SET (BIT 0 6 (CORENTRY DUNIT)) UT)
    (SET (CORENTRY DNAME) NM)
    (CODE (LDA (ENTRY DCALL) (RA R)) (BUC (ENTRY DSPCHR)))
    (RETURN (BIT 0 6 (CORENTRY DSTAT))))))
 PEN (SECTION (IO LISP SYS) SYMBOL)
  (FUNCTION (IUNLOCK NOVALUE) NIL (POSITION ICURFN KEY.))
  (FUNCTION (OUNLOCK NOVALUE) NIL (POSITION CURFN KEY.))
  (FUNCTION ((OPEN . LISP) SYMBOL)
   ((FN SYMBOL) (DL SYMBOL))
   (BLOCK NIL (IF (GET FN FILES.) (RETURN (ERROR MSG1)))
    (BLOCK ((IA (ARRAY INTEGER) (CREATE 18 (QUOTE INTEGER) 0))
         (FA (ARRAY FUNCTIONAL) (CREATE 8 (QUOTE FUNCTIONAL) NOP))       0037900
         (SA (ARRAY SYMBOL) (CREATE 2 (QUOTE SYMBOL) NIL))               0038000
         (U SYMBOL (GET (QUOTE UNIT) DL))                                0038100
         (F SYMBOL (GET (QUOTE FORM) DL))                                0038200
         (R SYMBOL (GET (QUOTE RECORD) DL))                              0038300
         (H SYMBOL (GET (QUOTE HORIZONTAL) DL))                          0038400
         (V SYMBOL (GET (QUOTE VERTICAL) DL))                            0038500
         (O SYMBOL (GET (QUOTE OVERFLOW) DL))                            0038600
         (X SYMBOL)                                                      0038700
         (T INTEGER 1)                                                   0038800
         (Y INTEGER 0)                                                   0038900
         (Z OCTAL 22Q)                                                   0039000
         (W INTEGER 1) (PROTEC SYMBOL (GET (QUOTE PROTECT) DL)))         0039100
        (IF (NULL (AND U F R H)) (RETURN (ERROR MSG2)))                  0039200
        (SET (IA 1) (CADR H))                                            0039300
        (SET (IA 2) (CADDR H))                                           0039400
        (SET (IA 3) (CADDDR H))                                          0039500
        (IF V (BLOCK NIL (SET (IA 4) (CADR V))                           0039600
          (SET (IA 5) (CADDR V)) (SET (IA 6) (CADDDR V)))                0039700
         (BLOCK NIL (SET (IA 4) 1) (SET (IA 5) 51) (SET (IA 6) 50)))     0039800
        (SET (IA 7) (CDR R))                                             0039900
        (SET (IA 8)                                                      0040000
         (IF (EQ (CDR F) (QUOTE BINARY))                                 0040100
          1 (PLUS 1 (IQUOTIENT (PLUS -1 (IA 3)) 6))))                    0040200
        TO (SET (IA 9) 1)                                                0040300
        (SET (IA 10) 1)                                                  0040400
        (SET (IA 11) 0)                                                  0040500
        (SET (IA 12) (CVRTNM FN))                                        0040600
        (SET (IA 13) 1)                                                  0040700
        (SET (IA 14) 0)                                                  0040800
        (SET (FA 6) CHSUPL)                                              0040900
        (IF (EQ (SET X (CDR U)) (QUOTE TTY))                             0041000
         (BLOCK NIL (IF (GR (IA 3) 72)                                   0041100
           (BLOCK NIL (SET (IA 3) 72) (SET (IA 2) 73)))                  0041200
          (SET (IA 7) 1)                                                 0041300
          (SET (FA 4) OUTTY)                                             0041400
          (SET (FA 5) INTTY)                                             0041500
          (SET (FA 6) TTYSPL) (SET Y (FDEC (IA 12) 8 21Q 19 0 0 0)))     0041600
         (EQ X (QUOTE TAPE))                                             0041700
         (BLOCK NIL (IF (NOT (EQ (CDR F) (QUOTE BCD)))                   0041800
           (BLOCK NIL (SET (FA 4) OUTPAS)                                0041900
            (SET (FA 5) INTPAS) (GO T21)))                               0042000
          (IF (OR (GR (IA 3) 120)                                        0042100
            (GR (TIMES (IA 7)                                            0042200
              (PLUS 1 (IQUOTIENT (PLUS -1 (IA 3)) 8))) 300))             0042300
           (BLOCK NIL (SET (IA 2) 81)                                    0042400
            (SET (IA 3) 80) (SET (IA 7) 30) (SET (IA 8) 14)))            0042500
          (SET (FA 4) OUTAPE)                                            0042600
          (SET (FA 5) INTAPE)                                            0042700
          (SET Z 23Q)                                                    0042800
          (SET T (PLUS 1 (IQUOTIENT (PLUS -1 (IA 3)) 8)))                0042900
          T21 (IF (SET X (GET (QUOTE REEL) DL)) (SET Y (CDR X)))         0043000
          (IF (AND PROTEC (MEMBER (QUOTE WRITE) (CDR PROTEC)))           0043100
           (SET W 0))                                                    0043200
          (SET Y (FDEC (IA 12) 3 Z (TIMES (IA 7) T) (CVRTNM Y) W 0)))    0043300
         (EQ X (QUOTE DISC))                                             0043400
         (BLOCK NIL (SET (IA 3) 80)                                      0043500
          (IF (SET X (GET (QUOTE DISC) DL))                              0043600
           (SET (IA 12) (CVRTNM (CDR X))))                               0043700
          (IF (EQ (CDR F) (QUOTE BCD)) (GO T31))                         0043800
          (SET (FA 4) OUTDCAS)                                           0043900
          (SET (FA 5) INDCAS)                                            0044000
          (GO T32)                                                       0044100
```

```
       T31 (IF (GR (IA 7) 51) (SET (IA 7) 51))
       (SET Z 23Q)
       (SET (IA 8) 14)
       (SET (FA 4) OUTDISC)
       (SET (FA 5) INDISC)
       T32 (IF (MEMBER (QUOTE OLD) DL) (GO T33))
       (SET (IA 15) 7)
       (SET Y (FDEC (IA 12) 11 Z 4096 0 0 0))
       (GO ALL)
       T33 (IF (GR (CAR (SET X (RDEC (IA 12)))) 3)
        (BLOCK NIL (IF (OR (GR (CAR X) 4) (GR T 100))
          (RETURN (ERROR MSG3))) (SET T (PLUS T 1)) (GO T33)))
       (IF (NQ Z (CADR X)) (RETURN (ERROR MSG2)))
       (SET (IA 15) (IQUOTIENT (PLUS (CADDR X) -1) 512)) (GO A1))
      (EQ X (QUOTE CORE))
      (RETURN (ERROR MSG6))
      (EQ X (QUOTE CRT))
      (RETURN (ERROR MSG6))
      (RETURN (ERROR (QUOTE (NOT A UNIT)))))
     ALL (IF (GR Y 3) (RETURN (ERROR MSG3)))
     A1 (IF (GR (IA 1) (IA 3)) (SET (IA 1) 1))
     (IF (GR (IA 4) (IA 6)) (SET (IA 4) 1))
     (SET (IA 18) 2Q15)
     (IF O (BLOCK NIL (SET (FA 3) (SET (FA 1) (CADR O)))
       (SET (FA 2) (CADDR O))
       (SET (FA 7) (CADDDR O)) (SET (FA 8) (CADDDDR O)))
      (BLOCK NIL (SET (FA 1) ENDIN)
       (SET (FA 3) ENDOUT)
       (SET (FA 7) IUNLOCK) (SET (FA 8) OUNLOCK)))
     (SET FILES. (CONS (CONS FN (APPEND (LIST (CONS (QUOTE BUF)
           (CREATE (PLUS 1 (TIMES (IA 7) (IA 8)))
            (QUOTE OCTAL) 1002004010020040Q1))
          (CONS (QUOTE SCA) SA)
          (CONS (QUOTE ICA) IA) (CONS (QUOTE FCA) FA)) DL)) FILES.))
     (IF PROTEC (SET (CAR FILES.)
       (CONS (CAAR FILES.) (CONS PROTEC (CDAR FILES.)))))
     (SET (IA 17) 100) (RETURN (MAPCAR FILES. CAR.))))))
(SHUT (SECTION (IO LISP SYS) SYMBOL)
  (FUNCTION ((SHUT . LISP) SYMBOL)
   ((A SYMBOL) (B SYMBOL))
   (BLOCK ((X SYMBOL (FILES. . IO)) (Y SYMBOL NIL))
    A (IF (NULL X)
     (RETURN (CONS A (QUOTE (NOT A FILE)))) (EQ A (CAAR X)) (GO B))
    (SET X (CDR (SET Y X)))
    (GO A)
    B (BLOCK ((FL SYMBOL (CAR X)))
     (IF Y (SET (CDR Y) (CDR X)) (SET FILES. (CDR FILES.)))
     (IF (EQ A (CURFN . IO)) (OUTPUT (OTTY . SYS)))
     (IF (EQ A (ICURFN . IO)) (INPUT (ITTY . SYS)))
     (BLOCK ((I (ARRAY INTEGER) (CDR (GET (QUOTE ICA) FL)))
       (NM INTEGER) (W OCTAL) (INS BOOLEAN NIL))
      (SET NM (I 12))
      (IF (OR (EQ (SET X (CDR (GET (QUOTE UNIT) FL))) (QUOTE TTY))
        (EQ X (QUOTE TAPE)))
       (BLOCK NIL (SET W (ENTRY DEFILE)) (GO C))
       (NQ X (QUOTE DISC))
       (RETURN (CONS A (QUOTE (IS CRT OR CORE))))
       (EQ (CDR (GET (QUOTE FILE) B)) (QUOTE DELETE))
       (BLOCK NIL (SET W (ENTRY DELETE)) (GO C)))
      (SET W (ENTRY INSERT))
      (SET INS TRUE)
      (SET (CORENTRY INNAME)
       (IF (SET Y (GET (QUOTE NAME) B))
        (I2O. (CVRTNM (CDR Y))) (I2O. NM)))
      (SET (CORENTRY INSIZE) (I2O. (TIMES 512 (PLUS (I 15) 1))))
      C (SET (CORE (PLUS 2 W)) (I2O. NM))
      D (CODE (LCA W) (BUC (ENTRY CSPCHR)))
      (IF (OR (NOT INS) (LQ (CORENTRY INSTAT) 3)) (GO LOG))
      (SET INS FALSE)
      (GO C)
      LOG (IF (SET Y (GET (QUOTE LOG) B)) (LOGTTY 9 (CDR Y)))))
    (RETURN (MAPCAR FILES. CAR.)))))
(IN.OUT (SECTION (IO LISP SYS) SYMBOL)
  (FUNCTION ((OUTPUT . LISP) SYMBOL)
   ((FN SYMBOL))
   (IF (EQN FN CURFN)
    FN (BLOCK ((DL SYMBOL (GET FN FILES.)))
     (IF (NULL DL) (RETURN (ERROR MSG4)))
     (BLOCK ((IA (ARRAY INTEGER) (CDR (GET (QUOTE ICA) (CDR DL))))
       (FA (ARRAY FUNCTIONAL) (CDR (GET (QUOTE FCA) (CDR DL))))
       (X SYMBOL (GET (QUOTE PROTECT) (CDR DL))) (Y INTEGER 0))
      (IF (AND X (MEMBER (QUOTE WRITE) (CDR X))) (RETURN NIL))
      (SET BUFLOC (CDR (GET (QUOTE BUF) (CDR DL))))
      (LOCSET LMG (IA 1))
      (LOCSET RMG (IA 2))
      (LOCSET MAXCOL (IA 3))
      (LOCSET TOP (IA 4))
      (LOCSET BOT (IA 5))
      (LOCSET PAGE (IA 6))
      (LOCSET RECORD (IA 7))
      (LOCSET WPL (IA 8))
      (LOCSET CURCOL (IA 9))
      (LOCSET CURLINE (IA 10))
      (LOCSET SUMLINE (IA 11))
      (LOCSET NAME (IA 12))
      (LOCSET STATUS (IA 13))
      (LOCSET SECTOR (IA 14))
      (LOCSET MAXSEC (IA 15))
      (LOCSET SIZE (IA 16))
      (LOCSET COUNT (IA 17))
      (LOCSET RMGO (FA 3))
      (LOCSET BOTO (FA 2))
      (LOCSET MOVE (FA 4))
      (LOCSET KEY (FA 8))
      (IF (GR SUMLINE 1) (SET Y (TIMES (PLUS -1 SUMLINE) WPL)))
      (LOCSET LINELOC (BUFLOC Y))
      (SET X CURFN) (SET CURFN FN) (RETURN X)))))
  (FUNCTION ((INPUT . LISP) SYMBOL)
   ((FN SYMBOL))
   (IF (EQN FN ICURFN)
    FN (BLOCK ((DL SYMBOL (GET FN FILES.)))
     (IF (NULL DL) (RETURN (ERROR MSG4)))
     (BLOCK ((IA (ARRAY INTEGER) (CDR (GET (QUOTE ICA) (CDR DL))))
       (FA (ARRAY FUNCTIONAL) (CDR (GET (QUOTE FCA) (CDR DL))))
       (SA (ARRAY SYMBOL) (CDR (GET (QUOTE SCA) (CDR DL))))
       (X SYMBOL (GET (QUOTE PROTECT) (CDR DL))) (Y INTEGER 0))
      (IF (AND X (MEMBER (QUOTE READ) (CDR X))) (RETURN NIL))
      (SET IBUFLOC (CDR (GET (QUOTE BUF) (CDR DL))))
      (LOCSET ILMG (IA 1))
      (LOCSET IRMG (IA 2))
      (LOCSET IMAXCOL (IA 3))
      (LOCSET ITOP (IA 4))
      (LOCSET IBOT (IA 5))
      (LOCSET IPAGE (IA 6))
      (LOCSET IRECORD (IA 7))
      (LOCSET IWPL (IA 8))
    NIL (BLOCK NIL (CLEAR ICONTN)
      (LOCSET ICURCOL (IA 9))
      (LOCSET ICURLINE (IA 10))
      (LOCSET ISUMLINE (IA 11))
      (LOCSET INAME (IA 12))
      (LOCSET ISTATUS (IA 13))
      (LOCSET ISECTOR (IA 14))
      (LOCSET IMAXSEC (IA 15))
      (LOCSET ISIZE (IA 16))
      (LOCSET ICOUNT (IA 17))
      (LOCSET TTYMAX (IA 18))
      (LOCSET IRNGC (FA 1))
      (LOCSET IBCTC (FA 2))
      (LOCSET IMOVE (FA 5))
      (LOCSET XXFUNC (FA 6))
      (LOCSET IKEY (FA 7))
      (LOCSET XXSAVE (SA 1))
      (IF (GR ISUMLINE 1) (SET Y (TIMES (PLUS -1 ISUMLINE) IWPL)))
      (LOCSET ILINELOC (IBUFLOC Y))
      (SET X ICURFN) (SET ICURFN FN) (RETURN X))))))
(POSITION (SECTION (IO LISP SYS) SYMBOL)
  (FUNCTION ((POSITION . LISP) SYMBOL)
   ((F SYMBOL) (A INTEGER))
   (BLOCK ((DL SYMBOL (GET F FILES.)))
    (IF (NULL DL)
     (RETURN (ERROR MSG4)) (OR (LS A 1) (GR A 8)) (RETURN NIL))
    (BLOCK ((U SYMBOL (GET (QUOTE UNIT) (CDR DL))))
     (IF (NULL U)
      (RETURN (ERROR MSG6))
      (EQ (CDR U) (QUOTE TAPE))
      (GO TO) (EQ (CDR U) (QUOTE DISC)) (GO DO) (RETURN NIL))
     TO (BLOCK ((N INTEGER 1))
      (IF (LS A 3) (SET DL (INPUT F)) (SET DL (OUTPUT F)))
      (CASE A (GO T1)
       (GO T2) (GO T3) (GO T4) (GO T5) (GO T6) (GO T7) (GO T8))
      T1 (SET N 0)
      T2 (BLOCK ((X INTEGER (TIMES IRECORD (PLUS 1 (IQUOTIENT (PLUS
            -1 IMAXCOL) 8)))) (S INTEGER 0))
       T21 (SET S (MOVEI X FIXLOC))
       (IF (OR (EQ N 0) (GQ S 4)) (GO T22))
       (SET N (PLUS N 1))
       (GO T21)
       T22 (SET ISUMLINE 0)
       (IF (LS S 4)
        (SET U 1)
        (GR S 5)
        (BLOCK NIL (INPUT DL) (RETURN (ERROR MSG7)))
        (LS N 2)
        (SET U (IF (EQ S 4) (QUOTE EOF) (QUOTE EOT)))
        (SET U (PLUS N -1))) (INPUT DL) (RETURN U))
      T5 (SET COUNT 100)
      T3 T4 T6 T7 (SET (CORENTRY TNAME) NAME)
      (SET (BIT 0 6 (CORENTRY ACTION)) (PLUS A 1))
      (CODE (LCA (ENTRY TAPOS) (R L567.7)) (BUC (ENTRY DSPCHR)))
      T8 (SET SUMLINE 0) (SET STATUS 1) (OUTPUT DL) (RETURN F))
     DO (BLOCK ((N INTEGER 1))
      (IF (OR (EQ A 3) (EQ A 4))
       (SET DL (OUTPUT F)) (SET DL (INPUT F)))
      (SET U F)
      (CASE A (GO D1)
       (GO D2) (GO D3) (GO D4) (GO D5) (GO D6) (GO D7) (GO D8))
      D1 (IF (EQ ISECTOR IMAXSEC) (SET N 0))
      (GO D21)
      D2 (SET N (DIFFERENCE IMAXSEC ISECTOR))
      D21 (IF (GR N 0) (SET U N) (SET U (QUOTE EOF)))                      C063100
      (SET ISECTOR (PLUS ISECTOR N))                                       C063200
      (GO D8)                                                              C063300
      D3 D4 (PRINCH (OCT2CH 34Q))                                          C063400
      (ENDOUT)                                                             C063500
      (IF (NQ SUMLINE 1) (ENDOUTR))                                        C063600
      (OUTPUT DL)                                                          C063700
      (RETURN U)                                                           C063800
      D6 (IF (GR ISECTOR 1)                                                C063900
       (BLOCK NIL (SET ISECTOR (PLUS ISECTOR -1)) (GO D8)))                C064000
      D5 D7 (SET ICOUNT 100)                                               C064100
      (SET ISECTOR 0)                                                      C064200
      D8 (SET ISUMLINE 0) (SET ISTATUS 1) (INPUT DL) (RETURN U))))))       C064300
(READF (SECTION (IO LISP SYS) SYMBOL)                                      C064400
  (FUNCTION ((ENDIN . LISP) NOVALUE)                                       C064500
   NIL (BLOCK ((INDEX INTEGER 0))                                          C064600
    (IF (EQ ISUMLINE 0) (BLOCK NIL (ENDINR) (GO E2)))                      C064700
    (SET ISUMLINE (PLUS ISUMLINE 1))                                       C064800
    (SET ISTATUS 1)                                                        C064900
    (IF (GR ISUMLINE IRECORD) (BLOCK NIL (SET ISUMLINE 0) (GO E1)))        C065000
    (SET ICURCOL ILMG)                                                     C065100
    (IF (GR ISUMLINE 1) (SET INDEX (TIMES (PLUS -1 ISUMLINE) IWPL)))       C065200
    (LOCSET ILINELOC (IBUFLOC INDEX))                                      C065300
    E2 (SET ICURLINE (PLUS ICURLINE 1))                                    C065400
    R1 (IF (EQ ICURLINE IBOT) (IBOTC))                                     C065500
    (IF (GR ICURLINE IPAGE) (ENDINP)) E1))                                 C065600
  (FUNCTION ((ENDINR . LISP) NOVALUE)                                      C065700
   NIL (BLOCK NIL (CLEAR ICURFN)                                           C065800
    (SET ISTATUS 2)                                                        C065900
    (IMOVE)                                                                C066000
    (SET ISUMLINE 1)                                                       C066100
    (SET ICURCOL ILMG)                                                     C066200
    (LOCSET ILINELOC (IBUFLOC 0))                                          C066300
    (IF (GQ ISTATUS 4) (SETCHAR (OCT2CH 34Q) ILINELOC ICURCOL))))          C066400
  (FUNCTION ((ENDINP . LISP) NOVALUE)                                      C066500
   NIL (BLOCK ((X INTEGER (PLUS ITOP IPAGE (MINUS ICURLINE)))              C066600
     (Y INTEGER 0))                                                        C066700
    (FOR Y (STEP 1 1 GR X)                                                 C066800
     (BLOCK NIL (ENDIN) (IF (GR ISTATUS 2) (GO T1))))                      C066900
    T1 (SET ICURLINE ITOP)))                                               C067000
  (FUNCTION ((READWORD . LISP) OCTAL)                                      C067100
   NIL (BLOCK ((X OCTAL))                                                  C067200
    (IF (EQ 6 IWPL)                                                        C067300
     (ERROR MSG5)                                                          C067400
     (EQ ISTATUS 4)                                                        C067500
     (IKEY)                                                                C067600
     (EQ ISUMLINE 0)                                                       C067700
     (ENDINR)                                                              C067800
     (GR ISUMLINE ISIZE) (ENDINR) (EQ ISTATUS 4) (RETURN X))               C067900
    (SET X (IBUFLOC ISUMLINE))                                             C068000
    (SET ISUMLINE (PLUS 1 ISUMLINE)) (RETURN X)))                          C068100
  (FUNCTION ((READCH . LISP) SYMBOL)                                       C068200
   NIL (BLOCK ((SW INTEGER 1) (X SYMBOL (OCT2CH 0Q)))                      C068300
    (IF XXSAVE (BLOCK NIL (SET X XXSAVE) (SET XXSAVE NIL) (GO BB))         C068400
     (EQ ISTATUS 4)                                                        C068500
     (IF (EQ 0 ISUMLINE)                                                   C068600
      (BLOCK NIL (IKEY) (ENDINR))                                          C068700
      (BLOCK NIL (SET ISUMLINE 0) (SET X (OCT2CH 34Q)) (GO BB))))          C068800
    (IF (EQ ISUMLINE 0) (ENDINR))                                          C068900
    (IF (EQ ICURCOL IRMG)                                                  C069000
     (BLOCK NIL (SET ISTATUS 0)                                            C069100
      (IRMGO) (IF (GR ISTATUS 0) (SET SW 2))))                             C069200
    (IF (GR ICURCOL IMAXCOL) (BLOCK NIL (SET SW 2) (ENDIN)))               C069300
   (CASE SW (GO AA) (GO BB))
   AA (SET X ((GETCHAR . IO) ILINELOC ICURCOL))
   (SET ICURCOL (PLUS ICURCOL 1)) BB (RETURN X)))
(SECTION (IO LISP SYS) SYMBOL)
(FUNCTION ((GETID . LISP) SYMBOL) ((A SYMBOL)))
(FUNCTION ((MAKEID . LISP) SYMBOL) ((A (ARRAY OCTAL))))
(FUNCTION ((MAKID . FSM) SYMBOL) NIL)
(FUNCTION ((MAKIDB . FSM) SYMBOL) NIL)
(FUNCTION ((MGENID . FSM) SYMBOL) NIL)
(FUNCTION ((SYM2OCT . LISP) OCTAL) ((A SYMBOL)))
(FUNCTION (RCNV OCTAL)
 ((A SYMBOL)) (BLOCK ((B REAL A)) (RETURN (R2O. B))))
(FUNCTION (ICNV OCTAL)
 ((A SYMBOL)) (BLOCK ((B INTEGER A)) (RETURN (I2O. B))))
(FUNCTION (RDFUNC SYMBOL) NIL)
(FUNCTION (SYMARY SYMBOL) NIL)
(FUNCTION (BOLARY SYMBOL) NIL)
(FUNCTION (FNCARY SYMBOL) NIL)
(FUNCTION (NUMULT SYMBOL)
 ((TYPE SYMBOL) (CNV (FUNCTIONAL OCTAL SYMBOL)))
 (ERROR (QUOTE (MULTI.DIMENSIONAL ARRAYS ILLEGAL))))
(DECLARE (READL BOOLEAN FLUID NIL) (READA BOOLEAN FLUID NIL))
(FUNCTION (ROWREAD SYMBOL)
 ((TYPE SYMBOL) (CNV (FUNCTIONAL OCTAL SYMBOL)) (L SYMBOL))
 (BLOCK ((S SYMBOL))
  A (SET S (READ))
  (IF READL (BLOCK NIL (SET L (CONS (CNV S) L)) (GO A))
   (NQ S 3) (RETURN (ERROR (QUOTE (ILLEGAL ARRAY SYNTAX)))))
  (BLOCK ((I INTEGER (LENGTH L)))
   (BLOCK ((AR (ARRAY OCTAL) (CREATE I TYPE 0)))
    (FOR I (STEP I -1 LS 1)
     (BLOCK NIL (SET (AR I) (CAR L)) (SET L (CDR L))))
    (RETURN AR)))))
(FUNCTION ((ARREAD . LISP) SYMBOL)
 NIL (BLOCK ((TYPE SYMBOL (READ)))
  (IF (EQ TYPE (QUOTE FUNCTION))
   (RETURN (RDFUNC))
   (EQ TYPE (QUOTE SYMBOL))
   (RETURN (SYMARY))
   (EQ TYPE (QUOTE BOOLEAN))
   (RETURN (BOLARY))
   (EQ TYPE (QUOTE FUNCTIONAL)) (RETURN (FNCARY)))
  (BLOCK ((CNV (FUNCTIONAL OCTAL SYMBOL)))
   (IF (EQ TYPE (QUOTE REAL))
    (SET CNV RCNV)
    (EQ TYPE (QUOTE INTEGER))
    (SET CNV ICNV)
    (EQ TYPE (QUOTE OCTAL))
    (SET CNV SYM2OCT)
    (RETURN (ERROR (CONS TYPE (QUOTE (ILLEGAL ARRAY TYPE))))))
   (BLOCK ((READL BOOLEAN FLUID TRUE)
     (READA BOOLEAN FLUID TRUE) (L SYMBOL))
    (SET L (READ))
    (IF READA (SET READA NIL) (RETURN (NUMULT TYPE CNV)))
    (RETURN (IF READL (ROWREAD TYPE CNV (LIST (CNV L)))
      (EQ L 3)
      (CREATE 0 TYPE 0)
      (ERROR (QUOTE (ILLEGAL ARRAY SYNTAX)))))))))
(FUNCTION ((RDLIST . LISP) SYMBOL)
 NIL (BLOCK ((S SYMBOL (LIST NIL))
   (R SYMBOL) (READL BOOLEAN FLUID TRUE) (P SYMBOL))
  (SET P S)
  A (SET R (READ))
```

```
    (IF READL (BLOCK NIL (SET P (SET (CDR P) (LIST R))) (GO A)))          0075700
    (CASE R (BLOCK NIL (SET (CDR P) (READ))                               0075800
      (SET READL TRUE)                                                    0075900
      (SET R (READ))                                                      0076000
      (IF (OR READL (NQ R 2)) (GO ERR) (RETURN (CDR S))))                 0076100
     (RETURN (CDR S)) (GO ERR))                                           0076200
    ERR (RETURN (ERROR (QUOTE (ILLEGAL LIST STRUCTURE))))))               0076300
  (SECTION FSM SYMBOL)                                                    0076400
  (FUNCTION (TOKEN INTEGER) NIL)                                          0076500
  (DECLARE (FSMSYM SYMBOL OWN)                                            0076600
   (FSMOCT OCTAL OWN) (FSMREL REAL OWN) (SPFLAG BOOLEAN OWN NIL))         0076700
  (SECTION (IO LISP FSM SYS) SYMBOL)                                      0076800
  (FUNCTION ((READ . IO) SYMBOL)                                          0076900
   NIL (BLOCK ((N INTEGER))                                               0077000
    A (CASE (SET N (TOKEN))                                               0077100
     (RETURN (MGENID))                                                    0077200
     (RETURN FSMSYM)                                                      0077300
     (RETURN FSMOCT)                                                      0077400
     (RETURN (C2I. FSMOCT))                                               0077500
     (RETURN FSMREL)                                                      0077600
     (BLOCK ((Y (ARRAY OCTAL) FSMSYM))                                    0077700
      (BLOCK ((X SYMBOL (READ)))                                          0077800
       (IF (NUMBP X)                                                      0077900
        (RETURN (IF (EQ ((GETCHAR . LISP) Y 1) (QUOTE '+))                0078000
          X (MINUS X)))) (SET FSMSYM (SCONCS Y FSMSYM)) (GO B2)))         0078100
     (RETURN (RDLIST))                                                    0078200
     (IF READA (BLOCK NIL (SET READA NIL) (RETURN 5))                     0078300
      (RETURN (ARREAD)))                                                  0078400
     (GO A)                                                               0078500
     (GO A)                                                               0078600
     (RETURN ((MAKEID . LISP) FSMSYM))                                    0078700
     (RETURN (MAKIDB))                                                    0078800
     (LABEL MID (BLOCK NIL (SET SPFLAG NIL)                               0078900
       (RETURN ((MAKID . FSM)))))                                         0079000
     (GO MID)                                                             0079100
     (LABEL GID (RETURN ((GETID . LISP) FSMSYM)))                         0079200
     (IF (OR READA READL)                                                 0079300
      (BLOCK NIL (ERROR (QUOTE (DATA.SEPARATOR READ WITHIN                0079400
         S.EXPRESSION))) (GO A)) (GO GID)) (GO B1))                       0079500
    B1 (IF READL (BLOCK NIL (SET READL NIL)                               0079600
      (RETURN (PLUS N -16))))                                             0079700
    B2 (RETURN (ERROR (CONS FSMSYM (QUOTE (ILLEGAL S.EXPRESSION))))))) 0079800
  (FUNCTION ((READ . LISP) SYMBOL)                                        0079900
   NIL (BLOCK ((READL BOOLEAN FLUID FALSE)                                0080000
     (READA BOOLEAN FLUID FALSE)) (RETURN ((READ . IO))))))               0080100
(PRINTF (SECTION (IO LISP SYS) SYMBOL)                                    0080200
  (FUNCTION ((ENDOUT . LISP) NOVALUE)                                     0080300
   NIL (BLOCK ((INDEX INTEGER 0))                                         0080400
    (SET SUMLINE (PLUS SUMLINE 1))                                        0080500
    (SET STATUS 1)                                                        0080600
    (IF (GR SUMLINE RECORD) (BLOCK NIL (ENDOUTR) (GO E1)))                0080700
    (SET CURCOL LMG)                                                      0080800
    (IF (GR SUMLINE 1) (SET INDEX (TIMES (PLUS -1 SUMLINE) WPL)))         0080900
    (LOCSET LINELOC (BUFLOC INDEX))                                       0081000
    E1 (SET CURLINE (PLUS CURLINE 1))                                     0081100
    P1 (IF (EQ CURLINE BOT) (BOTO))                                       0081200
    (IF (GR CURLINE PAGE) (ENDOUTP))))                                    0081300
  (FUNCTION ((ENDOUTR . LISP) NOVALUE)                                    0081400
   NIL (BLOCK NIL (MOVE)                                                  0081500
    (CLEAR CURFN)                                                         0081600
    (SET SUMLINE 1) (SET CURCOL LMG) (LOCSET LINELOC (BUFLOC 0))))        0081700
  (FUNCTION ((ENDOUTP . LISP) NOVALUE)                                    0081800
   NIL (BLOCK ((X INTEGER (PLUS TOP PAGE (MINUS CURLINE)))                0081900
   (Y INTEGER 0))
  (FOR Y (STEP 1 1 GR X) (ENDOUT)) T1 (SET CURLINE TOP)))
(FUNCTION ((PRINSTRING . LISP) SYMBOL)
 ((SS SYMBOL))
 (BLOCK ((X INTEGER (STRINGL SS))
   (Y INTEGER) (Z INTEGER 1) (SA (ARRAY OCTAL) SS))
  (BLOCK ((LMG INTEGER FLUID LOC Z))
   (IF (NULL (PRMODE . SYS))
    (BLOCK NIL (FOR Y (STEP 1 1 GR X)
      (PRINCH ((GETCHAR . LISP) SA Y))) (GO XT)))
   (PRINCH (OCT2CH 43Q))
   (FOR Y (STEP 1 1 GR X)
    (BLOCK ((Z SYMBOL ((GETCHAR . LISP) SA Y)))
     (IF (OR (EQ Z (QUOTE ''))
       (EQ Z (OCT2CH 43Q)) (EQ Z (OCT2CH 3Q)))
      (PRINCH (QUOTE ''))
      (NOT (NORMSP Z))
      (BLOCK ((CH SYMBOL (NUMSTR (CH2OCT Z))))
       (PRINCH XXCHAR)
       (PRINCH (QUOTE C))
       (BLOCK ((PRMODE BOOLEAN FLUID FALSE)) (PRINSTRING CH))
       (PRINCH XXDLIM) (GO XTFOR)))
     (PRINCH Z)
     (IF (EQ Z (XXCHAR . SYS)) (PRINCH (QUOTE I))) XTFOR))
   T1 (PRINCH (OCT2CH 43Q)) XT (RETURN SS))))
(FUNCTION ((PRINID . LISP) SYMBOL)
 ((TT SYMBOL))
 (BLOCK NIL (IF (NOT (IDP TT))
   (RETURN NIL)
   (AND (GNMODE . SYS) (GENIDP TT))
   (BLOCK NIL (PRINCH (XXCHAR . SYS)) (PRINCH (QUOTE G))))
  (IF (NORMSP TT)
   (BLOCK (((PRMODE . SYS) BOOLEAN FLUID FALSE))
    (PRINSTRING (TOSTRG TT)) (GO XT))
   (PRMODE . SYS) (PRINCH (XXCHAR . SYS)))
  T2 (PRINSTRING (TOSTRG TT)) XT (RETURN TT)))
(ROUTINE (IDNAME SYMBOL)
 ((A SYMBOL))
 (BLOCK NIL X (IF (IDP (SET A (O2S. (BIT 0 18 (CORE (PLUS 1 (S2O. A)
        )))))) (RETURN A) (GO X))))
(FUNCTION ((FVLIST . SYS) SYMBOL) ((S SYMBOL)))
(FUNCTION ((PRINATOM . LISP) SYMBOL)
 (TT)
 (BLOCK ((X SYMBOL))
  (IF (IDP TT)
   (BLOCK NIL (PRINID TT) (GO B))
   (ARRAYP TT)
   (BLOCK NIL (PRINARRAY TT) (GO B))
   (STRINGP TT)
   (SET X TT) (NUMBP TT) (SET X (NUMSTR TT)) (GO C))
  A (PRINSTRING X)
  B (RETURN TT)
  C (IF (FORMALP TT)
   (PRINSTRING (QUOTE (*STRING FORMAL ' )))
   (OR (OWNP TT) (FLUIDP TT))
   (GO D) (BOOLP TT) (BLOCK NIL (SET X (TOSTRG TT)) (GO A)))
  ERR (SET X (SCONCS (TOSTRG (QUOTE LA)) (NUMSTR (S2O. TT))))
  (GO A)
  D (SET X (CONS (IDNAME TT) (O2S. (BIT 24 18 (CORE (S2O. TT))))))
  (PRIN (CONS X (FVLIST TT))) (GO B)))
(FUNCTION O2B2S ((X OCTAL)) (NQ X 0))
(FUNCTION O2F2S ((X OCTAL)) (O2F. X))
(FUNCTION O2R2S ((X OCTAL)) (O2R. X))
```

```
(FUNCTION O2I2S ((X OCTAL)) (O2I. X))
(FUNCTION ARTYPE ((S SYMBOL))
 (CASE (PLUS 1 (BIT 42 3 (CORE (S2O. S))))
  (QUOTE SYMBOL)
  (QUOTE BOOLEAN)
  (QUOTE OCTAL)
  (QUOTE INTEGER)
  (QUOTE REAL)
  (QUOTE FUNCTIONAL) (ERROR (QUOTE (ILLEGAL ARRAY TYPE)))))
(FUNCTION PRINSMAR ((X (ARRAY SYMBOL)) (N INTEGER))
 (BLOCK ((I INTEGER 1))
  (FOR I (STEP I 1 EQ N)
   (BLOCK ((S SYMBOL (X I)))
    (PRINCH (QUOTE ' ))
    (IF (OR (ARRAYP S) (FORMALP S)) (PRINCH (QUOTE '.)))
    (PRINO S))) (RETURN X)))
(FUNCTION (PRINARRAY . LISP)
 ((X (ARRAY OCTAL)))
 (IF (NOT (ARRAYP X))
  X (BLOCK ((N INTEGER (ARSIZE (S2O. X)))
    (TYPE SYMBOL (ARTYPE X)))
   (PRINCH (QUOTE ' ))
   (PRINO TYPE)
   (IF (EQ TYPE (QUOTE SYMBOL))
    (PRINSMAR X N)
    (BLOCK ((I INTEGER 1)
      (CNV (FUNCTIONAL SYMBOL OCTAL)
       (IF (EQ TYPE (QUOTE BOOLEAN))
        O2B2S (EQ TYPE (QUOTE FUNCTIONAL))
        O2F2S (EQ TYPE (QUOTE REAL))
        O2R2S (EQ TYPE (QUOTE INTEGER))
        O2I2S (EQ TYPE (QUOTE OCTAL)) OCT2SYM OCT2SYM)))
     (FOR I (STEP I 1 EQ N)
      (BLOCK NIL (PRINCH (QUOTE ' )) (PRINO (CNV (X I)))))))
   (PRINCH (QUOTE ' )) (RETURN X))))
(FUNCTION ((PRINWORD . LISP) OCTAL)
 ((X OCTAL))
 (BLOCK NIL (IF (EQ 6 WPL)
   (ERROR MSG5)
   (EQ STATUS 4)
   (KEY)
   (GR SUMLINE RECORD) (ENDOUTR) (EQ SUMLINE 0) (SET SUMLINE 1))
  (SET (BUFLOC SUMLINE) X)
  (SET SUMLINE (PLUS 1 SUMLINE)) (RETURN X)))
(FUNCTION ((PRINT . LISP) SYMBOL)
 ((X SYMBOL)) (BLOCK NIL (PRIN X) (ENDOUT) (RETURN X)))
(FUNCTION ((PRINO . LISP) SYMBOL)
 ((X SYMBOL))
 (BLOCK ((J SYMBOL))
  (IF (ATOM X) (GO P4))
  (SET J X)
  (PRINCH (QUOTE '())
   P1 (PRINO (CAR J))
   (IF (NULL (CDR J)) (GO P3))
   (PRINCH (QUOTE ' ))
   (IF (ATOM (CDR J)) (GO P2))
   (SET J (CDR J))
   (GO P1)
   P2 (PRINCH (QUOTE '.))
   (PRINCH (QUOTE ' )) (PRINATOM (CDR J)) P3 (PRINCH (QUOTE ')))
  (RETURN X) P4 (PRINATOM X) XT (RETURN X)))
(FUNCTION ((SYMPRINT . LISP) SYMBOL)
 ((X SYMBOL)) (BLOCK NIL (SYMPRIN X) (ENDOUT) (RETURN X)))
```

```
  (FUNCTION ((SYMPRIN . LISP) SYMBOL)
   ((X SYMBOL))
   (BLOCK (((PRMODE . SYS) BOOLEAN FLUID TRUE)) (RETURN (PRINO X))))
  (FUNCTION ((PRIN . LISP) SYMBOL)
   ((X SYMBOL))
   (BLOCK (((PRMODE . SYS) BOOLEAN FLUID FALSE))
    (RETURN (PRINO X))))
  (FUNCTION ((PRINCH . LISP) SYMBOL)
   ((X SYMBOL))
   (BLOCK ((SW INTEGER 1))
    (IF (EQ STATUS 4) (KEY))
    (IF (EQ SUMLINE 0)
     (BLOCK NIL (SET SUMLINE 1)
      (SET CURCOL LMG) (LOCSET LINELOC (BUFLOC 0))))
    (IF (EQ X (OCT2CH 0Q)) (BLOCK NIL (SET SW 2) (ENDOUT)))
    (IF (EQ CURCOL RMG) (RMGO))
    (IF (GR CURCOL MAXCOL) (ENDOUT))
    (CASE SW (GO AA) (GO BB))
    AA (SETCHAR X LINELOC CURCOL)
    (SET CURCOL (PLUS CURCOL 1)) BB (RETURN X))))
(MOVEO (SECTION (IO LISP SYS) SYMBOL)
  (ROUTINE (MOVEO INTEGER)
   ((S INTEGER) (B OCTAL LOC))
   (BLOCK NIL (CODE (ACR A.) (STF (ENTRY MLOC)))
    (SET (CORENTRY MSIZE) S)
    (SET (CORENTRY MNAME) NAME)
    (SET (CORENTRY MINOUT) (CORENTRY OUT))
    (SET (CORENTRY MSECTR) SECTOR)
    (CODE (LDA (ENTRY MCALL) (RA R)) (BUC (ENTRY DSPCHR)))
    (RETURN (BIT 0 6 (CORENTRY MSTAT)))))
  (FUNCTION (OUTTY NOVALUE)
   NIL (BLOCK NIL (SETCHAR (OCT2CH 3) LINELOC CURCOL)
    (T8X12) (MOVEO 19 FIXLOC) XT))
  (FUNCTION (OUTAPE NOVALUE)
   NIL (BLOCK ((Y INTEGER 0) (J INTEGER 2) (I INTEGER 1))
    (SETBUF FIXLOC 606060606060606Q1)
    (SETCHAR (OCT2CH 36Q) LINELOC CURCOL)
    T0 (LOCSET LINELOC (BUFLOC Y))
    T1 (IF (T8X6 77Q LINELOC FIXLOC J I) (GO T2))
    (SET J (PLUS J 1))
    (SET I (PLUS I 1))
    (IF (LQ I MAXCOL) (GO T1))
    (SET I 0)
    (T8X6 77Q LINELOC FIXLOC J I)
    (SET J (PLUS J 2))
    (SET I 1)
    (SET Y (PLUS Y WPL))
    (GO T0)
    T2 (IF (LQ (MOVEO (TIMES RECORD (PLUS 1 (IQUOTIENT (PLUS MAXCOL
         -1) 8))) FIXLOC) 3) (GO XT)) (SET STATUS 5) XT))
  (FUNCTION (OUTDISC NOVALUE)
   NIL (BLOCK ((Y INTEGER 0) (J INTEGER 1) (I INTEGER 1))
    (IF (LQ SECTOR MAXSEC) (GO T1))
    (IF (LQ (MODIFY) 3) (GO T0))
    (SET STATUS 4)
    (GO XT)
    T0 (SET MAXSEC (PLUS MAXSEC 8))
    T1 (SETBUF FIXLOC 606060606060606Q1)
    (SETCHAR (OCT2CH 36Q) LINELOC 80)
    T2 (LOCSET LINELOC (BUFLOC Y))
    T3 (IF (T8X6 76Q LINELOC FIXLOC J I) (GO T4))
    (SET J (PLUS J 1))
    (SET I (PLUS I 1))
  (IF (LS I 80) (GO T3))
  (IF (T8X6 76Q LINELOC FIXLOC J I) (GO T4))
  (SET I 0)
  (T8X6 76Q LINELOC FIXLOC J I)
  (SET J (PLUS J 1))
  (SET I 1)
  (SET Y (PLUS Y WPL))
  (GO T2)
  T4 (SEQNC)
  (IF (LQ (MOVEO 512 FIXLOC) 3) (GO T5))
  (SET STATUS 5) (GO XT) T5 (SET SECTOR (PLUS SECTOR 1)) XT))
(FUNCTION (OUTPAS NOVALUE)
 NIL (BLOCK ((X INTEGER))
  (LOCSET LINELOC (BUFLOC 0))
  (SET X (MOVEO (TIMES RECORD WPL) LINELOC))
  (IF (LQ X 3) (GO XT)) (SET STATUS 5) XT))
(FUNCTION (OUTDCAS NOVALUE)
 NIL (BLOCK ((X INTEGER)
   (Y INTEGER) (Z INTEGER (TIMES RECORD WPL)))
  (SET Y (IQUOTIENT (PLUS Z -1) 512))
  (IF (LQ (PLUS SECTOR Y) MAXSEC) (GO T1))
  (SET X (MODIFY))
  (IF (LQ X 3) (GO T0))
  (SET STATUS 4)
  (GO XT)
  T0 (SET MAXSEC (PLUS MAXSEC 8))
  T1 (LOCSET LINELOC (BUFLOC 0))
  (SET X (MOVEO Z LINELOC))
  (IF (LQ X 3) (GO T2))
  (SET STATUS 5) (GO XT) T2 (SET SECTOR (PLUS SECTOR Y 1)) XT))
(FUNCTION (T8X6 BOOLEAN)
 ((EOR OCTAL)
  (SOURCE OCTAL LOC) (SINK OCTAL LOC) (J INTEGER) (I INTEGER))
 (BLOCK ((X OCTAL))
  (IF (EQ I 0) (BLOCK NIL (SET X 32Q14) (GO AB)))
  (SET X (CH2OCT ((GETCHAR . IO) SOURCE I)))
  (CODE (BXE (LABEL TEOR) A 36Q) (BXE (LABEL TEOF) A 34Q))
  (SET X (CNVRTB (PLUS X 1)))
  AB (CODE (LDL X L7.0)
   T1 (BLC (LABEL T2) 0 3)
   (STZ A.)
   (RETURN)
   T2 (LDA J)
   (STZ B.)
   (SOR A.)
   (LDM A.)
   (SFC 3 R)
   (ADD SINK RA)
   (AOR A. (RA S))
   (LDX A. 0 2)
   (STZ A.)
   (SFC -3 R)
   (XEC (LABEL T3) A)
   (BUC 0 3)
   T3 (STL 0 (S7.0 2))
   (STL 0 (S7.1 2))
   (STL 0 (S7.2 2))
   (STL 0 (S7.3 2))
   (STL 0 (S7.4 2))
   (STL 0 (S7.5 2))
   (STL 0 (S7.6 2))
   (STL 0 (S7.7 2))
   TEOF (LDL 77Q (L567.7 R))
   (LDA 80 (L567.7 R))
   (BSX ((LABEL T2) 1) 3 ((LABEL TEOR) 2))
   TEOR (LDL EOR RA)
   (BUC (LABEL T2) 0 3) (LDA 1 (RA R)) (RETURN))))
 (ROUTINE (T8X12 NOVALUE)
  NIL (BLOCK ((C INTEGER)
   (Y INTEGER 0) (I INTEGER 1) (J INTEGER 1))
  ST (LOCSET LINELOC (BUFLOC Y))
  ST1 (IF (EQ 3 (SET C (PLUS (MINUS CHO)
     (S20. ((GETCHAR . IO) LINELOC I)))))) (GO TEOM))
  (SET C (BIT 24 12 (CNVRTB (PLUS C 1))))
  (CODE (LDL C) (LDA J) (BUC (LABEL T2) 0 3))
  (SET I (PLUS I 1))
  (SET J (PLUS J 1))
  (IF (LQ I MAXCOL) (GO ST1))
  (SET Y (PLUS Y WPL))
  (SET I 1)
  (GO ST)
  T2 (CODE (STZ B.)
   (SOR A.)
   (LDM A.)
   (SFC 2 R)
   (ADD (ENTRY FIXBUF) (RA R))
   (AOR A. (RA S))
   (LDX A. 0 2)
   (STZ A.)
   (SFC -2 R)
   (XEC (LABEL T3) A)
   (BUC 0 3)
   T3 (STL 0 (S67.1 2))
   (STL 0 (S67.3 2)) (STL 0 (S67.5 2)) (STL 0 (S67.7 2)))
  TEOM (CODE (LDL 15Q (RA R))
   (LDA J)
   (BUC (LABEL T2) 0 3)
   (LDL 3 (RA R)) (LDA J) (AOR A. (RA S)) (BUC (LABEL T2)    3)))))
(MOVEI (ROUTINE (MOVEI INTEGER)
  ((S INTEGER) (B OCTAL LOC))
  (BLOCK NIL (CODE (ACR A.) (STF (ENTRY MLOC)))
   (SET (CORENTRY MSIZE) S)
   (SET (CORENTRY MNAME) INAME)
   (SET (CORENTRY MINCUT) (CORENTRY IN))
   (SET (CORENTRY MSECTR) ISECTOR)
   (CODE (LDA (ENTRY MCALL) (RA R)) (BUC (ENTRY DSPCHR)))
   (SET ISIZE (CORENTRY MWDSIN))
   (RETURN (BIT 0 6 (CORENTRY MSTAT)))))
 (FUNCTION (INTTY NOVALUE)
  NIL (BLOCK NIL (IF (NQ DDSW 0) (ICINI))
   (BLOCK ((X INTEGER (TIMES IRECORD (PLUS 1 (IQUOTIENT (PLUS
         IMAXCOL -1) 4)))))
    (CODE (LDA (ENTRY BELL) (R L567.7))
     (BUC (ENTRY DSPCHR)) (STX (DDSW . IO) 0 8))
    (MOVEI X FIXLOC) (SET DDSW 0) (SET TTYMAX (T12X8)))))
 (FUNCTION (INTAPE NOVALUE)
  NIL (BLOCK ((X INTEGER (TIMES IRECORD (PLUS 1 (IQUOTIENT (PLUS
        IMAXCOL -1) 8)))))
   (IF (LQ (SET X (PLUS -1 (MOVEI X FIXLOC))) 2)
    (T6X8 IMAXCOL TRANTP) (SET ISTATUS X))))
 (FUNCTION (INDISC NOVALUE)
  NIL (BLOCK ((X INTEGER))
   T0 (IF (LQ ISECTOR IMAXSEC) (GO T1))
   (SET ISTATUS 4)
   (GO XT)
   T1 (SET X (MOVEI 512 FIXLOC))
```

```
    (SET ISECTOR (PLUS ISECTOR 1))
    (IF (LQ X 3) (GO T2))
    (SET ISTATUS 5)
    (GO XT) T2 (IF (T75) (GO TO)) (T6X8 80 TRANDC) XT))
 (FUNCTION (INTPAS NOVALUE)
  NIL (BLOCK ((X INTEGER))
    (LOCSET ILINELOC (IBUFLOC 0))
    (SET X (PLUS -1 (MOVEI (TIMES IRECORD IWPL) ILINELOC)))
    (IF (LQ X 2) (GO XT)) (SET ISTATUS X) XT))
 (FUNCTION (INDCAS NOVALUE)
  NIL (BLOCK ((X INTEGER)
     (Y INTEGER) (Z INTEGER (TIMES IRECORD IWPL)))
    (SET Y (PLUS 1 (IQUOTIENT (PLUS Z -1) 512)))
    (IF (LQ (PLUS ISECTOR Y) IMAXSEC) (GO TO))
    (SET ISTATUS 4)
    (GO XT)
    TO (LOCSET ILINELOC (IBUFLOC 0))
    (SET X (MOVEI Z ILINELOC))
    (IF (LQ X 3) (GO T1))
    (SET ISTATUS 5) (GO XT) T1 (SET ISECTOR (PLUS ISECTOR Y 1)) XT))
 (ROUTINE (T12X8 INTEGER)
  NIL (BLOCK ((Y INTEGER 1)
     (B6 INTEGER)
     (I INTEGER 48)
     (OCT OCTAL)
     (L INTEGER)
     (CH OCTAL 0)
     (S INTEGER 1)
     (SC (ARRAY OCTAL) (C2S. (ENTRY FIXBUF))) (J INTEGER))
    ST (SET L Y)
    (SET B6 48)
    (SET OCT 0)
    (SET J 1)
    ST1 (IF (LS (SET I (PLUS I -12)) 0)
     (BLOCK NIL (SET I 36) (SET S (PLUS S 1))))
    (SET CH (WORDAND 177Q (BIT I 12 (SC S))))
    (IF (EQ CH 3Q)
     (IF (EQ Y 1) (GO B) (GO B0))
     (LS (SET B6 (PLUS B6 -8)) 0)
     (BLOCK NIL (SET (IBUFLOC L) OCT)
      (SET L (PLUS L 1)) (SET B6 40) (SET OCT 0)))
    (BITSET B6 8 OCT (BIT 0 12 (CNVRTB (PLUS CH 1))))
    TO (SET J (PLUS J 1))
    (IF (LQ J IMAXCOL) (GO ST1))
    (SET Y (PLUS Y IWPL))
    (SET (IBUFLOC L) OCT)
    (GO ST)
    B0 (SET J 3Q15) B (SET (IBUFLOC L) OCT) (RETURN (PLUS J 1))))
 (FUNCTION (T6X8 NOVALUE)
  ((COL INTEGER) (WHAT (FUNCTIONAL INTEGER OCTAL INTEGER)))
  (BLOCK ((Y INTEGER 1)
     (B6 INTEGER)
     (I INTEGER 48)
     (OCT OCTAL)
     (L INTEGER)
     (CH OCTAL 0)
     (S INTEGER 1)
     (SC (ARRAY OCTAL) (C2S. (ENTRY FIXBUF))) (J INTEGER))
    ST (SET L Y)
    (SET B6 48)
    (SET OCT 0)
    (SET J 1)
    ST1 (IF (LS (SET I (PLUS I -6)) 0)
     (BLOCK NIL (SET I 42) (SET S (PLUS S 1))))
    (CASE (WHAT (SET CH (BIT I 6 (SC S))) J) (GO TO) (GO B) NIL)
    CNV (IF (LS (SET B6 (PLUS B6 -8)) 0)
     (BLOCK NIL (SET (IBUFLOC L) OCT)
      (SET L (PLUS L 1)) (SET B6 40) (SET OCT 0)))
    (BITSET B6 8 OCT (BIT 12 12 (CNVRTB (PLUS CH 1))))
    TO (SET J (PLUS J 1))
    (IF (LQ J CCL) (GO ST1))
    (SET Y (PLUS Y IWPL))
    (SET (IBUFLOC L) OCT)
    (GO ST)
   B (SET (IBUFLOC L) OCT) (SET (IBUFLOC (PLUS Y IWPL)) 74Q13)))
  (ROUTINE (T75 BOOLEAN)
   NIL (EQ 75Q (BIT 0 6 (CORE (PLUS 10 (ENTRY FIXBUF)))))))
(ALX (SECTION (LISP SYS) SYMBOL)
  (FUNCTION DEVTYPE (FILE)
   (IF (SET FILE (GET FILE (FILES. . IO)))
    (CDR (GET (QUOTE UNIT) FILE)) NIL))
  (SECTION (IO LISP SYS) SYMBOL)
  (ROUTINE ((CLEAR . LISP) NOVALUE)
   ((FN SYMBOL))
   (BLOCK ((DL SYMBOL (GET FN FILES.)) (BB (ARRAY OCTAL)))
    (IF (NULL DL) (GO XT))
    (SET BB (CDR (GET (QUOTE BUF) (CDR DL))))
    TO (SETBUF (BB 0) 0) XT))
  (FUNCTION (SEQNO NOVALUE)
   NIL (BLOCK ((C INTEGER COUNT)
     (L INTEGER 10) (N INTEGER) (S INTEGER (PLUS -1 SUMLINE)))
    (FOR N (STEP 1 1 GR S)
     (BLOCK ((Z INTEGER (CVRTNM C)))
      (IF (LS C 1000)
       (SET Z (BIT 30 18 Z))
       (LS C 10000) (SET Z (BIT 24 24 Z)) (SET Z (BIT 18 30 Z)))
      (SET (BIT 6 42 (BUFIX L)) Z)
      (SET L (PLUS L 10)) (SET C (PLUS C 100))))
    (SET (BIT 24 24 (BUFIX 511)) COUNT)
    (SET (BIT 0 24 (BUFIX 511)) (PLUS C -100))
    (SET (BIT 24 24 (BUFIX 512)) S)
    (SET (BIT 0 24 (BUFIX 512)) RECORD) (SET COUNT C)))
  (ROUTINE ((GET . LISP) SYMBOL)
   ((AA SYMBOL) (X SYMBOL))
   (BLOCK ((Y SYMBOL))
    G1 (IF (NULL X) (RETURN NIL))
    (SET Y (CAR X))
    (IF (ATOM Y) (GO G2) (EQN (CAR Y) AA) (RETURN Y))
    G2 (SET X (CDR X)) (GO G1)))
  (FUNCTION (CVRTNM INTEGER)
   ((FN SYMBOL))
   (BLOCK ((ST (ARRAY OCTAL) (TOSTRG FN)))
    (RETURN (CVRTN1 (ST 0) 1))))
  (FUNCTION (CHSUPL SYMBOL)
   NIL (BLOCK ((X SYMBOL))
    A (IF (NQ (S20. (SET X (READCH))) (CHO . SYS))
     (RETURN X)
     (RETURN (OCT2CH (CASE ISTATUS 37Q 36Q 34Q 31Q 0Q))))))
  (FUNCTION (TTYSPL SYMBOL)
   NIL (BLOCK ((X SYMBOL))
    A (IF (NQ (S20. (SET X (READCH))) (CHO . SYS))
     (RETURN X)
     (GQ ICURCOL TTYMAX)
     (BLOCK NIL (ENDIN) (GO A))
     (RETURN (OCT2CH (CASE ISTATUS 37Q 36Q 34Q 31Q 0Q))))))
  (ROUTINE ((GETCHAR . LISP) SYMBOL)
 ((A (ARRAY OCTAL)) (B INTEGER)) ((GETCHAR . IO) (A 0) B))
(ROUTINE ((GETCHAR . IO) SYMBOL)
 ((SS OCTAL LOC) (CC INTEGER))
 (BLOCK NIL (CODE (SCR A.)
   (LDM A.)
   (MUL 1 (R L7))
   (DVD 6 (R L7))
   (ADD SS RA)
   (AOR A. (RA S))
   (LDA 0 A)
   (TST B. 7004Q3)
   (BSX (LABEL G1) 1 -8)
   (BSX (LABEL G1) 1 -16)
   (BSX (LABEL G1) 1 -24)
   (BSX (LABEL G1) 1 16)
   (BSX (LABEL G1) 1 8)
   (BSX (LABEL G1) 1 0)
   G1 (CYA 0 (R 1))
   (ANA (NUMBER 377Q)) (ADD (CH0 . SYS) RA) (RETURN))))
(ROUTINE ((SETCHAR . IO) SYMBOL)
 ((CH SYMBOL) (SS OCTAL LOC) (CC INTEGER))
 (BLOCK NIL (CODE (SCR A.)
   (LDM A.)
   (MUL 1 (R L7))
   (DVD 6 (R L7))
   (ADD SS RA)
   (AOR A. (RA S))
   (LDX A. 0 2)
   (TST B. 7004Q3)
   (BSX (LABEL S1) 1 8)
   (BSX (LABEL S1) 1 16)
   (BSX (LABEL S1) 1 24)
   (BSX (LABEL S1) 1 -16)
   (BSX (LABEL S1) 1 -8)
   (BSX (LABEL S1) 1 0)
   S1 (LDB 377Q (RA R))
   (LDA CH RA)
   (SUB (CH0 . SYS) (RA S))
   (LDM A.)
   (CYA 0 (R 1))
   (CYB 0 (R 1)) (CON 0 (624Q4 2)) (LDA CH RA) (RETURN))))
(ROUTINE (CVRTNI INTEGER)
 ((S OCTAL LOC) (I INTEGER))
 (BLOCK NIL (CODE (STZ PUSHA.)
   TO (ARGS)
   (LDA S)
   (STF PUSHP.)
   (LDA I RA)
   (CALL (GETCHAR . IO))
   (SUB (CH0 . SYS) (RA S))
   (ADD 1 (R L567.7))
   (ADD (CNVRTB . IO) L567.7)
   (LDA 0 (L7.0 A))
   (ADD TOP.)
   (CYA -6 R)
   (STF TOP.)
   (AOR I RA) (BXL (LABEL TO) A 8) (LDA TOP.) (RETURN))))
(ROUTINE ((SETBUF . IO) NOVALUE)
 ((SINK OCTAL LOC) (K OCTAL))
 (BLOCK NIL (CODE (LDI 4 (R 7Q6))
   (LDX SINK 0 2)
   (LDX SINK (I L) 4)
   (BAX (D. 2) 4 -1) (STF 1 (2 D)) (BPX (D. -1) 4 1))))
```

```
(ROUTINE (INCMOV NOVALUE)
 ((INC INTEGER))
 (SET (BIT 0 6 (CORENTRY MCALL))
  (PLUS INC (BIT 0 6 (CORENTRY MCALL)))))
(FUNCTION (LOGTTY NOVALUE)
 ((C INTEGER) (M SYMBOL))
 (BLOCK ((OUTF SYMBOL (OUTPUT (QUOTE OTTY))))
  (INCMOV 1)
  (SET (BIT 0 6 (CORENTRY MPOST)) C)
  (PRINT M) (INCMOV -1) (OUTPUT OUTF)))
(FUNCTION TOSTRG (A)
 (IF (STRINGP A)
  A (AND (IDP A) (NOT (GENIDP A)))
  (IF (CHARP A)
   (BLOCK NIL (SET (BIT 18 6 (CORE (S2O. SHORWD))) 1)
    (SET (CORE (PLUS 1 (S2O. SHORWD))) (SHIFT (CH2OCT A) 4.))
    (RETURN SHORWD))
   (BLOCK ((X OCTAL (BIT 18 6 (CORE (PLUS 1 (S2O. A))))))
     (Y OCTAL (CORE (PLUS -1 (S2O. A)))))
    (IF (EQ 0 X) (RETURN (O2S. (BIT 0 18 Y))))
    (SET (CORE (PLUS 1 (S2O. SHORWD))) Y)
    (SET (BIT 18 6 (CORE (S2O. SHORWD))) X) (RETURN SHORWD)))
  (NUMBP A) (NUMSTR A) ((TOSTRG . LISP) A)))
(FUNCTION (TRANDC INTEGER)
 ((CH OCTAL) (J INTEGER))
 (IF (NQ J 80)
  3 (OR (EQ CH 32Q) (LS CH 76Q))
  1 (EQ CH 76Q)
  2 (BLOCK NIL (SET ISECTOR (PLUS IMAXSEC 1)) (RETURN 2))))
(FUNCTION (TRANTF INTEGER)
 ((CH OCTAL) (J INTEGER)) (IF (EQ CH 32Q) 1 (EQ CH 77Q) 2 3)))
RETTYP (SECTION (IO LISP SYS) SYMBOL)
(FUNCTION ((FITATOM . LISP) SYMBOL)
 ((T SYMBOL))
 (BLOCK ((L INTEGER))
  (IF (OR (ARRAYP T) (FLUIDP T) (OWNP T) (FORMALP T)) (GO EO))
  (SET L (STRINGL (TOSTRG T)))
  (IF (LQ (PLUS CURCOL L 2) (IF (GR RMG MAXCOL) MAXCOL RMG))
   (GO RET)) EO (ENDOUT) RET (RETURN (PRINATOM T))))
(FUNCTION (B1 SYMBOL)
 ((S SYMBOL))
 (BLOCK NIL (IF (GR CURCOL LMG) (ENDOUT))
  (PRINCH (QUOTE '())
   (FITATOM (CAR S))
   (PRINCH (QUOTE ' ))
   (BLOCK ((Z INTEGER (DIFFERENCE CURCOL LMG))
     (LM INTEGER CURCOL) (E SYMBOL))
    (BLOCK ((LMG INTEGER FLUID LOC LM))
     (FOR E (IN (CDR S))
      (BLOCK NIL (IF (IDP E)
        (BLOCK NIL (ENDOUT)
         (SET CURCOL (PLUS LMG (MINUS Z)))
         (FITATOM E) (PRINCH (QUOTE ' )))
        (BLOCK NIL (IF (GR CURCOL LMG) (ENDOUT) (SET CURCOL LMG))
         (F1 E)))))))) (PRINCH (QUOTE '))) (RETURN S)))
(DECLARE (L1 SYMBOL FLUID (QUOTE (IF FOR AND OR FUNCTION ROUTINE
   SECTION MACRO)))
 (L2 SYMBOL FLUID (QUOTE (BLOCK DECLARE CODE)))
 (INDENT INTEGER OWN 40))
(FUNCTION (F1 SYMBOL)
 ((S SYMBOL))
 (BLOCK ((Z INTEGER (IF (GR LMG INDENT) 7 (PLUS LMG 1))))
  (BLOCK ((LMG INTEGER FLUID LOC Z))
     (IF (ATOM S)
      (RETURN (FITATOM S))
      (ATOM (CDR S))
      (GO T1)
      (MEMBER (CAR S) L2)
      (RETURN (B1 S))
      (AND (MEMBER (CAR S) L1) (GR CURCOL LMG))
      (ENDOUT) (EQ (CAR S) (QUOTE LAP)) (GO T2))
     T1 (BLOCK ((J SYMBOL S)
       (L INTEGER (IF (AND (ATOM (CAR S))
          (NOT (OR (ARRAYP (CAR S))
            (OWNP (CAR S)) (FLUIDP (CAR S)) (FORMALP (CAR S)))))
         (STRINGL (TOSTRG (CAR S))) 8)))
      (IF (GR (PLUS CURCOL L 2) (IF (GR RMG MAXCOL) MAXCOL RMG))
       (ENDOUT))
      (PRINCH (QUOTE '())
       T11 (F1 (CAR J))
       (IF (NULL (CDR J)) (GO T13))
       (PRINCH (QUOTE ' ))
       (IF (ATOM (CDR J)) (GO T12))
       (SET J (CDR J))
       (GO T11)
       T12 (PRINCH (QUOTE '.))
       (PRINCH (QUOTE ' )) (FITATOM (CDR J)) T13 (PRINCH (QUOTE ')))
      (RETURN S))
     T2 (IF (GR CURCOL LMG) (ENDOUT))
     (PRINCH (QUOTE '())
      (FITATOM (CAR S))
      (PRINCH (QUOTE ' ))
      (BLOCK ((LM INTEGER CURCOL))
       (BLOCK ((LMG INTEGER FLUID LOC LM))
        (B1 (CADR S))
        (ENDOUT) (F1 (CADDR S)) (ENDOUT) (F1 (CADDDR S))))
      (PRINCH (QUOTE ')))) (RETURN S))))
  (FUNCTION ((PRETTYP . LISP) SYMBOL)
   ((S SYMBOL))
   (BLOCK ((ZZ INTEGER 1))
    (BLOCK ((LMG INTEGER FLUID LOC ZZ))
     (IF (ATOM S) (FITATOM S) (F1 S)) (ENDOUT) (RETURN S)))))
****END OF FILE DETECTED
```

```
(FSMDEC (SECTION IO SYMBOL)
  (DECLARE (XXFUNC (FUNCTIONAL SYMBOL) FLUID LOC)
   (XXSAVE SYMBOL FLUID LOC))
  (SECTION SYS SYMBOL)
  (DECLARE (CHO OCTAL OWN) (XXCHAR SYMBOL OWN))
  (SECTION (FSM SYS) SYMBOL)
  (DECLARE (FSCHAR SYMBOL OWN)
   (FSMSYM SYMBOL OWN)
   (FSMOCT OCTAL OWN)
   (FSMREL REAL OWN)
   (FSNUL BOOLEAN OWN FALSE)
   (CHARRAY (ARRAY OCTAL)
    OWN (QUOTE (*OCTAL 5065064000007Q 5065071000007Q 5065071000007Q
     3065073312004Q 5065071000007Q 5065071000007Q 5065071000007Q
     5065071000007Q 5065071000007Q 5065071000007Q 5065071000007Q
     5065071000007Q 5065071000007Q 3065073312004Q 5065071000007Q
     5065071000007Q 5065071000007Q 5065071000007Q 5065071000007Q
     5065071000007Q 5065071000007Q 5065071000007Q 5065071000007Q
     5065071000007Q 5065071000007Q 5065071020004Q 5065071000007Q
     5065071000007Q 5065071020004Q 5065071000007Q 3065073312004Q
     3065073312004Q 206507210001Q1 5065071000007Q 5065071000007Q
     5065072400006Q 5065072117004Q 5065071000005Q 5065071000007Q
     5065072217004Q 5065072107004Q 5065072122004Q 5065072100011Q
     5075072100012Q 5065072117004Q 5075072100012Q 5054132100113Q
     5065072100011Q 5012112100102Q 5012112100102Q 5012112100102Q
     5012112100102Q 5012112100102Q 5012112100102Q 5012112100102Q
     5012112100102Q 5023122100103Q 5023122100103Q 5065072100011Q
     3065073117004Q 5065072100011Q 5065072100011Q 5065072100011Q
     5065071000007Q 5065071000007Q 5105072100101Q 5105072100101Q
     4105072100101Q 5105072100101Q 5035052100101Q 5105072100101Q
     5105072100101Q 5105072100101Q 1105072100101Q 5105072100101Q
     5105072100101Q 5105072100101Q 5105072100101Q 5105072100101Q
     5105072100101Q 5105072100101Q 5041042100101Q 5105072100101Q
     5105072100101Q 5105072100101Q 5105072100101Q 5105072100101Q
     5105072100101Q 5105072100101Q 5105072100101Q 5105072100101Q
     5065072110004Q 5065072100011Q 5065072123004Q 5065072100011Q
     5065072100011Q 5065071000007Q 5065071000007Q 5065071000007Q
     5065071000007Q 5065071000007Q 5065071000007Q 5065071000007Q
     5065071000007Q 5065071000007Q 5065071000007Q 5065071000007Q
     5065071000007Q 5065071000007Q 5065071000007Q 5065071000007Q
     5065071000007Q 5065071000007Q 5065071000007Q 5065071000007Q
     5065071000007Q 5065071000007Q 5065071000007Q 5065071000007Q
     5065071000007Q 5065071000007Q 5065071000007Q 5065071000007Q
     5065071000007Q 5065071000007Q 5065071000007Q 5065071000007Q
     5065071000007Q))) (CARCDR BOOLEAN OWN))
  MACRO1 (((STRSP (LAMBDA (A) (QUOTE 2)))
    (UNSOCT (LAMBDA (A) (QUOTE 3)))
    (UNSINT (LAMBDA (A) (QUOTE 4)))
    (UNSREL (LAMBDA (A) (QUOTE 5)))
    (STRNM (LAMBDA (A) (QUOTE 11)))
    (LITRL (LAMBDA (A) (QUOTE 12)))
    (DLITRL (LAMBDA (A) (QUOTE 13)))
    (GENSP (LAMBDA (A) (QUOTE 1)))
    (OPRTR (LAMBDA (A) (QUOTE 14)))
    (UMARK (LAMBDA (A) (QUOTE 15)))
    (SIGN (LAMBDA (A) (QUOTE 6)))
    (DOT (LAMBDA (A) (QUOTE 17)))
    (LPAR (LAMBDA (A) (QUOTE 7)))
    (RPAR (LAMBDA (A) (QUOTE 18)))
    (LBRAC (LAMBDA (A) (QUOTE 8)))
    (RBRAC (LAMBDA (A) (QUOTE 19)))
    (DATSEP (LAMBDA (A) (QUOTE 16)))
    (UNREC (LAMBDA (A) (QUOTE 20)))
    (REMARK (LAMBDA (A) (QUOTE 9)))
    (SPACER (LAMBDA (A) (QUOTE 10)))
    (F1 (LAMBDA (A) (LIST (QUOTE BIT) 0 6 (CADR A))))
    (F2 (LAMBDA (A) (LIST (QUOTE BIT) 6 3 (CADR A))))
    (F3 (LAMBDA (A) (LIST (QUOTE BIT) 9 6 (CADR A))))
    (F4 (LAMBDA (A) (LIST (QUOTE BIT) 15 3 (CADR A))))
    (F5 (LAMBDA (A) (LIST (QUOTE BIT) 18 3 (CADR A))))
    (F6 (LAMBDA (A) (LIST (QUOTE BIT) 21 3 (CADR A))))
    (F7 (LAMBDA (A) (LIST (QUOTE BIT) 24 3 (CADR A))))
    (F8 (LAMBDA (A) (LIST (QUOTE BIT) 27 3 (CADR A))))
    (F9 (LAMBDA (A) (LIST (QUOTE BIT) 30 6 (CADR A))))
    (F10 (LAMBDA (A) (LIST (QUOTE BIT) 36 3 (CADR A))))))
  (FUNCTION (MAKEST SYMBOL) NIL))
(CHREAD (FUNCTION (CHREAD NOVALUE)
   NIL (BLOCK NIL A1 (SET FSCHAR ((XXFUNC . IO)))
    A14 (IF (EQ FSCHAR ((OCT2CH . LISP) 0Q))
     (LABEL A15 (IF FSNUL (RETURN NIL) (GO A1)))
     (BLOCK NIL (IF (NQ FSCHAR (XXCHAR . SYS)) (RETURN NIL))
      A2 (CHREAD)
      (CASE (F10 (CHARRAY (PLUS 1 (S20. FSCHAR)
          (MINUS (CHO . SYS)))))
       (GO E1) (GO E2) (GO B3) (GO E4) (GO B1))
      E1 (SET FSCHAR (XXCHAR . SYS))
      (RETURN NIL)
      E2 (CHREAD)
      (CASE (F5 (CHARRAY (PLUS 1 (S2C. FSCHAR)
          (MINUS (CHO . SYS))))) (GO B1) (GO E2) (GO B3) (GO E2))
      B1 (SET FSCHAR ((OCT2CH . LISP) 25Q))
      (RETURN NIL)
      B3 (SET FSCHAR ((OCT2CH . LISP) 0Q))
      (GO A15)
      E4 (BLOCK ((O OCTAL 0Q)
        (E INTEGER 0) (I INTEGER 0) (S INTEGER 1))
       C0 (CHREAD)
       (CASE (F6 (CHARRAY (PLUS 1 (S20. FSCHAR)
           (MINUS (CHO . SYS)))))
        (GO C1) (GO C2) (GO C3) (GO C4) (GO C5) (GO C0) (GO CE))
       C1 (CASE S (GO C1S1)
        (GO C1S2) (GO C1S3) (GO C1S4) (GO C1S3))
       C1S1 (SET S 2)
       C1S2 (SET O (WORDOR (SHIFT O 3)
         (DIFFERENCE (S2C. FSCHAR) (S2C. (QUOTE '0)))))
       C1S4 (SET I (PLUS (TIMES 10 I)
         (DIFFERENCE (S2C. FSCHAR) (S2C. (QUOTE '0)))))
       (GO C0)
       C1S3 (SET E (PLUS (TIMES 10 E)
         (DIFFERENCE (S2C. FSCHAR) (S2C. (QUOTE '0)))))
       (GO C0)
       C2 (CASE S (GO C2S1)
        (GO C2S1) (GO C1S3) (GO C1S4) (GO C1S3))
       C2S1 (SET S 4)
       (GO C1S4)
       C3 (CASE S (GO CE) (GO C3S2) (GO C3S3) (GO C3S2))
       CE (SET FSCHAR ((OCT2CH . LISP) 25Q))
       (RETURN NIL)
       C3S2 (SET O I)
       C3S2A (IF (EQ E 0) (GO CR))
       (SET E (PLUS E -1))
       (SET O (TIMES 10 O))
       (GO C3S2A)
       C3S3 (SET O (SHIFT O (TIMES 3 E)))
       CR (IF (AND (LQ 0 O) (LQ O 177Q))
        (SET FSCHAR ((OCT2CH . LISP) O))
        (SET FSCHAR ((OCT2CH . LISP) 25Q)))
       (GO A14)
       C4 (CASE S (GO CE) (GO C4S2) (GO CE))
       C4S2 (SET S 3)
       (GO C0)
       C5 (CASE S (GO CE) (GO C5S2) (GO CE) (GO C5S2) (GO CE))
       C5S2 (SET S 5) (GO C0)))))))
(STSP (FUNCTION (STSP BOOLEAN)
   NIL (BLOCK ((I INTEGER))
    S0 (CHREAD)
    (SET I (CHARRAY (PLUS 1 (CHEAT SYMBOL INTEGER FSCHAR)
       (MINUS (CHEAT OCTAL INTEGER (CHO . SYS))))))
    (CASE (PLUS 1 (F4 I))
     (GO S1) (GO S4) (GO S3) (GO S0) (GO S5))
    S1 (MAKEST)
    (SET (XXSAVE . IO) FSCHAR)
    (SET FSCHAR NIL)
    (SET FSMSYM (MAKEST))
    (RETURN FALSE)
    S3 (SET FSNUL TRUE)
    (CHREAD)
    (SET FSNUL FALSE)
    S4 (MAKEST)
    (GO S0) S5 (SET FSCHAR NIL) (SET FSMSYM (MAKEST)) (RETURN TRUE))))
(LTRL (FUNCTION (LTRL NOVALUE)
   NIL (BLOCK ((P INTEGER 1))
    F1 (MAKEST)
    (CASE P (GO P1) (GO P2) (GO P3) (GO P4) (GO P5))
    P1 (IF (EQ FSCHAR (QUOTE 'C)) (SET P 2) (SET P 5))
    (GO P5)
    P2 (IF (OR (EQ FSCHAR (QUOTE 'A)) (EQ FSCHAR (QUOTE 'D)))
     (SET P 3) (SET P 5))
    (GO P5)
    P3 (IF (EQ FSCHAR (QUOTE 'R))
     (BLOCK NIL (SET P 4) (GO P5)) (GO P2))
    P4 (SET P 5)
    P5 (CHREAD)
    (IF (EQ (F2 (CHARRAY (PLUS 1 (CHEAT SYMBOL INTEGER FSCHAR)
         (MINUS (CHEAT OCTAL INTEGER (CHO . SYS)))))) 1) (GO F1))
    (SET (XXSAVE . IO) FSCHAR)
    (SET FSCHAR NIL)
    (SET FSMSYM (MAKEST)) (SET CARCDR (EQ P 4)) (RETURN NIL))))
(CPER1 (FUNCTION (CPER1 NOVALUE)
   NIL (BLOCK NIL F1 (MAKEST)
    (CHREAD)
    (IF (EQ (F1 (CHARRAY (PLUS 1 (CHEAT SYMBOL INTEGER FSCHAR)
         (MINUS (CHEAT OCTAL INTEGER (CHO . SYS)))))) 9) (GO F1))
    (SET (XXSAVE . IO) FSCHAR)
    (SET FSCHAR NIL) (SET FSMSYM (MAKEST)) (RETURN NIL))))
(CPER2 (FUNCTION (CPER2 INTEGER)
   NIL (BLOCK NIL (MAKEST)
    (CHREAD)
    (SET (XXSAVE . IO) FSCHAR)
    (SET FSCHAR NIL)
    (SET FSMSYM (MAKEST))
    (CASE (PLUS 1 (F7 (CHARRAY (PLUS 1 (CHEAT SYMBOL INTEGER (XXSAVE
          . IO)) (MINUS (CHEAT OCTAL INTEGER (CHO . SYS)))))))
     (RETURN (CPRTR)) (RETURN (SIGN))))))
(DCTL (FUNCTION (DCTL BOOLEAN)
   NIL (BLOCK ((I INTEGER))
    (CHREAD)
    (MAKEST)
    (IF (AND (NQ (SET I (F1 (CHARRAY (PLUS (CHEAT SYMBOL INTEGER
           FSCHAR)
           (MINUS (CHEAT OCTAL INTEGER (CHO . SYS))) 1)))) 1)
      (NQ I 11))
     (BLOCK NIL (SET (XXSAVE . IO) FSCHAR)
      (SET FSCHAR NIL) (SET FSMSYM (MAKEST)) (RETURN FALSE))
     (BLOCK NIL G (CHREAD)
      (IF (EQ (F2 (CHARRAY (PLUS 1 (CHEAT SYMBOL INTEGER FSCHAR)
           (MINUS (CHEAT OCTAL INTEGER (CHO . SYS)))))) 1)
       (BLOCK NIL (MAKEST) (GO G))
       (BLOCK NIL (SET (XXSAVE . IO) FSCHAR)
        (SET FSCHAR NIL) (SET FSMSYM (MAKEST)) (RETURN TRUE))))))))
(TOKEN (FUNCTION (TOKEN INTEGER)
   NIL (BLOCK ((I INTEGER)
     (S INTEGER) (DP INTEGER) (ES SYMBOL) (K1 INTEGER) (K2 INTEGER))
    (CHREAD)
    (SET I (CHARRAY (PLUS 1 (CHEAT SYMBOL INTEGER FSCHAR)
       (MINUS (CHEAT OCTAL INTEGER (CHO . SYS))))))
    (CASE (F1 I)
     (GO C1)
     (GO C2)
     (GO C3)
     (GO C4)
     (GO C5) (GO C6) (GO C7) (GO C8) (GO C9) (GO C10) (GO C11))
    C1 (LTRL)
    (RETURN (LITRL))
    C2 (SET S 1)
    (SET FSMOCT 0Q)
    (SET FSMOCT (WORDOR (SHIFT FSMOCT 3)
      (DIFFERENCE (S20. FSCHAR) (S20. (QUOTE '0)))))
    C21 (SET K1 0)
    (SET K2 0)
    (GO I1S2)
    C3 (SET S 2)
    (GO C21)
    N (CHREAD)
    (SET I (CHEAT OCTAL INTEGER (F9 (CHARRAY (PLUS 1 (CHEAT SYMBOL
         INTEGER FSCHAR)
         (MINUS (CHEAT OCTAL INTEGER (CHO . SYS))))))))
    (CASE I (GO I1)
     (GO I2) (GO I3) (GO I4) (GO I5) (GO I6) (GO I7) (GO AU))
    I1 (CASE S (GO I1S1)
     (GO I1S2)
     (GO I1S2)
     (GO I1S2)
     (GO I1S5) (GO I1S6) (GO I1S2) (GO I1S6) (GO I1S9))
    I1S1 (SET FSMOCT (WORDOR (SHIFT FSMOCT 3)
      (DIFFERENCE (S20. FSCHAR) (S20. (QUOTE '0)))))
    I1S2 (SET K1 (PLUS (TIMES K1 10)
      (DIFFERENCE (S20. FSCHAR) (S20. (QUOTE '0)))))
    (MAKEST)
    (GO N)
    I1S5 (SET DP (PLUS 1 DP))
    (GO I1S2)
    I1S6 (SET S 7)
    (GO I1S2)
    I1S9 (SET S 3)
    (GO I1S2)
    I2 (CASE S (GO I2S1)
     (GO I1S2)
     (GO I1S2)
     (GO I1S2)
     (GO I1S5) (GO I1S6) (GO I1S2) (GO I1S6) (GO I1S9))
    I2S1 (SET S 2)
```

```
(GO I1S2)
I3 (MAKEST)
(CASE S (GO I3S1) (GO I3S1) (GO U) (GO U) (GO I3S5) (GO U))
I3S1 (SET S 9)
(SET DP 0)
I3S1A (SET K2 K1)
(SET K1 0)
(GO N)
I3S5 (SET ES (QUOTE '+))
(SET S 6)
(GO I3S1A)
I4 (MAKEST)
(CASE S (GO I4S1) (GO U))
I4S1 (SET S 4)
(GO I3S1A)
I5 (MAKEST)
(CASE S (GO I5S1) (GO I5S1) (GO U))
I5S1 (SET DP 0)
(SET S 5)
(GO N)
I6 (CASE S (GO IFIN)
 (GO IFIN)
 (GO IFINE) (GO OFIN) (GO RFIN) (GO AU) (GO RFINE) (GO AU))
I7 (CASE S (GO IFIN)
 (GO IFIN)
 (GO IFINE) (GO OFIN) (GO RFIN) (GO I7S6) (GO AU))
I7S6 (SET ES FSCHAR)
(SET S 8)
(MAKEST)
(GO N)
AU (MAKEST)
U (SET (XXSAVE . IO) FSCHAR)
(SET FSCHAR NIL)
(SET FSMSYM (MAKEST))
(RETURN (UNREC))
OFIN (SET FSMOCT (SHIFT FSMOCT (TIMES 3 K1)))
(SET I (UNSOCT))
NRET (SET (XXSAVE . IO) FSCHAR)
(SET FSCHAR NIL)
(SET FSMSYM (MAKEST))
(RETURN I)
IFINE (SET I K1)
(SET K1 K2)
(SET K2 I)
IFIN (SET I (UNSINT))
IFIN1 (IF (EQ 0 K2) (GO IFIN2))
(SET K2 (PLUS K2 -1))
(SET K1 (TIMES 10 K1))
(GO IFIN1)
IFIN2 (SET FSMOCT (CHEAT INTEGER OCTAL K1))
(GO NRET)
RFINE (SET I K1)
(SET K1 K2)
(SET K2 I)
RFIN (SET I (UNSREL))
(SET FSMREL (CHEAT REAL REAL K1))
(IF (EQ ES (QUOTE '-)) (SET K2 (MINUS K2)))
(SET DP (DIFFERENCE K2 DP))
(SET FSMREL (TIMES FSMREL (EXPT 10.0 DP)))
(GO NRET)
C4 (MAKEST)
(SET FSCHAR NIL)
(SET FSMSYM (MAKEST))
```

```
(RETURN (F3 I))
C5 (CHREAD)
(IF (NQ FSCHAR (QUOTE 'R)) (GO C5A))
C5R (CHREAD)
(CASE (F5 (CHARRAY (PLUS 1 (S2C. FSCHAR) (MINUS (CHO . SYS)))))
 (GO C5R1) (GO C5R2) (GO C5R3) (GO C5R))
C5R1 (SET (XXSAVE . IO) FSCHAR)
(GO C7)
C5R2 (MAKEST)
(GO C5R)
C5R3 (SET FSCHAR NIL)
(SET FSMSYM (MAKEST))
(RETURN (REMARK))
C5A (IF (NQ FSCHAR (QUOTE ' ))
 (GO C5B) (IF (STSP) (RETURN (STRNM)) (RETURN (UNREC))))
C5B (IF (NQ FSCHAR (QUOTE 'G)) (GO C5C))
(CHREAD)
(CASE (F1 (CHARRAY (PLUS 1 (CHEAT SYMBOL INTEGER FSCHAR)
    (MINUS (CHEAT OCTAL INTEGER (CHO . SYS))))))
 (GO C51)
 (GO C52)
 (GO C52)
 (GO C52)
 (GO C55)
 (GO C52) (GO C52) (GO C52) (GO C59) (GO C510) (GO C511))
C51 (LTRL)
(RETURN (GENSP))
C52 (SET (XXSAVE . IO) FSCHAR)
(GO C7)
C55 (IF (STSP) (RETURN (GENSP)) (RETURN (UNREC)))
C59 (OPER1)
(RETURN (GENSP))
C510 (IF (EQ (OPER2) (OPRTR))
 (RETURN (GENSP)) (RETURN (UNREC)))
C511 (MAKEST)
(IF (OCTL) (RETURN (GENSP)) (RETURN (UNREC)))
C5C (IF (NQ FSCHAR (QUOTE 'H)) (GO C5R1))
(SET FSMOCT 0)
C5CG (CHREAD)
(SET I (DIFFERENCE (S2C. FSCHAR) CHO))
(IF (AND (GQ I 6Q1) (LQ I 71Q))
 (SET I (PLUS I 777777777777717Q)) (GO C5C2))
C5C1 (SET FSMOCT (PLUS (SHIFT FSMOCT 4) I))
(MAKEST)
(GO C5CG)
C5C2 (IF (AND (GQ I 101Q) (LQ I 106Q))
 (BLOCK NIL (SET I (PLUS I 777777777777771Q1)) (GO C5C1))
 (BLOCK NIL (SET I (UNSOCT)) (GO NRET)))
C6 (IF (STSP) (RETURN (STRSP)) (RETURN (UNREC)))
C7 (MAKEST)
(SET FSCHAR NIL)
(SET FSMSYM (MAKEST))
(RETURN (UNREC))
C8 (SET FSMOCT 1)
(SET FSMSYM (QUOTE (*STRING ' )))
C81 (CHREAD)
(IF (EQ FSCHAR (QUOTE ' ))
 (BLOCK NIL (SET FSMOCT (PLUS FSMOCT 1)) (GO C81)))
(SET (XXSAVE . IO) FSCHAR)
(RETURN (SPACER))
C9 (OPER1)
(RETURN (OPRTR))
C10 (RETURN (OPER2))
```

```
    C11 (MAKEST)
    (CHREAD)
    (SET (XXSAVE . IO) FSCHAR)
    (IF (EQ (F2 (SET I (CHARRAY (PLUS 1 (CHEAT SYMBOL INTEGER FSCHAR)
          (MINUS (CHEAT OCTAL INTEGER (CH0 . SYS))))))) 0)
     (BLOCK NIL (SET FSCHAR NIL)
      (SET FSMSYM (MAKEST)) (RETURN (DOT))))
    (IF (OR (EQ (F1 I) 1) (EQ (F1 I) 11))
     (BLOCK NIL (IF (DOTL) (RETURN (DLITRL)) (RETURN (UNREC))))
     (BLOCK NIL (SET K1 0) (SET K2 0) (GO I5S1))))))

****END OF FILE DETECTED
```

```
(DUMMYS (SECTION SYS SYMBOL)
  (DECLARE ((PDBUF . GC) OCTAL OWN 100)
   ((PDADD . GC) OCTAL OWN 0) ((BPMIN . GC) OCTAL OWN 0))
  (DECLARE ((BACTRC . LISP) SYMBOL FREE)
   ((PRNERR . LISP) BOOLEAN FREE TRUE)
   ((INTERACT . LISP) BOOLEAN FLUID)
   ((SIGNON . LISP)
    SYMBOL OWN (QUOTE (*STRING 'L 'I 'S 'P '2 '  'N 'E 'W '  'C 'I 'G)
    )) ((MSGFILE . SUPV) SYMBOL OWN (QUOTE (OTTY))))
  (DECLARE ((FSCHAR . FSM) SYMBOL OWN NIL))
  (DECLARE (FMCALL (FUNCTIONAL NOVALUE) OWN)
   (GNLIST SYMBOL FREE NIL)
   (LAPSTOP BOOLEAN FREE)
   (AA (ARRAY OCTAL) FREE)
   (RR INTEGER FREE) (PP INTEGER FREE) (WW OCTAL FREE))
  (DECLARE ((LAPSTL . LAP) SYMBOL OWN)
   ((ERRFLG . SUPV) BOOLEAN FREE)
   ((FIXLOC . IO) OCTAL FLUID LOC)
   ((BUFIX . IO) (ARRAY OCTAL) FLUID))
  (DECLARE ((SECTOR . IO) INTEGER FLUID LOC))
  (DECLARE ((FILES. . IO) SYMBOL FREE)
   ((ICURFN . IO) SYMBOL FLUID)
   ((CURFN . IO) SYMBOL FLUID)
   ((TTY. . LISP) SYMBOL FREE)
   ((DISC. . LISP) SYMBOL FREE) ((TAPE. . LISP) SYMBOL FREE))
  (ROUTINE ((FXFN . GC) NOVALUE) ((X OCTAL) (B BOOLEAN)))
  (FUNCTION (SUPV SYMBOL) NIL)
  (FUNCTION (EVAL SYMBOL) ((E SYMBOL)))
  (FUNCTION ((MAKEST . FSM) SYMBOL) NIL)
  (FUNCTION ((CVRTNM . IO) INTEGER) ((A SYMBOL)))
  (FUNCTION (LAPGO SYMBOL) NIL)
  (ROUTINE (SYNTYPE OCTAL) ((A SYMBOL)))
  (ROUTINE (RDTYPC OCTAL) NIL)
  (ROUTINE (ADPOCK NOVALUE) ((X INTEGER)))
  (FUNCTION ((PRETTYP . LISP) SYMBOL) ((X SYMBOL)))
  (FUNCTION ((PRINWORD . LISP) OCTAL) ((X OCTAL)))
  (ROUTINE ((GET . LISP) SYMBOL) ((FN SYMBOL) (DL SYMBOL)))
  (FUNCTION ((ENDINR . LISP) NOVALUE) NIL)
  (FUNCTION ((ENDOUTR . LISP) NOVALUE) NIL)
  (FUNCTION ((LISP . LISP) SYMBOL)
   ((IN SYMBOL) (OUT SYMBOL) (FORM SYMBOL)))
  (DECLARE (ITTY SYMBOL OWN (QUOTE ITTY))
   (OTTY SYMBOL OWN (QUOTE OTTY))
   (INERR BOOLEAN FLUID FALSE)
   (TRACING BOOLEAN FREE FALSE)
   ((GC7 . GC) INTEGER OWN)
   (FREEZE BOOLEAN FREE FALSE)
   (INTCNT INTEGER OWN 3000)
   (SUPVFN (FUNCTIONAL SYMBOL SYMBOL SYMBOL SYMBOL) OWN)
   (TRYPT OCTAL FLUID 40002Q)
   (TRYVAR SYMBOL OWN)
   (EOFCH SYMBOL OWN) (XXCHAR SYMBOL OWN) (PDOUT SYMBOL OWN))
  (DECLARE (DUMPS SYMBOL OWN NIL) (LFILES SYMBOL OWN NIL)))
(INDEX (SECTION SYS SYMBOL)
  (ROUTINE (FNCALD FUNCTIONAL) ((RA OCTAL)))
  (ROUTINE (LARG SYMBOL) ((FN (FUNCTIONAL NOVALUE))))
  (ROUTINE (RESUME OCTAL)
   ((FN (FUNCTIONAL NOVALUE)) (MODE BOOLEAN)))
  (SECTION SYS SYMBOL)
  (FUNCTION (MESSAGE SYMBOL) ((M SYMBOL)))
  (ROUTINE ((MEMBERN . LISP) BOOLEAN) ((X SYMBOL) (L SYMBOL)))
  (FUNCTION ((DELE . LISP) SYMBOL) ((X SYMBOL) (L SYMBOL)))
  (FUNCTION (GETFN SYMBOL) ((N SYMBOL) (S SYMBOL)))
```

```
(FUNCTION (GETFRT SYMBOL) ((N SYMBOL) (S SYMBOL)))
(FUNCTION (GETFN1 SYMBOL) ((N SYMBOL) (S SYMBOL) (L SYMBOL)))
(FUNCTION (VARNAME SYMBOL) ((X SYMBOL)))
(ROUTINE (UNLDN SYMBOL) ((FN (FUNCTIONAL NOVALUE))))
(FUNCTION (FACTIVE BOOLEAN) ((FN (FUNCTIONAL NOVALUE))))
(SECTION SYS SYMBOL)
(FUNCTION ((ERROR . LISP) SYMBOL) ((S SYMBOL)))
(FUNCTION (CONDERR NOVALUE) NIL)
(FUNCTION ((EXIT . LISP) SYMBOL) ((S SYMBOL)))
(FUNCTION ((BACKFUNCTIONS . LISP) SYMBOL) ((I INTEGER)))
(FUNCTION (BACKUP SYMBOL) ((S SYMBOL) (I INTEGER) (M BOOLEAN)))
(ROUTINE (FLREST BOOLEAN)
 ((A INTEGER) (R INTEGER) (P INTEGER) (M BOOLEAN)))
(SECTION SYS SYMBOL)
(FUNCTION ((TRACEARGS . LISP) SYMBOL) ((L SYMBOL)))
(FUNCTION (TRACEA NOVALUE) NIL)
(FUNCTION ((TRACER . LISP) SYMBOL)
 ((N SYMBOL) (S SYMBOL) (FT (FUNCTIONAL NOVALUE))))
(FUNCTION ((UNTRACE . LISP) SYMBOL) ((L SYMBOL)))
(FUNCTION ((UNTRACER . LISP) SYMBOL) ((N SYMBOL) (S SYMBOL)))
(SECTION SYS SYMBOL)
(FUNCTION (FTRANS SYMBOL) ((P SYMBOL)))
(FUNCTION (ONTRAC NOVALUE)
 ((FN (FUNCTIONAL NOVALUE)) (FT (FUNCTIONAL NOVALUE))))
(FUNCTION (OFFTRAC SYMBOL) ((FN (FUNCTIONAL NOVALUE))))
(ROUTINE (SETFD NOVALUE) ((FD SYMBOL) (J INTEGER)))
(ROUTINE (SETRAP NOVALUE)
 ((FN (FUNCTIONAL NOVALUE)) (FT (FUNCTIONAL NOVALUE))))
(FUNCTION (UNDEFN NOVALUE) ((FN (FUNCTIONAL NOVALUE)) (S SYMBOL)))
(ROUTINE (VREFCT NOVALUE) ((V SYMBOL) (I INTEGER)))
(SECTION SYS SYMBOL)
(FUNCTION (FNTRAP NOVALUE) NIL)
(FUNCTION (FMTRAP NOVALUE) NIL)
(FUNCTION (LCTRAP NOVALUE) NIL)
(FUNCTION (PCGONE NOVALUE) NIL)
(FUNCTION (SCS SYMBOL) ((M SYMBOL)))
(SECTION SYS SYMBOL)
(FUNCTION ((EXCISE . LISP) SYMBOL) ((N SYMBOL) (S SYMBOL)))
(FUNCTION (EXCISF SYMBOL) ((FD SYMBOL)))
(ROUTINE (FXRUB OCTAL) ((I OCTAL)))
(SECTION SYS SYMBOL)
(FUNCTION (GETBPS INTEGER) ((I INTEGER)))
(FUNCTION (FITBPS INTEGER) ((I INTEGER)))
(FUNCTION (PACBPS NOVALUE) ((R INTEGER)))
(FUNCTION (UNLBPS BOOLEAN) ((I INTEGER)))
(FUNCTION (INTERRUPT OCTAL) NIL)
(FUNCTION (ITRAP1 NOVALUE) NIL)
(ROUTINE (FIXBPI NOVALUE) ((BP INTEGER) (A INTEGER) (B INTEGER)))
(SECTION SYS SYMBOL)
(FUNCTION ((DUMPSEC . LISP) SYMBOL)
 ((LNAME SYMBOL) (OUT SYMBOL) (SEC SYMBOL)))
(FUNCTION ((DUMPL . LISP) SYMBOL)
 ((LNAME SYMBOL) (OUT SYMBOL) (L SYMBOL)))
(FUNCTION (DUMPFN SYMBOL) ((X SYMBOL) (LSIZE INTEGER LOC)))
(FUNCTION ((UNLOADL . LISP) SYMBOL) ((LNAME SYMBOL)))
(FUNCTION ((UNLOADFN . LISP) SYMBOL) ((NAME SYMBOL) (SEC SYMBOL)))
(FUNCTION (UNLDFN SYMBOL) ((V SYMBOL) (LL SYMBOL)))
(SECTION SYS SYMBOL)
(FUNCTION ((LOADL . LISP) SYMBOL) ((LNAME SYMBOL)))
(FUNCTION ((OPENL . LISP) SYMBOL) ((X SYMBOL)))
(FUNCTION (READLIB INTEGER)
 ((FILE SYMBOL) (SECT INTEGER) (I INTEGER)))
(ROUTINE (INLIB INTEGER)
  ((BP INTEGER) (NAME INTEGER) (SECT INTEGER) (I INTEGER)))
 (FUNCTION (REMFN NOVALUE) ((FD SYMBOL)))
 (FUNCTION ((REMOVEL . LISP) SYMBOL) ((LNAME SYMBOL)))
 (SECTION SYS SYMBOL)
 (FUNCTION ((START . SYS) NOVALUE) NIL)
 (FUNCTION (SYSINI NOVALUE) NIL)
 (FUNCTION (PRBACK NOVALUE) ((X SYMBOL)))
 (FUNCTION (IOINI NOVALUE) NIL)
 (FUNCTION (RESTART NOVALUE) NIL)
 (FUNCTION (RECOV NOVALUE) NIL)
 (FUNCTION (LSUPV NOVALUE)
  ((INFILE SYMBOL) (OUTFILE SYMBOL) (FORM SYMBOL)))
 (SECTION LAP SYMBOL) (SECTION SYS OCTAL))
(MACROS DEFINE (((MSUBST (LAMBDA (L S)
     (SUBST (CADR L) (QUOTE ALPHA) S)))))
 MACRO1 (((VAR2FUNC (LAMBDA (L)
     (MSUBST L (QUOTE (O2F. (I2O. (PLUS (S2O. ALPHA) 2Q7 -1)))))))
   (FUNC2VAR (LAMBDA (L)
     (MSUBST L (QUOTE (O2S. (I2O. (PLUS (BIT 0 18 (F2O. ALPHA)) 1)))))
     )))
   (FUNCD (LAMBDA (L)
     (MSUBST L (QUOTE (CORE (BIT 0 18 (F2O. ALPHA)))))))
   (TRAPP (LAMBDA (L)
     (MSUBST L (QUOTE (O2S. (BIT 24 18 (FUNCD ALPHA)))))))
   (LADDR (LAMBDA (L) (MSUBST L (QUOTE (BIT 24 18 (CORE ALPHA))))))
   (RADDR (LAMBDA (L) (MSUBST L (QUOTE (BIT 0 18 (CORE ALPHA))))))
   (SLADDR (LAMBDA (L)
     (MSUBST L (SUBST (GENID)
       (QUOTE X)
       (QUOTE (BLOCK ((X OCTAL LEXICAL (CORE ALPHA)))
         (RETURN (CODE (LDA X (L567.3 S))))))))))
   (SRADDR (LAMBDA (L)
     (MSUBST L (SUBST (GENID)
       (QUOTE X)
       (QUOTE (BLOCK ((X OCTAL LEXICAL (CORE ALPHA)))
         (RETURN (CODE (LDA X (L567.7 S))))))))))
   (FNSTAT (LAMBDA (L)
     (MSUBST L (QUOTE (SHIFT (FUNCD ALPHA) -45)))))
   (FREADY (LAMBDA (L)
     (MSUBST L (QUOTE (EQ (WORDAND (FUNCD ALPHA) 2Q15) 0)))))
   (IDLINK (LAMBDA (L)
     (MSUBST L (QUOTE (O2S. (RADDR (PLUS (S2O. ALPHA) 1)))))))
   (AGE (LAMBDA (L)
     (SUBST (CDR L)
      (QUOTE A) (QUOTE (BIT 18 3 (CORE (S2O. . A)))))))
   (TRAPEQ (LAMBDA (L)
     (SUBST (CADR L)
      (QUOTE FN)
      (SUBST (CADDR L)
       (QUOTE FT) (QUOTE (EQ (O2F. (BIT 0 24 (FUNCD FN))) FT)))))))
   (ROUTINEDIES (LAMBDA (L) NIL))))
 MACRO1 (((TRAPM (LAMBDA (L)
     (SUBST (CADDDR L)
      (QUOTE EXP)
      (SUBST (CADR L)
       (QUOTE FNAME)
       (SUBST (CADDR L)
        (QUOTE MODE)
        (QUOTE (BLOCK ((ALA. OCTAL (CODE)))
          (BLOCK ((FNAME (FUNCTIONAL NOVALUE)
             ((FNCALD . SYS) (CODE (LDA 0 8)))))
           (BLOCK ((SLA. SYMBOL (IF (EQ ((LARG . SYS) FNAME)
                (QUOTE S)) (O2S. ALA.) NIL)))
             EXP (BLOCK ((CA OCTAL ((RESUME . SYS) FNAME MODE)))
              (SET (FMCALL . SYS) FNAME)
              (IF SLA. (SET ALA. (S2C. SLA.)))
              (CODE (LDA ALA.) (BUC CA I))))))))))))
    (TRACEM (LAMBDA (L)
      (LIST (QUOTE (TRAPM . SYS)) (GENSYM) TRUE (CADR L)))))))
(TRAPM (SECTION SYS SYMBOL)
  (ROUTINE (FNCALD FUNCTIONAL)
   ((RA OCTAL))
   (BLOCK ((FA OCTAL (RADDR (PLUS RA -1))))
    (IF (EQ (BIT 42 6 (CORE FA)) 0) (SET FA (RADDR FA)))
    (RETURN (O2F. (WORDCR 2Q7 FA)))))
  (ROUTINE (LARG SYMBOL)
   ((FN (FUNCTIONAL NOVALUE)))
   (BLOCK ((WW OCTAL FREE (SYNTYPE (FUNC2VAR FN)))
     (PP INTEGER FREE 12)
     (RR INTEGER FREE 1)
     (AA (ARRAY OCTAL) FREE) (LA SYMBOL) (C OCTAL))
    (IF (EQ (BIT 24 6 WW) 1)
     (BLOCK NIL (SET AA (O2S. (BIT 6 18 WW)))
      (SET WW (AA 1)) (SET PP 30)))
    (FOR C (LOOP (RDTYPC))
     (WHILE (NQ C 63))
     (SET LA (IF (OR (RELATION 0 LS C LS 5)
        (AND (EQ C 31) (RDTYPC))) (QUOTE A) (QUOTE S))))
    (RETURN LA)))
  (ROUTINE (RESUME OCTAL)
   ((FN (FUNCTIONAL NOVALUE)) (MODE BOOLEAN))
   (BLOCK ((CP OCTAL))
    A (IF (EQ (FNSTAT FN) 0)
     (SET CP (BIT 0 18 (FUNCD FN)))
     (AND (FREADY FN) MODE)
     (SET CP (BIT 24 18 (FUNCD FN)))
     (BLOCK NIL (SET FN (O2F. (BIT 0 24 (FUNCD FN)))) (GO A)))
    (RETURN (IF (LARG FN) (PLUS CP 2) (PLUS CP 1))))))
(SUBFNS (SECTION SYS SYMBOL)
  (FUNCTION (MESSAGE SYMBOL)
   ((M SYMBOL))
   (BLOCK ((K SYMBOL))
    (FOR K (IN (MSGFILE . SUPV))
     (BLOCK ((OUT SYMBOL (OUTPUT K))) (PRETTYP M) (OUTPUT OUT)))
    (RETURN M)))
  (ROUTINE ((MEMBERN . LISP) BOOLEAN)
   ((X SYMBOL) (L SYMBOL))
   (BLOCK ((Y SYMBOL)) (FOR Y (IN L) (IF (EQN X Y) (RETURN TRUE)))))
  (FUNCTION ((DELE . LISP) SYMBOL)
   ((X SYMBOL) (L SYMBOL))
   (BLOCK ((P SYMBOL) (M SYMBOL))
    (FOR L (ON L) (IF (NQ (CAR (SET P L)) X) (GO A)))
    (RETURN NIL)
    A (IF (NULL (SET M (CDR P)))
     (RETURN L) (EQ (CAR M) X) (SET (CDR P) (CDR M)) (SET P M))
    (GO A)))
  (FUNCTION (GETFN SYMBOL)
   ((N SYMBOL) (S SYMBOL))
   (GETFN1 N S (QUOTE (FUNCTION MACRO INSTRUCTIONS))))
  (FUNCTION (GETFRT SYMBOL)
   ((N SYMBOL) (S SYMBOL))
   (GETFN1 N S (QUOTE (FUNCTION MACRO INSTRUCTIONS ROUTINE))))
  (FUNCTION (GETFN1 SYMBOL)
   ((N SYMBOL) (S SYMBOL) (L SYMBOL))
   (BLOCK ((X SYMBOL (GETFREE N S)))
    (IF (AND X (MEMBER (FVKIND X) L)) (RETURN X))))
  (FUNCTION (VARNAME SYMBOL)
   ((X SYMBOL))
   (CONS (BLOCK ((Y SYMBOL X))
     (FOR Y (LOOP (IDLINK Y))
      (UNLESS (NQ (BIT 42 6 (CORE (S20. Y))) 7)) (RETURN Y)))
    (O2S. (LADDR (S20. X)))))
  (ROUTINE (UNLDN SYMBOL)
   ((FN (FUNCTIONAL NOVALUE)))
   (IF (FREADY FN)
    NIL (ATOM (TRAPP FN)) (TRAPP FN) (CDR (TRAPP FN))))
  (FUNCTION (FACTIVE BOOLEAN)
   ((FN (FUNCTIONAL NOVALUE)))
   (BLOCK ((BP INTEGER (PLUS (BIT 0 18 (FUNCD FN)) -1))
     (P INTEGER) (R INTEGER) (A INTEGER))
    (CODE (STX P 0 8))
    A (FOR A (RESET (CORE P) (RADDR A))
     (WHILE (NQ (BIT 18 6 (CORE A)) 1)) NIL)
    (IF (EQ A BP)
     (RETURN TRUE)
     (LS (SET P (PLUS P (LADDR (PLUS (CORE P) -1)))) BPO) (GO A)))))
(ERROR (SECTION SYS SYMBOL)
  (FUNCTION ((ERROR . LISP) SYMBOL)
   ((S SYMBOL))
   (BLOCK NIL (IF (AND (NOT INERR) (OR (PRNERR . LISP) INTERACT))
     (BLOCK (((INERR . SYS) TRUE)) (MESSAGE S)))
    (IF (NOT INTERACT) (EXIT S))
    (RETURN (SUPVFN ITTY OTTY (QUOTE IL)))))
  (FUNCTION (CONDERR NOVALUE)
   NIL (ERROR (QUOTE (IF EXPRESSION UNSATISFIED))))
  (FUNCTION ((EXIT . LISP) SYMBOL)
   ((S SYMBOL))
   (BLOCK NIL (SET (BACTRC . LISP) NIL) (BACKUP S 50000 TRUE)))
  (FUNCTION ((BACKFUNCTIONS . LISP) SYMBOL)
   ((I INTEGER))
   (BLOCK (((BACTRC . LISP) SYMBOL FREE))
    (BACKUP NIL (PLUS I 1) FALSE) (RETURN (CDR (BACTRC . LISP)))))
  (FUNCTION (BACKUP SYMBOL)
   ((S SYMBOL) (I INTEGER) (M BOOLEAN))
   (BLOCK ((A INTEGER) (R INTEGER) (P INTEGER))
    (CODE (STX P 0 8))
    (FOR I (STEP I -1 EQ 0)
     (WHILE (LS P BPO))
     (BLOCK NIL (SET R (CORE P))
      (SET P (PLUS P (SLADDR (PLUS R -1))))
      (FOR A (RESET R (RADDR A))
       (WHILE (NQ (BIT 18 6 (CORE A)) 1)) NIL)
      (IF (EQ (RADDR A) 0)
       (GO L) (AND M (FLREST A R P TRUE)) (GO BACK))
      (SET P (PLUS P (MINUS BPO)))
      (SET (BACTRC . LISP)
       (CONS (VARNAME (O2S. (I20. (PLUS (RADDR A) 1))))
        (BACTRC . LISP))) (SET P (PLUS P BPO)) L))
    (IF (NOT M) (RETURN S))
    BACK (SET A (PLUS A (TRYPT . SYS)))
    (CODE (LDA S) (LDX A 0 4) (LDX P 0 8) (BUC 0 4))))
  (LAP (PATCH (ORG)
    (ENTRY FLRCAL (LABEL A))
    A (BSX (ENTRY FLREST) 7) (END)) NIL SYS)
  (ROUTINE (FLREST BOOLEAN)
   ((A INTEGER) (R INTEGER) (P INTEGER) (M BOOLEAN))
   (BLOCK ((B INTEGER)
     (BC INTEGER) (J INTEGER) (K INTEGER) (V INTEGER))
    (SET B (PLUS A (LADDR A)))
    (FOR A (STEP A 1)                                                      0031600
     (WHILE (LS A B))                                                      0031700
     (BLOCK NIL (IF (LS (SET BC (PLUS BC -2)) 0)                           0031800
       (BLOCK NIL (SET B (PLUS B -1)) (SET BC 46)))                        0031900
      (IF (NOT (AND (GR A R)                                               0032000
         (EQ (WORDAND (CORE A) 7700000077777777Q) (CORENTRY FLRCAL))       0032100
         (EQ (WORDAND (SHIFT (CORE B) (MINUS BC)) 3Q) 2)                   0032200
         (LS (SET J (LADDR A)) R))) (GO L))                                0032300
      (SET K (PLUS P (SLADDR (PLUS J -1))))                                0032400
      (FOR J (STEP J 1)                                                    0032500
       (BLOCK NIL (SET K (PLUS K -1))                                      0032600
        (SET V (RADDR J))                                                  0032700
        (IF (AND M (EQN (O2S. (I2O. (PLUS V 1))) TRYVAR))                  0032800
         (RETURN TRUE))                                                    0032900
        (SET (CORE V) (CORE K))                                            0033000
        (IF (EQ (BIT 43 1 (CORE J)) 1) (GO L)))) L)))))                    0033100
(TRACE (SECTION SYS SYMBOL)                                                0033200
  (FUNCTION ((TRACEARGS . LISP) SYMBOL)                                    0033300
   ((L SYMBOL))                                                            0033400
   (MAPCAR L (FUNARG SYMBOL ((X SYMBOL))                                   0033500
     (IF (ATOM X) NIL (TRACER (CAR X) (CDR X) TRACEA)))))                  0033600
  (FUNCTION (TRACEA NOVALUE)                                               0033700
   NIL (TRAPM FN TRUE (IF (NOT TRACING)                                    0033800
     (BLOCK ((V SYMBOL (FUNC2VAR FN))                                      0033900
       (TRACING BOOLEAN FREE TRUE) (L SYMBOL) (J INTEGER))                 0034000
      (SET L (CDDADR (FVLIST V)))                                          0034100
      (SET J (LENGTH L))                                                   0034200
      (MESSAGE (CONS (VARNAME V)                                           0034300
        (QUOTE OF)                                                         0034400
        (MAPCAR L (FUNARG SYMBOL ((X SYMBOL))                              0034500
          (BLOCK ((W OCTAL))                                               0034600
           (SET W (IF (EQ (SET J (PLUS J -1)) 0)                           0034700
             (IF SLA. (S2O. SLA.) ALA.)                                    0034800
             (CORE (PLUS (CODE (LDA ALA.)) J 1))))                         0034900
           (IF (EQ (CADR X) (QUOTE LOC)) (SET W (CORE W)))                 0035000
           (SET X (CAR X))                                                 0035100
           (RETURN (IF (EQ X (QUOTE OCTAL))                                0035200
             W (EQ X (QUOTE INTEGER))                                      0035300
             (O2I. W)                                                      0035400
             (EQ X (QUOTE REAL))                                           0035500
             (O2R. W)                                                      0035600
             (EQ X (QUOTE FUNCTIONAL)) (O2F. W) (O2S. W))))))))))))        0035700
  (FUNCTION ((TRACER . LISP) SYMBOL)                                       0035800
   ((N SYMBOL) (S SYMBOL) (FT (FUNCTIONAL NOVALUE)))                       0035900
   (BLOCK ((X SYMBOL (GETFN N S)))                                         0036000
    (IF (NULL X) (RETURN NIL))                                             0036100
    (ONTRAC (VAR2FUNC X) FT) (RETURN (CONS N S))))                         0036200
  (FUNCTION ((UNTRACE . LISP) SYMBOL)                                      0036300
   ((L SYMBOL))                                                            0036400
   (MAPCAR L (FUNARG SYMBOL ((X SYMBOL))                                   0036500
     (IF (ATOM X) NIL (UNTRACER (CAR X) (CDR X))))))                       0036600
  (FUNCTION ((UNTRACER . LISP) SYMBOL)                                     0036700
   ((N SYMBOL) (S SYMBOL))                                                 0036800
   (BLOCK ((X SYMBOL (GETFN N S)))                                         0036900
    (RETURN (IF (OR (NULL X) (NULL (OFFTRAC (VAR2FUNC X))))                0037000
      NIL (CONS N S))))))                                                  0037100
 ENCD (SECTION SYS SYMBOL)                                                 0037200
  (FUNCTION (FTRANS SYMBOL)                                                0037300
   ((P SYMBOL))                                                            0037400
   (BLOCK ((FN (FUNCTIONAL NOVALUE) (VAR2FUNC (CAR P)))                    0037500
     (FNT (FUNCTIONAL NOVALUE) (VAR2FUNC (CDR P)))                         0037600
     (FREEZE BOOLEAN FREE TRUE))                                           0037700
    (BLOCK ((S SYMBOL) (J INTEGER))                                        0037800
     (IF (NOT (FREADY FNT))
      (IF (NULL (SET S (UNLDN FNT))) (RETURN NIL) (LOADL S)))
     (REMFN (CDR P))
     (SET S (OFFTRAC FNT))
     (EXCISF (CAR P))
     (SET J (PLUS (BIT 0 18 (FUNCD FNT)) -1))
     (SET (RADDR J) (BIT 0 18 (F2O. FN)))
     (SETFD (CAR P) J)
     (VREFCT (CAR P) 1)
     (VREFCT (CDR P) -1)
     (UNDEFN FNT S) (RETURN (VARNAME (CAR P))))))
  (FUNCTION (ONTRAC NOVALUE)
   ((FN (FUNCTIONAL NOVALUE)) (FT (FUNCTIONAL NOVALUE)))
   (IF (FREADY FN)
    (SETRAP FN FT)
    (SET (TRAPP FN)
     (CONS FT (IF (ATOM (TRAPP FN)) (TRAPP FN) (CDR (TRAPP FN)))))))
  (FUNCTION (OFFTRAC SYMBOL)
   ((FN (FUNCTIONAL NOVALUE)))
   (BLOCK ((S SYMBOL))
    (IF (NOT (FREADY FN))
     (IF (NOT (ATOM (TRAPP FN)))
      (BLOCK NIL (SET S (CAR (TRAPP FN)))
       (SET (TRAPP FN) (CDR (TRAPP FN)))))
     (IF (NQ (FNSTAT FN) 0))
      (BLOCK NIL (SET S (C2F. (BIT 0 24 (FUNCD FN))))
       (SET (FUNCD FN)
        (WORDOR (WORDAND 7Q14 (FUNCD FN))
         (BIT 24 18 (FUNCD FN))))))) (RETURN S)))
  (ROUTINE (SETFD NOVALUE)
   ((FD SYMBOL) (J INTEGER))
   (BLOCK ((FN (FUNCTIONAL NOVALUE) (VAR2FUNC FD)) (S SYMBOL))
    (SET S (TRAPP FN))
    (SET (FUNCD FN)
     (WORDOR (WORDAND 7Q14 (FUNCD FN)) (I2O. (PLUS J 1))))
    (IF (NOT (ATOM S))
     (SETRAP FN (O2F. (CORE (PLUS (S2O. (CAR S)) 1)))))))
  (ROUTINE (SETRAP NOVALUE)
   ((FN (FUNCTIONAL NOVALUE)) (FT (FUNCTIONAL NOVALUE)))
   (BLOCK NIL (IF (EQ (FNSTAT FN) 0)
     (BLOCK NIL (SET (BIT 45 3 (FUNCD FN)) 1Q)
      (SET (BIT 24 18 (FUNCD FN)) (BIT 0 18 (FUNCD FN)))))
    (SET (BIT 0 24 (FUNCD FN)) (BIT 0 24 (F2O. FT)))))
  (FUNCTION (UNDEFN NOVALUE)
   ((FN (FUNCTIONAL NOVALUE)) (S SYMBOL))
   (BLOCK NIL (SET (FUNCD FN)
     (WORDOR 2Q15 (WORDAND 7Q14 (FUNCD FN)) (F2O. (FNTRAP . SYS))))
    (IF S (SET (TRAPP FN) (LIST S)))))
  (ROUTINE (VREFCT NOVALUE)
   ((V SYMBOL) (I INTEGER))
   (SET (RADDR (S2O. V)) (PLUS (RADDR (S2O. V)) I))))
(FNTRAP (SECTION SYS SYMBOL)
  (FUNCTION (FNTRAP NOVALUE)
   NIL (TRAPM FN NIL (BLOCK ((X SYMBOL (FUNC2VAR FN)))
     (ROUTINEDIES FN)
     (ERROR (CONS (VARNAME X) (QUOTE UNDEFINED) (FVLIST X))))))
  (FUNCTION (FMTRAP NOVALUE)
   NIL (ERROR (QUOTE (UNSET FUNCTIONAL VARIABLE APPLIED))))
  (FUNCTION (LDTRAP NOVALUE)
   NIL (TRAPM FN NIL (BLOCK ((X SYMBOL (TRAPP FN)))
     (IF (NOT (ATOM X)) (SET X (CDR X))) (LOADL X))))
  (FUNCTION (PCGONE NOVALUE)
   NIL (TRAPM FN NIL (BLOCK ((J INTEGER (CODE (LDA (Z. 8))))
      (P SYMBOL PDOUT))
     (ADPDCK (MINUS (PDBUF . GC)))
     (SET PDOUT (QUOTE B))
     (IF (EQ P (QUOTE B)) (SOS (QUOTE PDGONE)))
     (IF (NULL P) (RECLAIM 0))
     (SET (PDACC . GC) 0)
     (IF (NULL P) (MESSAGE (QUOTE (STACK GREW)))))))
  (FUNCTION (SOS SYMBOL)
   ((M SYMBOL))
   (BLOCK NIL (TRY M E (MESSAGE (LIST (QUOTE SOS) M))) E (CODE (0)))))
(EXCISE (SECTION SYS SYMBOL)
  (FUNCTION ((EXCISE . LISP) SYMBOL)
   ((N SYMBOL) (S SYMBOL))
   (BLOCK ((X SYMBOL (GETPRT N S)))
    (RETURN (IF (NULL X) NIL (EXCISF X)))))
  (FUNCTION (EXCISF SYMBOL)
   ((FD SYMBOL))
   (BLOCK ((FN (FUNCTIONAL NOVALUE) (VAR2FUNC FD))
     (S SYMBOL) (J INTEGER) (FREEZE BOOLEAN FREE TRUE))
    (IF (NOT (FREADY FN))
     (IF (NULL (SET S (UNLDN FN))) (RETURN NIL) (LOADL S)))
    (SET S (OFFTRAC FN))
    (IF (FACTIVE FN)
     (BLOCK NIL (IF S (ONTRAC FN S))
      (ERROR (CONS (VARNAME FD) (QUOTE (ACTIVE CANT EXCISE))))))
    (REMFN FD)
    (SET J (PLUS (BIT 0 18 (FUNCD FN)) -1))
    (SET (BIT 42 6 (CORE J)) 0Q)
    (IF (GQ J BPO) ((FXFN . GC) J TRUE) (GO A))
    (IF (EQ (PLUS J (LADDR J)) BPP)
     (SET BPP J)
     (SET (BPMIN . GC) (PLUS (BPMIN . GC) (MINUS (LADDR J))))))
    A (UNDEFN FN S) (RETURN (VARNAME FD))))
  (ROUTINE (FXRUB OCTAL)
   ((I OCTAL))
   (BLOCK ((J OCTAL (BIT 42 6 (CORE (PLUS I 1)))))
    (IF (LS I TRO)
     (GO R)
     (AND (LQ 7 J) (LQ J 31))
     (SET I (I2O. (PLUS I 1))) (SET J (BIT 42 6 (CORE I))))
    (IF (EQ J 7)
     (SET (LADDR (PLUS I 1)) (PLUS (LADDR (PLUS I 1)) -1))
     (SET (RADDR I) (PLUS (RADDR I) -1))) R)))
(GETBPS (SECTION SYS SYMBOL)
  (FUNCTION (GETBPS INTEGER)
   ((I INTEGER))
   (BLOCK ((BP INTEGER) (J INTEGER) (K INTEGER))
    A (SET J (PLUS ARO (MINUS BPP)))
    (SET K (PLUS J (MINUS (BPMIN . GC))))
    (IF (AND (LS (PLUS I (BPMIN . GC)) 0)
      (NQ (SET BP (FITBPS I)) 0))
     (GO R)
     (GQ J I)
     (BLOCK NIL (SET BP BPP) (SET BPP (PLUS BPP I)) (GO R)))
    (INTERRUPT)
    (IF (GQ K I)
     (BLOCK NIL (PACBPS 0) (GO A))
     (AND (NOT FREEZE)
      (GR (DIFFERENCE BPP BPO) (GC7 . GC)) (UNLBPS I))
     (GO A)
     (BLOCK NIL (SET (BPMIN . GC) (PLUS (BPMIN . GC) I))
      (RECLAIM 0) (GO A)))
    R (SET (CORE BP) 0) (SET (LADDR BP) I) (RETURN BP)))
```

```
(FUNCTION (FITBPS INTEGER)                                               C050500
 ((I INTEGER))                                                           C050600
 (BLOCK ((BP INTEGER)                                                    C050700
   (BS INTEGER) (BMP INTEGER) (BMS INTEGER 50000))                       C050800
  (FOR BP (RESET BPO (PLUS BP (LADDR BP)))                               C050900
   (WHILE (NQ BP BPP))                                                   C051000
   (IF (EQ (BIT 42 6 (CORE BP)) 0)                                       C051100
    (SET BS (PLUS BS (LADDR BP)))                                        C051200
    (AND (GQ BS I) (LS BS BMS))                                          C051300
    (BLOCK NIL (SET BMP (PLUS BP (MINUS BS)))                            C051400
     (SET BMS BS) (SET BS 0))) (SET BS 0)))                              C051500
  (SET (BPMIN . GC) (PLUS (BPMIN . GC) BS))                              C051600
  (SET BPP (PLUS BPP (MINUS BS)))                                        C051700
  (IF (EQ BMS 50000)                                                     C051800
   (RETURN 0)                                                            C051900
   (GR BMS I)                                                            C052000
   (BLOCK NIL (SET (CORE (PLUS BMP I)) 0)                                C052100
    (SET (LADDR (PLUS BMP I)) (PLUS BMS (MINUS I)))))                    C052200
  (SET (BPMIN . GC) (PLUS (BPMIN . GC) 1)) (RETURN BMP)))                C052300
(FUNCTION (PACBPS NOVALUE)                                               C052400
 ((R INTEGER))                                                           C052500
 (ORG NIL (BLOCK ((A INTEGER)                                            C052600
    (P INTEGER) (D INTEGER) (I INTEGER))                                 C052700
   REASSIGN (SET A (PLUS BPO R 1))                                       C052800
   (FOR P (RESET BPO (PLUS P (LADDR P)))                                 C052900
    (WHILE (NQ P BPP))                                                   C053000
    (UNLESS (EQ (BIT 42 6 (CORE P)) 0))                                  C053100
    (BLOCK NIL (SET D (RADDR P))                                         C053200
     (IF (NQ (WORDAND 1Q15 (CORE D)) 0)                                  C053300
      (SET (LADDR D) (I2C. A)) (SET (RADDR D) (I2C. A)))                 C053400
     (SET A (PLUS A (LADDR P)))))                                        C053500
   FIXRETS (CODE (STX P 0 8))                                            C053600
   (FOR P (RESET P (PLUS P I))                                           C053700
    (WHILE (NQ P BPO))                                                   C053800
    (BLOCK NIL (FOR A (RESET (CORE P) (RADDR A))                         C053900
      (WHILE (NQ (BIT 18 6 (CORE A)) 1)) NIL)                            C054000
     (SET I (SLADDR (PLUS (CORE P) -1)))                                 C054100
     (IF (NQ (SET D (RADDR A)) 0)                                        C054200
      (SET (CORE P)                                                      C054300
       (PLUS (CORE P)                                                    C054400
        (MINUS A)                                                        C054500
        -1 (IF (NQ (WORDAND 1Q15 (CORE D)) 0)                            C054600
         (LADDR D) (RADDR D)))))))                                       C054700
   MOVEBPIS (SET P BPO)                                                  C054800
   (SET BPO (SET A (PLUS P (IF (GR R 0) 0 R))))                          C054900
   (FOR P (RESET P (PLUS P I))                                           C055000
    (WHILE (NQ P BPP))                                                   C055100
    (UNLESS (BLOCK NIL (SET I (LADDR P))                                 C055200
      (RETURN (EQ (BIT 42 6 (CORE P)) 0))))                              C055300
    (IF (EQ A P)                                                         C055400
     (SET A (PLUS A (LADDR P)))                                          C055500
     (BLOCK NIL (FIXBPI P P (PLUS A R))                                  C055600
      (FOR D (STEP P 1 EQ (PLUS P (LADDR P)))                            C055700
       (BLOCK NIL (SET (CORE A) (CORE D)) (SET A (PLUS A 1)))))))  0055800
   (SET BPP A) (SET (BPMIN . GC) 0))))                                   C055900
(FUNCTION (UNLBPS BOOLEAN)                                               C056000
 ((I INTEGER))                                                           C056100
 (BLOCK ((N INTEGER)                                                     C056200
   (X SYMBOL) (V SYMBOL) (FN (FUNCTIONAL NOVALUE)))                      C056300
  (FOR N (STEP 7 -1 EQ -2)                                               C056400
   (IF (LQ I (PLUS ARC (MINUS BPP) (MINUS (BPMIN . GC))))                C056500
    (RETURN TRUE)                                                        C056600
   (FOR X (IN DUMPS)                                                     C056700
      (FOR V (IN (CADDR X))
       (IF (EQ (FNSTAT (SET FN (VAR2FUNC V))) 1)
        (IF (AND (EQ (AGE V) N) (TRAPEQ FN ITRAP1))
         (BLOCK NIL (SET (FUNCD FN)
           (WORDOR (WORDAND 7Q14 (FUNCD FN))
            (BIT 24 18 (FUNCD FN)))) (UNLDFN V X)))
        (AND (LS N 0) (FREADY FN)) (UNLDFN V X))))))
   (RETURN FALSE)))
  (FUNCTION (INTERRUPT OCTAL)
   NIL (BLOCK ((ALA. OCTAL (CODE)))
    (BLOCK ((X SYMBOL) (V SYMBOL) (FN (FUNCTIONAL NOVALUE)))
     (FOR X (IN DUMPS)
      (FOR V (IN (CADDR X))
       (UNLESS (NOT (FREADY (SET FN (VAR2FUNC V)))))
       (IF (EQ (FNSTAT FN) 0)
        (BLOCK NIL (SETRAP FN ITRAP1) (SET (AGE V) 0))
        (TRAPEQ FN ITRAP1)
        (IF (LS (AGE V) 7) (SET (AGE V) (PLUS (AGE V) 1))))))
     (CODE (LDX (INTCNT . SYS) 0 6)) (SET INTCNT INTCNT))
    (RETURN ALA.)))
  (FUNCTION (ITRAP1 NOVALUE)
   NIL (BLOCK ((ALA. OCTAL (CODE)))
    (BLOCK ((FN (FUNCTIONAL NOVALUE)
       ((FNCALC . SYS) (CODE (LDA 0 8)))))
     (SET (FUNCD FN)
      (WORDOR (WORDAND 7Q14 (FUNCD FN)) (BIT 24 18 (FUNCD FN))))
     (BLOCK ((CA OCTAL ((RESUME . SYS) FN NIL)))
      (SET (FMCALL . SYS) FN) (CODE (LDA ALA.) (BUC CA I))))))
  (ROUTINE (FIXBPI NOVALUE)
   ((BP INTEGER) (A INTEGER) (B INTEGER))
   (ORG NIL (BLOCK ((I INTEGER (LADDR BP))
      (D INTEGER (PLUS B (MINUS A))))
     (BLOCK ((BE INTEGER (PLUS BP I)) (BW OCTAL) (C INTEGER 0))
      (FOR BP (STEP BP 1)
       (WHILE (LS BP BE))
       (BLOCK NIL (IF (LS (SET C (PLUS C -1)) 0)
         (BLOCK NIL (SET BE (PLUS BE -1))
          (SET BW (CORE BE)) (SET C 23)))
        (IF (AND (NQ (WORDAND BW 4Q15) 0)
          (LQ A (LADDR BP)) (LS (LADDR BP) (PLUS A I)))
         (SET (LADDR BP) (PLUS (LADDR BP) D)))
        (IF (AND (NQ (WORDAND BW 2Q15) 0)
          (LQ A (RADDR BP)) (LS (RADDR BP) (PLUS A I)))
         (SET (RADDR BP) (PLUS (RADDR BP) D)))
        (SET BW (SHIFT BW 2)))))))))
(DUMP (SECTION SYS SYMBOL)
  (FUNCTION ((DUMPSEC . LISP) SYMBOL)
   ((LNAME SYMBOL) (OUT SYMBOL) (SEC SYMBOL))
   (BLOCK ((I INTEGER 1) (L SYMBOL) (A SYMBOL))
    (FOR I (STEP I 1 GR OBLSIZ)
     (FOR A (RESET (OBLIST I) (IDLINK A))
      (WHILE A) (IF (GETFREE A SEC) (SET L (CONS (CONS A SEC) L)))))
    (RETURN (DUMPL LNAME OUT L))))
  (FUNCTION ((DUMPL . LISP) SYMBOL)
   ((LNAME SYMBOL) (OUT SYMBOL) (L SYMBOL))
   (BLOCK ((LSIZE INTEGER 0)
     (B SYMBOL) (LF SYMBOL) (S SYMBOL) (M SYMBOL) (X SYMBOL))
    (IF (NOT (AND (IDP LNAME) (IDP OUT) (LISTP L)))
     (ERROR (QUOTE (BAD ARGS TO DUMPL))))
    (OPENL OUT)
    (SET B (GET (QUOTE BUF) (GET OUT (FILES. . IO))))
    (SET LF (FINDN OUT LFILES))
    (SET (CDR B) (CREATE 512 (QUOTE OCTAL) 0))
    (SET S (OUTPUT OUT))
    (SET (SECTOR . IO) (CDR LF))
    (IF (FINDN LNAME DUMPS) (SET L (APPEND (REMOVEL LNAME) L)))
    (FOR X (IN L) (IF (SET X (DUMPFN X LSIZE)) (SET M (CONS X M))))
    (SET M (REVERSE M))
    (ENDOUTR)
    (SET DUMPS (CONS (LIST LNAME (LIST OUT (CDR LF) LSIZE) M NIL)
      DUMPS))
    (SET (CDR LF) (SECTOR . IO))
    (OUTPUT S) (SET (CDR B) NIL) (RETURN (MAPCAR M VARNAME))))
  (FUNCTION (DUMPFN SYMBOL)
   ((X SYMBOL) (LSIZE INTEGER LOC))
   (BLOCK ((V SYMBOL)
     (FN (FUNCTIONAL NOVALUE))
     (S SYMBOL)
     (A (ARRAY OCTAL))
     (BP INTEGER) (I INTEGER) (FREEZE BOOLEAN FREE TRUE))
    (IF (OR (ATOM X) (NULL (SET V (GETFN (CAR X) (CDR X)))))
     (RETURN NIL))
    (SET FN (VAR2FUNC V))
    (IF (NOT (FREADY FN))
     (IF (NULL (SET S (UNLDN FN))) (RETURN NIL) (LOADL S)))
    A (SET BP (PLUS -1 (IF (EQ (FNSTAT FN) 0)
       (BIT 0 18 (FUNCD FN)) (BIT 24 18 (FUNCD FN)))))
    (IF (LS BP BP0) (RETURN NIL))
    (IF (NULL A)
     (BLOCK NIL (SET A (CREATE (LADDR BP) (QUOTE OCTAL) 0)) (GO A)))
    (FOR I (STEP (LADDR BP) -1 EQ 0)
     (SET (A I) (CORE (PLUS BP I -1))))
    (FIXBPI (PLUS (S20. A) 1) BP 2Q5)
    (FOR I (STEP 1 1 GR (LADDR BP))
     (BLOCK NIL (PRINWORD (A I)) (SET LSIZE (PLUS LSIZE 1))))
    (IF (EQ (PLUS (S20. A) I) ARP) (SET ARP (S20. A))) (RETURN V)))
  (FUNCTION ((UNLOADL . LISP) SYMBOL)
   ((LNAME SYMBOL))
   (BLOCK ((LL SYMBOL (FINDN LNAME DUMPS)) (X SYMBOL))
    (IF (NULL LL)
     (ERROR (CONS LNAME (QUOTE (ILLEGAL LIBRARY FILE NAME)))))
    (FOR X (IN (CADDR LL)) (UNLDFN X LL))))
  (FUNCTION ((UNLOADFN . LISP) SYMBOL)
   ((NAME SYMBOL) (SEC SYMBOL))
   (BLOCK ((V SYMBOL (GETFN NAME SEC)) (LL SYMBOL))
    (IF (NULL V) (RETURN NIL))
    (FOR LL (IN DUMPS)
     (UNLESS (NOT (MEMBERN V (CADDR LL)))
     (BLOCK NIL (UNLDFN V LL) (GO R)))) R))
  (FUNCTION (UNLDFN SYMBOL)
   ((V SYMBOL) (LL SYMBOL))
   (IF (NOT FREEZE)
    (BLOCK ((FN (FUNCTIONAL NOVALUE) (VAR2FUNC V))
      (BP INTEGER) (S SYMBOL) (FREEZE BOOLEAN FREE TRUE))
     (SET S (OFFTRAC FN))
     (IF (OR (NOT (FREADY FN)) (FACTIVE FN))
      (BLOCK NIL (IF S (ONTRAC FN S)) (GO R)))
     (SET BP (PLUS (BIT 0 18 (FUNCD FN)) -1))
     (SET (BIT 42 6 (CORE BP)) 0Q)
     (SET (BPMIN . GC) (PLUS (BPMIN . GC) (MINUS (LADDR BP))))
     (SET (FUNCD FN)
      (WORDOR 2Q15 (WORDAND (FUNCD FN) 7Q14)
       (BIT 0 24 (F20. LOTRAP))))
     (SET (TRAPP FN) (CAR LL))
     (IF S (SET (TRAPP FN) (CONS S (CAR LL)))) R) NIL)))
(LOADL (SECTION SYS SYMBOL)
(FUNCTION ((LOADL . LISP) SYMBOL)
 ((LNAME SYMBOL))
 (BLOCK ((LL SYMBOL (FINDN LNAME DUMPS))
   (BP INTEGER)
   (I INTEGER)
   (J INTEGER)
   (M SYMBOL) (U SYMBOL) (FF (FUNCTIONAL SYMBOL SYMBOL)))
  (IF (NULL LL)
   (ERROR (CONS LNAME (QUOTE (NOT LIBRARY FILE NAME)))))
  (SET M (CADR LL))
  (IF (SET U (CDDDR M)) (BLOCK NIL (SET FF (CAR U)) (FF (CDR U))))
  (SET I (CADDR M))
  (IF (GR I (DIFFERENCE ARO BPP)) (UNLOADL LNAME))
  (SET BP (READLIB (CAR M) (CADR M) I))
  (SET J (PLUS BP I))
  (FOR BP (RESET BP (PLUS BP (LADDR BP)))
   (WHILE (LS BP J))
   (BLOCK ((FN (FUNCTIONAL NOVALUE)
      (O2F. (WORDOR 2Q7 (RADDR BP)))))
    (SET U (FUNC2VAR FN))
    (IF (OR (FREADY FN) (NOT (MEMBERN U (CADDR LL))))
     (BLOCK NIL (SET (BIT 42 6 (CORE BP)) 0Q)
      (SET (BPMIN . GC) (PLUS (BPMIN . GC) (MINUS (LADDR BP))))
      (GC R))) (SETFD U BP) (FIXBPI BP 2Q5 BP) R))
  (RETURN LNAME)))
(FUNCTION ((OPENL . LISP) SYMBOL)
 ((X SYMBOL))
 (BLOCK ((S SYMBOL (FINDN X LFILES)) (Y SYMBOL))
  (IF (GET X (FILES. . IO))
   (IF S (RETURN NIL)
    (ERROR (CONS X (QUOTE (NOT LIBRARY FILE))))))
  (IF S (SET Y (LIST (QUOTE OLD)))
   (SET LFILES (CONS (CONS X 0) LFILES)))
  (OPEN X (APPEND Y (APPEND (QUOTE ((FORM . BINARY)
      (RECORD . 512))) DISC.)))
  (SET (CDR (GET (QUOTE BUF) (GET X (FILES. . IO)))) NIL)
  (RETURN X)))
(FUNCTION (READLIB INTEGER)
 ((FILE SYMBOL) (SECT INTEGER) (I INTEGER))
 (BLOCK (((SHORWD . IO) (QUOTE (*STRING ANYSIX))))
  (BLOCK ((NAME INTEGER ((CVRTNM . IO) FILE)))
   (OPENL FILE)
   (BLOCK ((BP INTEGER (GETBPS I)))
    (BLOCK ((B INTEGER BP) (J INTEGER I) (K INTEGER) (S INTEGER))
     (FOR J (RESET J (PLUS J (MINUS K)))
      (WHILE (NQ J 0))
      (BLOCK NIL (IF (GR (INLIB B NAME SECT (SET K (MIN J (TIMES
            512 (SET S (PLUS 8 (MINUS (REMAINDER SECT 8))))))))) 3)
        (BLOCK NIL (SET (CORE BP) 0)
         (SET (LADDR BP) I)
         (SET (BPMIN . GC) (PLUS (BPMIN . GC) (MINUS I)))
         (ERROR (QUOTE (READLIB ERROR)))))
       (SET B (PLUS B K)) (SET SECT (PLUS SECT S)))))
     (RETURN BP))))))
(ROUTINE (INLIB INTEGER)
 ((BP INTEGER) (NAME INTEGER) (SECT INTEGER) (I INTEGER))
 (BLOCK NIL (SET (CORENTRY MNAME) NAME)
  (SET (CORENTRY MINOUT) (CORENTRY IN))
  (SET (CORENTRY MLOC) BP)
  (SET (CORENTRY MSECTR) SECT)
  (SET (CORENTRY MSIZE) I)
  (CODE (LDA (ENTRY MCALL) (RA R)) (BUC (ENTRY DSPCHR)))
  (RETURN (BIT 0 6 (CORENTRY MSTAT)))))
  (FUNCTION (REMFN NOVALUE)
   ((FD SYMBOL))
   (BLOCK ((X SYMBOL))
    (FOR X (IN DUMPS)
     (UNLESS (NOT (MEMBERN FD (CADDR X))))
     (BLOCK NIL (IF (SET (CADDR X) (DELE FD (CADDR X)))
       (SET (CADDDR X) (CONS FD (CADDDR X))) (REMOVEL (CAR X)))))))
  (FUNCTION ((REMOVEL . LISP) SYMBOL)
   ((LNAME SYMBOL))
   (BLOCK ((LL SYMBOL (FINDN LNAME DUMPS))
     (FREEZE BOOLEAN FREE TRUE))
    (IF (NULL LL) (RETURN NIL))
    (LOADL (CAR LL))
    (SET DUMPS (DELE LL DUMPS))
    (RETURN (MAPCAR (CADDR LL) VARNAME)))))
(START (SECTION SYS SYMBOL)
  (FUNCTION ((START . SYS) NOVALUE)
   NIL (BLOCK ((X SYMBOL))
    (IOINI)
    A (RESTART)
    (TRY X E (SUPVFN ITTY OTTY (QUOTE IL)))
    (GO A) E (TRY X A (PRBACK X)) (GO A)))
  (FUNCTION (SYSINI NOVALUE)
   NIL (BLOCK ((X (ARRAY OCTAL) (O2S. (ENTRY FIXBUF))))
    (SET (BUFIX . IO) X)
    (LOCSET (FIXLOC . IO) (X 0))
    (SET EOFCH (OCT2CH 34Q))
    (SET XXCHAR (OCT2CH 14Q1))
    (SET TRYVAR (GETFREE (QUOTE TRYPT) (QUOTE SYS)))
    (CODE (LDA (TRP . SYS))
     (ADD (PDBUF . GC))
     (STA (ENTRY PDOK1) S7.123)
     (ADD (NUMBER 25)) (STA ((ENTRY PDOK) 1) S7.123))))
  (FUNCTION (PRBACK NOVALUE)
   ((X SYMBOL))
   (BLOCK NIL (SET BACKTRACE BACTRC)
    (MESSAGE (APPEND (QUOTE (TOP LEVEL EXIT VALUE)) X))
    (MESSAGE (CONS (QUOTE BACKTRACE) (LASTN PRNMAX BACKTRACE)))))
  (FUNCTION (IOINI NOVALUE)
   NIL (BLOCK NIL (SET (FILES. . IO) NIL)
    (SET (ICURFN . IO) (SET (CURFN . IO) NIL))
    (SET (CORENTRY DNAME) (CORE (PLUS (ENTRY BELL) 2)))
    (SET (BIT 0 6 (CORENTRY DUNIT)) 1Q1)
    (SET (BIT 0 6 (CORENTRY DFORM)) 21Q)
    (SET (CORENTRY DSIZE) 1)
    (CODE (LDA (ENTRY DCALL) (L567.7 R)) (BUC (ENTRY DSPCHR)))
    (OPEN ITTY TTY.)
    (OPEN OTTY TTY.) (INPUT ITTY) (OUTPUT OTTY) (MESSAGE SIGNON)))
  (FUNCTION (RESTART NOVALUE)
   NIL (BLOCK NIL (SET (FSCHAR . FSM) NIL)
    ((MAKEST . FSM)) (SET (BACTRC . LISP) NIL)))
  (FUNCTION (RECOV NOVALUE) NIL (CODE (0)))
  (FUNCTION (LSUPV NOVALUE)
   ((INFILE SYMBOL) (OUTFILE SYMBOL) (FORM SYMBOL))
   (BLOCK ((A SYMBOL)
     (B SYMBOL)
     (G (FUNCTIONAL SYMBOL))
     (INF SYMBOL (INPUT INFILE)) (OUT SYMBOL (OUTPUT OUTFILE)))
    (ENDIN)
    A (SET A (READ))
    B (IF (NOT (ATOM A))
     (IF (EQ (CAR A) (QUOTE LAP))
      (GO LAP) (EQ INFILE ITTY) (GO PR) (GO A))
```

```
 (OR (EQN A EOFCH) (EQ A (QUOTE STOP)) (EQ A (QUOTE END)))           0082000
 (GO EXIT))                                                          0082100
AT (SET B (READ))                                                    0082200
(IF (EQ A (QUOTE DISC))                                              0082300
 (SET A (OPEN (CAR B) (APPEND (CDR B) DISC.)))                       0082400
 (EQ A (QUOTE TAPE))                                                 0082500
 (SET A (OPEN (CAR B) (APPEND (CDR B) TAPE.)))                       0082600
 (EQ A (QUOTE LSUPV))                                                0082700
 (LSUPV (CAR B) (CADR B) (QUOTE IL))                                 0082800
 (EQ A (QUOTE ID))                                                   0082900
 (SET A (S2O. B))                                                    0083000
 (EQ A (QUOTE TRACEARGS))                                            0083100
 (SET A (TRACEARGS B))                                               0083200
 (EQ A (QUOTE UNTRACE))                                              0083300
 (SET A (UNTRACE B))                                                 0083400
 (EQ A (QUOTE DUMPSEC))                                              0083500
 (SET A (DUMPSEC (CAR B) (CADR B) (CADDR B)))                        0083600
 (EQ A (QUOTE DUMPL))                                                0083700
 (SET A (DUMPL (CAR B) (CADR B) (CADDR B)))                          0083800
 (EQ A (QUOTE UNLOADL))                                              0083900
 (SET A (UNLOADL B))                                                 0084000
 (EQ A (QUOTE EVAL))                                                 0084100
 (SET A (EVAL B))                                                    0084200
 (EQ A (QUOTE FREE))                                                 0084300
 (BLOCK ((V SYMBOL (GETFREE (CAR B) (CDR B))))                       0084400
  (IF V (SET A (LIST A (S2O. V)                                      0084500
     (CORE (PLUS (S2O. V) -1)) (FVLIST V)))))                        0084600
 (EQ A (QUOTE LAPSTOP))                                              0084700
 (BLOCK NIL (IF (NOT B) (SET A (LAPGO))) (SET LAPSTOP B))            0084800
 (EQ A (QUOTE SUPV)) (SUPV) (BLOCK NIL (SET A B) (GO B)))            0084900
(GO PR)                                                              0085000
LAP (SET (ERRFLG . SUPV) FALSE)                                      0085100
(SET A (LAP (CADR A) (CADDR A) (CADDDR A)))                          0085200
(IF (ERRFLG . SUPV) (GO A) (NQ (CDR A) (QUOTE RUN)) (GO PR))         0085300
(SET B A)                                                            0085400
(SET G (VAR2FUNC (GETFREE (CAR B) (CDR B))))                         0085500
(SET A (G))                                                          0085600
(EXCISE (CAR B) (CDR B))                                             0085700
PR (PRETTYP A) (GO A) EXIT (INPUT INF) (OUTPUT OUT))))               0085800
                                                                     0085900
****END OF FILE DETECTED
```

```
(DUMMY (SECTION (LAP SYS) SYMBOL)
  (FUNCTION ((LAP . LISP) SYMBOL)
   ((A SYMBOL) (B SYMBOL) (C SYMBOL)))
  (FUNCTION (LAPP SYMBOL)
   ((LISTING SYMBOL) (DLIST SYMBOL) (SNAME SYMBOL FLUID)))
  (FUNCTION (LAPDECLARE NOVALUE) ((D SYMBOL)))
  (FUNCTION (FREEDECL SYMBOL) ((D SYMBOL)))
  (FUNCTION (FORMALIZE SYMBOL) ((V SYMBOL) (D SYMBOL)))
  (FUNCTION (VNAMER SYMBOL) ((N SYMBOL)))
  (FUNCTION (LAP1 NOVALUE) ((MODE SYMBOL) (PDK INTEGER)))
  (ROUTINE (LAPPUSH1 NOVALUE) ((N INTEGER)))
  (ROUTINE (LAPPOP1 NOVALUE) ((N INTEGER)))
  (ROUTINE (NUMADDR BOOLEAN) ((X SYMBOL)))
  (FUNCTION (ROUTP SYMBOL) ((V SYMBOL)))
  (FUNCTION (LAP2 NOVALUE) ((MODE SYMBOL) (PDK INTEGER)))
  (FUNCTION (FLBIND SYMBOL) ((L SYMBOL)))
  (FUNCTION (FLRESTS SYMBOL) NIL)
  (FUNCTION (LAPADDR NOVALUE) ((X SYMBOL) (P BOOLEAN)))
  (FUNCTION (LAPABSADDR INTEGER) ((X SYMBOL)))
  (FUNCTION (LAPADD1 BOOLEAN) ((X SYMBOL) (P BOOLEAN)))
  (FUNCTION (LAPTAG NOVALUE) ((X SYMBOL)))
  (FUNCTION (BYTMD SYMBOL) ((L SYMBOL)))
  (FUNCTION (APMOD SYMBOL) ((P OCTAL) (A SYMBOL)))
  (FUNCTION (LAPPUSH NOVALUE) ((N INTEGER) (B OCTAL)))
  (ROUTINE (LAPPOP NOVALUE) ((N INTEGER)))
  (FUNCTION (LAPCALL2 NOVALUE) NIL)
  (FUNCTION (BPINSTR NOVALUE) ((C OCTAL) (R INTEGER) (Y SYMBOL)))
  (FUNCTION (BPINST NOVALUE)
   ((C OCTAL) (L INTEGER) (X SYMBOL) (R INTEGER) (Y SYMBOL)))
  (ROUTINE (BPADDR OCTAL)
   ((C INTEGER) (M SYMBOL) (A INTEGER) (H SYMBOL)))
  (ROUTINE (BPLOC INTEGER) ((C INTEGER)))
  (FUNCTION (REMWORD NOVALUE) ((C OCTAL) (R INTEGER) (Y SYMBOL)))
  (FUNCTION (NCUP SYMBOL) ((N INTEGER) (X SYMBOL)))
  (FUNCTION (LAPNIX NOVALUE) ((M SYMBOL)))
  (ROUTINE (LAPID SYMBOL) ((X SYMBOL)))
  (ROUTINE (LAPFREE SYMBOL) ((N SYMBOL) (SN SYMBOL)))
  (FUNCTION ((LAPGC . SYS) SYMBOL) NIL)
  (FUNCTION (PRINCOM NOVALUE) ((M SYMBOL) (I INTEGER)))
  (FUNCTION (BLANKS NOVALUE) ((I INTEGER)))
  (FUNCTION (PRINOCT NOVALUE) ((C OCTAL) (I INTEGER))))
(LAPINIT (SECTION OPCODE SYMBOL)
  (DECLARE (PER OCTAL OWN 4Q13)
   (XEC OCTAL OWN 1Q14)
   (BUC OCTAL OWN 14Q13)
   (BUS OCTAL OWN 15Q13)
   (SFC OCTAL OWN 2Q14)
   (SFA OCTAL OWN 201Q12)
   (CYC OCTAL OWN 24Q13)
   (CYA OCTAL OWN 241Q12)
   (CYB OCTAL OWN 242Q12)
   (STF OCTAL OWN 5Q14)
   (STZ OCTAL OWN 51Q13)
   (ADD OCTAL OWN 1Q15)
   (ADM OCTAL OWN 104Q13)
   (SUB OCTAL OWN 11Q14)
   (SBM OCTAL OWN 114Q13)
   (MUL OCTAL OWN 12Q14)
   (DVD OCTAL OWN 134Q13)
   (LDA OCTAL OWN 2Q15)
   (LDM OCTAL OWN 204Q13)
   (LDC OCTAL OWN 21Q14)
   (LMC OCTAL OWN 214Q13)
```

```
    (LDB OCTAL OWN 22Q14)
    (LBC OCTAL OWN 224Q13)
    (LDL OCTAL OWN 23Q14)
    (LLC OCTAL OWN 234Q13)
    (FAD OCTAL OWN 3Q15)
    (FAM OCTAL OWN 304Q13)
    (FSB OCTAL OWN 31Q14)
    (FSM OCTAL OWN 314Q13)
    (FLT OCTAL OWN 32Q14)
    (FRN OCTAL OWN 324Q13)
    (FMP OCTAL OWN 33Q14)
    (FDV OCTAL OWN 334Q13)
    (CAS OCTAL OWN 4Q15)
    (INS OCTAL OWN 404Q13)
    (COM OCTAL OWN 41Q14)
    (TST OCTAL OWN 414Q13)
    (LDX OCTAL OWN 42Q14)
    (ATX OCTAL OWN 424Q13)
    (CON OCTAL OWN 43Q14)
    (ANA OCTAL OWN 430004Q10)
    (XOR OCTAL OWN 43003Q11)
    (CRA OCTAL OWN 430034Q10)
    (ANS OCTAL OWN 430204Q10)
    (STMZ OCTAL OWN 430274Q10)
    (LDS OCTAL OWN 434Q13)
    (STS OCTAL OWN 44Q14)
    (LDI OCTAL OWN 444Q13)
    (STA OCTAL OWN 5Q15)
    (STB OCTAL OWN 504Q13)
    (STL OCTAL OWN 51Q14)
    (STP OCTAL OWN 514Q13)
    (STX OCTAL OWN 52Q14)
    (ECH OCTAL OWN 524Q13)
    (AOR OCTAL OWN 53Q14)
    (SOR OCTAL OWN 534Q13)
    (ATR OCTAL OWN 54Q14)
    (BOZ OCTAL OWN 6002Q12)
    (BNZ OCTAL OWN 6012Q12)
    (BOZP OCTAL OWN 6Q15)
    (BNZP OCTAL OWN 601Q13)
    (BOZM OCTAL OWN 6001Q12)
    (BNZM OCTAL OWN 6011Q12)
    (BSN OCTAL OWN 604Q13)
    (BSG OCTAL OWN 61Q14)
    (BOP OCTAL OWN 61000177Q8)
    (BOM OCTAL OWN 61040177Q8)
    (BAR OCTAL OWN 614Q13)
    (BXH OCTAL OWN 7Q15)
    (BXL OCTAL OWN 71Q14)
    (BXE OCTAL OWN 72Q14)
    (BSX OCTAL OWN 73Q14)
    (BAX OCTAL OWN 74Q14)
    (BPX OCTAL OWN 75Q14) (BMX OCTAL OWN 76Q14))
   (SECTION MODCODE SYMBOL)
   (DECLARE (A OCTAL OWN 17Q)
    (AC OCTAL OWN 17Q)
    (A. OCTAL OWN 17Q)
    (I OCTAL OWN 2Q1)
    (D OCTAL OWN 4Q1)
    (L OCTAL OWN 4Q6)
    (R OCTAL OWN 1Q7)
    (S OCTAL OWN 4Q4)
    (T OCTAL OWN 4Q5) (N OCTAL OWN 1Q7) (RA OCTAL OWN 37Q3))
```

```
  (SECTION (LAP SYS) SYMBOL)
  (DECLARE (ENTRIES SYMBOL OWN))
  (DECLARE ((LAPSTOP . SYS) BOOLEAN FREE) (LAPSTL SYMBOL OWN))
  (DECLARE (SNAME SYMBOL FLUID)
   (FNAME SYMBOL FLUID)
   (FSEC SYMBOL FLUID)
   (ROUT BOOLEAN FLUID)
   (FDESC SYMBOL FLUID)
   (FSIZ INTEGER FLUID)
   (ORIGIN BOOLEAN FLUID)
   (ORGMODE BOOLEAN FLUID)
   (ILC INTEGER FLUID)
   (RLC INTEGER FLUID)
   (PDC INTEGER FLUID)
   (PDMAP SYMBOL FLUID)
   (MAPS SYMBOL FLUID)
   (PDMIN INTEGER FLUID)
   (PDMAX INTEGER FLUID)
   (LABELS SYMBOL FLUID)
   (ALIST SYMBOL FLUID)
   (APLIST SYMBOL FLUID)
   (FL SYMBOL FLUID)
   (FA INTEGER FLUID)
   (FM SYMBOL FLUID)
   (FR INTEGER FLUID)
   (TAG INTEGER FLUID)
   (LG SYMBOL FLUID)
   (IT SYMBOL FLUID) (ERRS BOOLEAN FLUID) (OUTLAP SYMBOL OWN)))
(MACROS (SECTION SYS SYMBOL)
  MACRO1 (((FLVAL (LAMBDA (X)
      (LIST (QUOTE CORE)
       (LIST (QUOTE PLUS)
        -1 (APPEND (QUOTE (CHEAT SYMBOL INTEGER)) (CDR X))))))))
  (ROUTINE (SETFD NOVALUE) ((FD SYMBOL) (J INTEGER)))
  (FUNCTION (MESSAGE SYMBOL) ((M SYMBOL)))
  (FUNCTION (EXCISF SYMBOL) ((V SYMBOL)))
  (FUNCTION (FTRANS SYMBOL) ((P SYMBOL)))
  (FUNCTION (GETBPS INTEGER) ((I INTEGER)))
  (ROUTINE (VREFCT NOVALUE) ((V SYMBOL)(I INTEGER)))
  (DECLARE ((ERRFLG . SUPV) BOOLEAN FREE)))
(LAP (SECTION (LAP SYS) SYMBOL)
  (FUNCTION ((LAP . LISP) SYMBOL)
   ((A SYMBOL) (B SYMBOL) (C SYMBOL))
   (BLOCK ((APLIST SYMBOL FREE NIL)) (RETURN (LAPP A B C))))
  (FUNCTION (LAPP SYMBOL)
   ((LISTING SYMBOL) (DLIST SYMBOL) (SNAME SYMBOL FLUID))
   (BLOCK ((FNAME SYMBOL FLUID)
     (FSEC SYMBOL FLUID)
     (ROUT BOOLEAN FLUID)
     (FDESC SYMBOL FLUID)
     (FSIZ INTEGER FLUID)
     (ORIGIN BOOLEAN FLUID)
     (ORGMODE BOOLEAN FLUID)
     (ILC INTEGER FLUID)
     (RLC INTEGER FLUID)
     (PDC INTEGER FLUID)
     (PDMAP SYMBOL FLUID)
     (MAPS SYMBOL FLUID)
     (PDMIN INTEGER FLUID)
     (PDMAX INTEGER FLUID)
     (LABELS SYMBOL FLUID)
     (ALIST SYMBOL FLUID)
     (FL SYMBOL FLUID)
```

```
 (HA INTEGER FLUID)
 (HM SYMBOL FLUID)
 (HR INTEGER FLUID)
 (TAG INTEGER FLUID)
 (LG SYMBOL FLUID)
 (IT SYMBOL FLUID)
 (ERRS BOOLEAN FLUID)
 (FD SYMBOL)
 (A SYMBOL) (D SYMBOL) (I INTEGER) (J INTEGER) (PATCH BOOLEAN))
(SET DLIST (MAPCAR DLIST FREEDECL))
(IF (EQ (CAR LISTING) (QUOTE PATCH))
 (GO B) (EQ (CAR LISTING) (QUOTE ROUTINE)) (SET ROUT TRUE))
(SET FNAME (VNAMER (CAADR LISTING)))
(SET FSEC (CDR FNAME))
(SET D (CADDR LISTING))
(SET A (LIST (CAR LISTING)
  (FORMALIZE (CADADR LISTING) D) (QUOTE VALUE)))
(SET FD (FREEDECL (CONS FNAME A)))
(SET FDESC (FREEDECL (CONS (CONS (GENID) (QUOTE TEMP)) A)))
(SET FNAME (CAR FNAME))
(IF (NULL FD) (GO ERRS))
(SET FSIZ (PLUS FPP (SET ILC 1)))
(SET LG (CDDDR LISTING))
(GO LAP1)
B (SET FNAME (QUOTE PATCH))
(SET ILC (SET FSIZ FPP))
(SET ORIGIN (SET ORGMODE (SET PATCH TRUE)))
(SET LG (CDR LISTING))
LAP1 (SET PDMAX 1)
(SET RLC (SET PDC (SET PDMIN 0)))
(IF D (SET PDC 1))
(LAP1 (QUOTE END) 0)
J (LAP1 NIL 0)
(IF LG (GO J) ERRS (GO ERRS) (NOT PATCH) (VREFCT FDESC 1))
(IF (NOT ORIGIN) (GO C) (NOT ORGMODE) (SET ILC FSIZ))
(SET FSIZ (PLUS ILC RLC (MINUS FPP)))
(SET ORGMODE TRUE)
(SET RLC ILC)
(SET J (SET ILC FPP))
(IF PATCH (GO E) (GO D))
C (SET FSIZ (PLUS ILC RLC))
(SET I (IQUOTIENT (PLUS FSIZ 23) 24))
(SET FSIZ (PLUS FSIZ I))
(SET J (GETBPS FSIZ))
(FOR I (STEP I -1 EQ 0) (SET (CORE (PLUS J FSIZ (MINUS I))) 0Q))
(SET RLC ILC)
(SET ILC 0)
D (SETFD FDESC J)
(BPINST 4000000001Q6 FSIZ (QUOTE A) (S20. FDESC) (QUOTE F))
(PRINCOM NIL 0)
(SET LG (CDDDR LISTING))
(SET MAPS (LIST (CONS 0 (PLUS ILC -1)) (CONS -1 (PLUS ILC -1))))
(GO LAP2)
E (SET LG (CDR LISTING))
(SET MAPS (QUOTE ((0 . 4002Q1) (-1 . 0))))
LAP2 (SET PDC (MINUS (LENGTH D)))
(SET ALIST (SET FL NIL))
(LAPDECLARE D)
(IF PDMAP (SET (CDR PDMAP) NIL))
(SET PDMIN 0)
(LAP2 (QUOTE END) 0)
K (LAP2 NIL 0)
(IF LG (GO K))
```

```
    (IF PATCH (GC R)                                                    C025300
     ERRS (BLOCK NIL (EXCISF FDESC) (GC ERRS))                          C025400
     (AND LAPSTCP (NQ FSEC (QUOTE RUN)))                                C025500
     (SET LAPSTL (CONS (CONS FD FDESC) LAPSTL))                         C025600
     (FTRANS (CCNS FD FDESC)))                                          C025700
    R (RETURN (CCNS FNAME FSEC))                                        C025800
    ERRS (SET (ERRFLG . SUPV) TRUE) (GC R))))                           C025900
(LAPDECLARE (FUNCTION (LAPDECLARE NCVALUE)                              0026000
   ((D SYMBOL))                                                         C026100
   (BLOCK ((V SYMBCL) (DT SYMBOL) (DF SYMBOL) (DL SYMBCL))              C026200
    A (IF (NULL D) (GC EXIT))                                           C026300
    (SET V (CAR D))                                                     C026400
    (SET D (CDR D))                                                     C026500
    (SET DT (CADR V))                                                   0026600
    (SET DF (OR (MEMBER (QUOTE FREE) (CDDR V))                          C026700
      (MEMBER (QUCTE FLUID) (CDDR V))))                                 C026800
    (SET DL (MEMBER (QUCTE LOC) (CDDR V)))                              0026900
    (SET V (CAR V))                                                     0027000
    (IF (AND (ATCM V) (NCT DF)) (GC C))                                 C027100
    (SET V (VNAMER V))                                                  C027200
    (MAKEFREE (CAR V)                                                   C027300
     (CDR V) (QUCTE STET) DT (IF DL (QUOTE LOC) (QUCTE VALUE)))         C027400
    (SET DF (IF DL (QUCTE (LOC)) (QUCTE (FLUID))))                      C027500
    C (LAPPUSH 1 (IF (OR DL (EQ DT (QUCTE SYMBOL))                      C027600
       (NCT (ATOM DT))) 1Q 0Q))                                         C027700
    (IF (EQ PDC 0) (SET PDC 1))                                         C027800
    (SET ALIST (CONS (CCNS V PDC DF) ALIST)) (GO A) EXIT))              C027900
  (FUNCTICN (FREEDECL SYMBCL)                                           0028000
   ((D SYMBOL))                                                         C028100
   (BLOCK ((N SYMBCL))                                                  C028200
    (SET N (VNAMER (CAR D)))                                            C028300
    (RETURN (MAKEFREE (CAR N)                                           C028400
      (CDR N)                                                           C028500
      (IF (OR (EQ (CADR D) (QUOTE FREE))                                C028600
        (EQ (CADR D) (QUCTE FLUID))) (QUOTE STET) (CADR D))             0028700
      (CADDR D)                                                         C028800
      (IF (MEMBER (QUOTE LCC) (CDDDR D))                                0028900
       (QUCTE LCC) (QUCTE VALUE))))))                                   C029000
  (FUNCTICN (FORMALIZE SYMBOL)                                          C029100
   ((V SYMBOL) (D SYMBCL))                                              0029200
   (CONS (QUOTE FUNCTICNAL)                                             C029300
    V (MAPCAR D (FUNCTICN ((G02446 . G02447) SYMBOL)                    0029400
      ((J SYMBCL))                                                      C029500
      (IF (MEMBER (QUOTE LCC) (CDDR J))                                 0029600
       (LIST (FTYPER (CADR J)) (QUCTE LOC)) (FTYPER (CADR J)))))))      0029700
  (FUNCTICN (VNAMER SYMBCL)                                             C029800
   ((N SYMBOL)) (IF (ATCM N) (CONS N SNAME) N)))                        C029900
(LAP1 (FUNCTION (LAP1 NCVALUE)                                          C030000
   ((MODE SYMBCL) (PDK INTEGER))                                        C030100
   (BLOCK ((CP SYMBOL) (X SYMBOL) (U SYMBOL))                           C030200
    A (IF (NULL LG)                                                     0030300
     (GO E)                                                             C030400
     (AND (NCT (ATCM (CAR LG))) (EQ (CAAR LG) (QUOTE DITTO)))           C030500
     (GC DITTO))                                                        C030600
    (SET IT (CAR LG))                                                   C030700
    (SET LG (CDR LG))                                                   C030800
    V (IF (NCT (ATCM IT))                                               C030900
     (GC B)                                                             C031000
     (OR (IDP IT) (FIXP IT)) (GO LABEL) IT (LAPNIX (QUCTE ITEM)))       C031100
    (GO A)                                                              C031200
    LABEL (SET U (FIND IT LABELS))                                      C031300
    (IF U (LAPNIX (QUOTE (MULTIPLE LABEL)))                             C031400
     (SET LABELS (CONS (CONS IT ILC) LABELS)))                          C031500
```

```
(GO A)
B (SET OP (CAR IT))
(SET X (CDR IT))
(SET U (FINDN OP (QUOTE ((ORG . 0)
    (ENTRY . 0)
    (BEGIN . 2)
    (RETURN . 1)
    (BLOCK . 0)
    (DECLARE . 0)
    (END . 0)
    (ARGS . 0)
    (CALL . 2)
    (CALL1 . 1)
    (CALL2 . 1)
    (FLBIND . 1)
    (PUSHA. . 0) (PUSHP. . 0) (POP. . 0) (COMMENT . 0))))))
(IF (NULL U)
 (GO C)
 (AND (EQ OP (QUOTE BEGIN)) ROUT) (SET U (QUOTE (BEGIN . 1))))
(SET ILC (PLUS ILC (CDR U)))
(IF (EQ OP (QUOTE ORG))
 (GO ORG)
 (EQ OP (QUOTE BLOCK))
 (GO BLOCK)
 (EQ OP (QUOTE DECLARE))
 (GO DECLARE)
 (EQ OP (QUOTE END))
 (GO END)
 (EQ OP (QUOTE ARGS))
 (GO ARGS)
 (EQ OP (QUOTE CALL))
 (GO CALL)
 (EQ OP (QUOTE CALL2))
 (GO CALL2)
 (EQ OP (QUOTE CALL1))
 (GO CALL1)
 (EQ OP (QUOTE FLBIND))
 (GO FLBIND)
 (EQ OP (QUOTE PUSHA.))
 (GO PUSHA.)
 (EQ OP (QUOTE PUSHP.))
 (GO PUSHP.) (EQ OP (QUOTE POP.)) (GO POP.))
(GO A)
C (IF (NULL X)
 (GO D)
 (OR (EQ (CAR X) (QUOTE PUSHP.)) (EQ (CAR X) (QUOTE PUSHA.)))
 (LAPPUSH1 1)
 (EQ (CAR X) (QUOTE POP.))
 (LAPPOP1 1) (NUMADDR (CAR X)) (SET RLC (PLUS RLC 1)))
(IF (AND (CDR X) (CDDR X) (NUMADDR (CADDR X)))
 (SET RLC (PLUS RLC 1)))
D (SET ILC (PLUS ILC 1))
(GO A)
DITTO (IF (EQ (CADAR LG) 0) (GO W))
(SET LG (CONS (LIST (QUOTE DITTO) (PLUS (CADAR LG) -1))
  (CDR LG)))
(GO V)
W (SET LG (CDR LG))
(GO A)
ORG (SET ORIGIN TRUE)
(IF ORGMODE (SET FSIZ ILC))
(SET ORGMODE (OR (NULL X) (NULL (CAR X))))
(SET ILC (IF ORGMODE FSIZ (LAPABSADDR (CAR X))))
```

```
    (GO A)
    BLOCK (BLOCK ((FL SYMBOL FLUID)) (LAP1 (QUOTE DECLARE) PDC))
    (GO A)
    DECLARE (LAPPOP1 (DIFFERENCE PDC PDK))
    (LAPPUSH1 (LENGTH X))
    (IF (EQ MODE (QUOTE DECLARE))
     (SET MODE (QUOTE END)) (LAPNIX (QUOTE (DECLARE MISPLACED))))
    (GO A)
    END (IF (NOT (EQ MODE (QUOTE END)))
     (LAPNIX (QUOTE (END MISPLACED))))
    R (SET ILC (PLUS ILC (LENGTH FL)))
    (LAPPOP1 (DIFFERENCE PDC PDK))
    (GO EXIT)
    ARGS (BLOCK ((FL SYMBOL FLUID)) (LAP1 (QUOTE CALL) PDC))
    (GO A)
    CALL CALL2 (IF (ROUTP (CAR X))
     (SET ILC (PLUS ILC -1))
     (BLOCK NIL (IF ROUT (LAPNIX (QUOTE (FUNCTION CALL))))
      (SET RLC (PLUS RLC (TIMES (MAX 0 (IQUOTIENT (PLUS PDC (MINUS
            PDMIN) -1) 24)) 2))) (SET PDMIN PDC)))
    CALL1 (IF (EQ MODE (QUOTE CALL)) (GO R))
    (LAPNIX (QUOTE (MISPLACED CALL)))
    (GO A)
    FLBIND (IF (NULL X) (GO A))
    (SET U (LENGTH X))
    (SET ILC (PLUS ILC U))
    (LAPPUSH1 U)
    (SET FL (CONS NIL FL))
    (GO A)
    PUSHA. PUSHP. (LAPPUSH1 (CAR X))
    (GO A)
    POP. (LAPPOP1 (CAR X))
    (GO A)
    E (SET IT NIL)
    (IF (NULL MODE) (GO EXIT))
    (LAPNIX (QUOTE (END MISSING))) (GO R) EXIT))
  (ROUTINE (LAPPUSH1 NOVALUE)
   ((N INTEGER))
   (BLOCK NIL (SET PDC (PLUS PDC N)) (SET PDMAX (MAX PDMAX PDC))))
  (ROUTINE (LAPPOP1 NOVALUE)
   ((N INTEGER))
   (BLOCK NIL (SET PDC (DIFFERENCE PDC N))
    (SET PDMIN (MIN PDMIN PDC))))
  (ROUTINE (NUMADDR BOOLEAN)
   ((X SYMBOL)) (AND (NOT (ATOM X)) (EQ (CAR X) (QUOTE NUMBER))))
  (FUNCTION (ROUTP SYMBOL)
   ((V SYMBOL))
   (BLOCK NIL (SET V (VNAMER V))
    (IF (SET V (GETFREE (CAR V) (CDR V)))
     (RETURN (EQ (FVKIND V) (QUOTE ROUTINE))))))
(LAP2 (FUNCTION (LAP2 NOVALUE)
   ((MODE SYMBOL) (PDK INTEGER))
   (BLOCK ((CT SYMBOL)
     (OP SYMBOL)
     (OPV OCTAL)
     (RA INTEGER)
     (RM SYMBOL) (X SYMBOL) (U SYMBOL) (V SYMBOL) (L SYMBOL))
    A (IF L (GO K) (NULL LG) (GO E))
    (SET X (SET CT (CAR LG)))
    (SET LG (CDR LG))
    (GO M)
    K (SET CT NIL)
    L (SET X (CAR L))
```

```
(SET L (CDR L))
M (IF (NOT (ATOM X)) (GO Z) X (PRINCOM OT 26))
(GO A)
Z (IF (EQ (CAR X) (QUOTE DITTO)) (GO DITTO))
(SET IT X)
B (SET OP (CAR IT))
(SET X (CDR IT))
(IF (EQ OP (QUOTE ORG))
 (GO ORG)
 (EQ OP (QUOTE ENTRY))
 (GO ENTRY)
 (EQ OP (QUOTE BEGIN))
 (GO BEGIN)
 (EQ OP (QUOTE RETURN))
 (GO RETURN)
 (EQ OP (QUOTE BLOCK))
 (GO BLOCK)
 (EQ OP (QUOTE DECLARE))
 (GO DECLARE)
 (EQ OP (QUOTE END))
 (GO END)
 (EQ OP (QUOTE ARGS))
 (GO ARGS)
 (EQ OP (QUOTE CALL))
 (GO CALL)
 (EQ OP (QUOTE CALL1))
 (GO CALL1)
 (EQ OP (QUOTE CALL2))
 (GO CALL2)
 (EQ OP (QUOTE FLBIND))
 (GO FLBIND)
 (EQ OP (QUOTE PUSHA.))
 (GO PUSHA.)
 (EQ OP (QUOTE PUSHP.))
 (GO PUSHP.)
 (EQ OP (QUOTE POP.))
 (GO POP.) (EQ OP (QUOTE COMMENT)) (GO COMMENT))
C (SET OPV (IF (NUMBP OP)
  OP (SET U (GETFREE OP (QUOTE OPCODE)))
   (FLVAL U) (BLOCK NIL (LAPNIX (QUOTE OPCODE)) (RETURN 0Q))))
(SET HM (SET RM NIL))
(SET TAG 0)
(IF (NULL X) (GO D))
(LAPADDR (CAR X) TRUE)
(SET RA HA)
(SET RM HM)
(SET HM NIL)
(SET X (CDR X))
(IF (NULL X) (GO D))
(LAPTAG (CAR X))
(SET X (CDR X))
(IF (NULL X) (GO D))
(LAPADDR (CAR X) NIL)
D (SET OPV (WORDOR OPV (SHIFT TAG 18)))
(BPINST OPV HA HM RA RM)
(GO P)
DITTO (IF (EQ (CADR X) 0) (GO Q))
(SET L (CONS IT (LIST (QUOTE DITTO) (PLUS (CADR X) -1)) L))
(GO L)
ORG (SET ORGMODE (OR (NULL X) (NULL (CAR X))))
(SET ILC (IF ORGMODE FPP (BLOCK NIL (LAPABSADDR (CAR X))
   (RETURN HA))))
(GO Q)
```

```
ENTRY (LAPABSADDR (CADR X))                                             0050500
(IF (FINDN (CAR X) ENTRIES) (LAPNIX (QUOTE (ENTRY REDEFINED))))         0050600
(SET ENTRIES (CONS (CONS (CAR X) (DRIVE OCTAL HA)) ENTRIES))            0050700
(SET CT (APPEND CT (LIST (QUOTE '=) (DRIVE OCTAL HA))))                 0050800
(GO Q)                                                                  0050900
BEGIN (SET IT (QUOTE (STP 0 8)))                                        0051000
(IF ROUT (GO B))                                                        0051100
(SET L (CONS (IF (LS PDMAX 25)                                          0051200
   (QUOTE (XEC (ENTRY PDOK1)))                                          0051300
   (APPEND (QUOTE (BPX (ENTRY PDOK) 8)) (LIST PDMAX))) L))              0051400
(GO B)                                                                  0051500
RETURN (SET IT (IF ROUT (QUOTE (BUC (ENTRY ROUT)))                      0051600
  (QUOTE (BUC (ENTRY RETURN)))))                                        0051700
(GO B)                                                                  0051800
BLOCK (PRINCOM CT 31)                                                   0051900
(BLOCK ((ALIST SYMBOL FLUID ALIST) (FL SYMBOL FLUID NIL))               0052000
 (LAP2 (QUOTE DECLARE) PDC))                                            0052100
(GO A)                                                                  0052200
DECLARE (LAPPOP (DIFFERENCE PDC PDK))                                   0052300
(LAPDECLARE X)                                                          0052400
(IF (EQ MODE (QUOTE DECLARE))                                           0052500
 (SET MODE (QUOTE END)) (LAPNIX (QUOTE (DECLARE MISPLACED))))           0052600
(PRINCOM CT 3)                                                          0052700
(GO A)                                                                  0052800
END (IF FL (GO FLRESTS)                                                 0052900
 (NOT (EQ MODE (QUOTE END))) (LAPNIX (QUOTE (END MISPLACED))))          0053000
(IF CT (PRINCOM CT 31))                                                 0053100
(LAPPOP (DIFFERENCE PDC PDK))                                           0053200
(GO EXIT)                                                               0053300
FLRESTS (SET L (APPEND (FLRESTS) (CONS IT L)))                          0053400
(GO L)                                                                  0053500
ARGS (PRINCOM CT 31)                                                    0053600
(BLOCK ((FL SYMBOL FLUID)) (LAP2 (QUOTE CALL) PDC))                     0053700
(GO A)                                                                  0053800
CALL (IF (ROUTP (CAR X))                                                0053900
 (BLOCK NIL (SET IT (CONS (QUOTE CALL1) X)) (GO B)))                    0054000
(SET L (CONS (QUOTE (CALL2)) L))                                        0054100
(GO G)                                                                  0054200
CALL1 CALL2 (IF (NOT (EQ MODE (QUOTE CALL)))                            0054300
 (BLOCK NIL (IF (NULL L) (SET L (QUOTE (NIL))))                         0054400
  (LAPNIX (QUOTE (MISPLACED CALL))))                                    0054500
 (BLOCK NIL (SET MODE (QUOTE END))                                      0054600
  (SET L (CONS (QUOTE (END)) L))))                                      0054700
(IF (EQ OP (QUOTE CALL1)) (GO G))                                       0054800
(LAPCALL2)                                                              0054900
(PRINCOM NIL 0)                                                         0055000
(GO L)                                                                  0055100
G (SET IT (SUBST (CAR X) (QUOTE W) (QUOTE (BPX W (8 I) PDS.))))         0055200
(GO B)                                                                  0055300
FLBIND (SET L (APPEND (IF X (FLBIND X) (QUOTE (NIL))) L))               0055400
(GO L)                                                                  0055500
PUSHA. (LAPPUSH (CAR X) 0Q)                                             0055600
(GO P)                                                                  0055700
PUSHP. (LAPPUSH (CAR X) 1Q)                                             0055800
(GO P)                                                                  0055900
POP. (LAPPOP (CAR X))                                                   0056000
(GO P)                                                                  0056100
COMMENT (GO P)                                                          0056200
P (PRINCOM CT 7)                                                        0056300
(GO A)                                                                  0056400
Q (IF CT (PRINCOM CT 31))                                               0056500
(GO A)                                                                  0056600
E (SET IT NIL)                                                          0056700
```

```
    (IF (NULL MODE) (GO EXIT))
    (LAPNIX (QUOTE (END MISSING)))
    (SET L (CONS (QUOTE (END)) L)) (GO K) EXIT)))
(FLBIND (FUNCTION (FLBIND SYMBOL)
   ((L SYMBOL))
   (BLOCK (V M U (OP OCTAL 2Q14) (N INTEGER (LENGTH L)))
    (SET FL (CONS (PLUS ILC 1) FL))
    A (SET V (VNAMER (CAR L)))
    (IF (NULL (SET U (FIND V ALIST))) (GO E))
    (SET M (CONS (IF U (LIST (IF (EQ (CADDR U) (QUOTE LOC))
         OP (PLUS OP 1Q14)) V 0 (MINUS (CADR U)))
       (BLOCK NIL (LAPNIX (LIST V)) (RETURN (QUOTE (0))))) M))
    B (SET OP 0Q)
    (IF (SET L (CDR L)) (GO A))
    (RETURN (CONS (QUOTE (BSX (ENTRY FLBIND) 3 TOP.))
      (LIST (QUOTE PUSHP.) N) M))
    E (LAPNIX V) (SET M (CONS (QUOTE (0)) M)) (GO B)))
  (FUNCTION (FLRESTS SYMBOL)
   NIL (BLOCK ((U SYMBOL))
    (SET U (MAPCAR FL (FUNCTION ((G02448 . G02449) SYMBOL)
       ((J SYMBOL))
       (SUBST J (QUOTE W)
        (QUOTE (BSX (ENTRY FLREST) 7 (ORG. W)))))))
    (SET FL NIL) (RETURN U))))
(LAPADDR (FUNCTION (LAPADDR NOVALUE)
   ((X SYMBOL) (P BOOLEAN))
   (BLOCK ((HR INTEGER) (U SYMBOL))
    (SET HA (SET HR 0))
    (SET HM NIL)
    X (IF (NULL X)
     (GO EXIT)
     (LAPADD1 X P)
     (GO A)
     (IDP X)
     (GO B)
     (ATOM X)
     (BLOCK NIL (LAPNIX (LIST (QUOTE FIELD) X)) (GO EXIT))
     (IDP (CDR X))
     (GO V2)
     (EQ (CAR X) (QUOTE NUMBER))
     (GO NUMBER)
     (EQ (CAR X) (QUOTE QUOTE))
     (GO QUOTE)
     (EQ (CAR X) (QUOTE ID))
     (GO ID) (EQ (CAR X) (QUOTE LAP)) (GO LAP))
    SUM (IF (NOT (LAPADD1 (CAR X) P))
     (LAPNIX (LIST (QUOTE SUMMAND) (CAR X))))
    (IF (SET X (CDR X)) (GO SUM))
    A (IF (OR ORIGIN (EQ HR 0))
     (SET HM (QUOTE A))
     (EQ HR 1) (SET HM (QUOTE R)) (LAPNIX (QUOTE RELOCATION)))
    (GO EXIT)
    B (IF (EQ X (QUOTE PUSHA.))
     (LAPPUSH 1 0Q)
     (EQ X (QUOTE PUSHP.))
     (LAPPUSH 1 1Q) (EQ X (QUOTE POP.)) (LAPPOP 1) (GO V1))
    (IF (NOT P)
     (BLOCK NIL (LAPNIX (LIST (QUOTE DECREMENT) X)) (GO EXIT)))
    (SET HA (MINUS (IF (EQ X (QUOTE POP.)) (PLUS PDC 1) PDC)))
    (SET HM (QUOTE A))
    (SET TAG 8)
    (GO EXIT)
    V1 (IF (NULL (SET U (FIND X ALIST))) (GO V6) P (SET TAG 8))
  V5 (SET HA (MINUS (CADR U)))
  (SET HM (QUOTE A))
  (GO EXIT)
  V6 (IF (NULL (SET U (FIND X APLIST))) (GO V3) P (SET TAG 7))
  (GO V5)
  V2 V3 (SET X (VNAMER X))
  (SET U (CDR X))
  (SET X (CAR X))
  V4 (IF (NULL (SET U (LAPFREE X U)))
   (BLOCK NIL (LAPNIX (QUOTE VARIABLE)) (GO EXIT)))
  (SET HA (S2C. U))
  (SET HM (QUOTE F))
  (GO EXIT)
  NUMBER (SET HA RLC)
  (SET HM (QUOTE R))
  (REMWORD (IF (FIXP (CADR X))
    (I2C. (CADR X)) (R2C. (CADR X))) 0 NIL)
  (GO EXIT)
  QUOTE (SET HA (S2C. (MAKEQUOTE (CADR X))))
  (SET HM (QUOTE F))
  (GO EXIT)
  ID (SET HA (S2C. (LAPID (CADR X))))
  (SET HM (QUOTE S))
  (GO EXIT)
  LAP (SET X (BLOCK ((APLIST SYMBOL FREE ALIST))
    (RETURN (LAPP (CADR X) (CADDR X) (CADDDR X))))) (GO X) EXIT))
(FUNCTION (LAPABSADDR INTEGER)
 ((X SYMBOL))
 (BLOCK NIL (LAPADDR X NIL)
  (IF (OR (EQ HM (QUOTE R)) (EQ HM (QUOTE F)) (EQ HM (QUOTE S)))
   (LAPNIX (QUOTE (ABS ADDRESS)))) (RETURN HA)))
(FUNCTION (LAPADD1 BOOLEAN)
 ((X SYMBOL) (P BOOLEAN))
 (BLOCK ((V INTEGER) (R INTEGER) (S BOOLEAN))
  A (IF (FIXP X)
   (SET V X)
   (EQ X (QUOTE A.))
   (SET V 777621Q)
   (EQ X (QUOTE Z.))
   (SET V 7776Q2)
   (EQ X (QUOTE B.))
   (SET V 777622Q)
   (EQ X (QUOTE L.))
   (SET V 777745Q)
   (EQ X (QUOTE PDS.))
   (SET V (PLUS PDC 1))
   (EQ X (QUOTE TOP.))
   (BLOCK NIL (IF P (SET TAG 8)) (SET V (MINUS PDC)))
   (EQ X (QUOTE C.))
   (BLOCK NIL (SET R 1) (SET V ILC))
   (EQ X (QUOTE ORG.))
   (SET R 1)
   (ATOM X)
   (RETURN NIL)
   (EQ (CAR X) (QUOTE LABEL))
   (BLOCK ((U SYMBOL))
    (SET R 1)
    (SET U (FIND (CADR X) LABELS))
    (IF U (SET V (CDR U)) (LAPNIX X)))
   (EQ (CAR X) (QUOTE ENTRY))
   (BLOCK ((U SYMBOL))
    (SET U (FIND (CADR X) ENTRIES))
    (IF U (SET V (CDR U)) (LAPNIX X)))
```

```
     (EQ (CAR X) (QUOTE MINUS)) (GC M) (RETURN NIL))
    (IF S (BLOCK NIL (SET V (MINUS V)) (SET R (MINUS R))))
    (SET HA (PLUS HA V))
    (SET HR (PLUS HR R))
    (RETURN TRUE) M (SET S (NOT S)) (SET X (CADR X)) (GO A)))
  (FUNCTION (LAPTAG NOVALUE)
   ((X SYMBOL))
   (BLOCK ((U SYMBCL) (V SYMBOL))
    (IF (NULL X) (RETURN NIL) (ATOM X) (GO A))
    (SET U (CDR X))
    (SET X (CAR X))
    A (SET TAG (PLUS TAG (IF (FIXP X)
       X (SET V (GETFREE X (QUOTE MCDCCDE)))
       (FLVAL V)
       (SET V (BYTMD (EXPLODE X)))
       V (BLOCK NIL (LAPNIX (QUOTE TAG)) (RETURN 0Q)))))
    (IF (NULL U) (GO EXIT))
    (SET X (CAR U)) (SET U (CDR U)) (GO A) EXIT))
  (FUNCTION (BYTMD SYMBCL)
   ((L SYMBOL))
   (BLOCK ((X SYMBCL)
     (C SYMBCL) (LF SYMBCL) (RT SYMBCL) (I INTEGER))
    (IF (OR (EQ (SET C (CAR L)) (QUOTE 'L)) (EQ C (QUOTE 'S)))
     (SET L (CDR L)) (RETURN NIL))
    A (IF (NULL L)
     (GO COMP)
     (SET X (FIND (CAR L)
       (QUOTE (('0 . 0Q)
         ('1 . 1Q)
         ('2 . 2Q)
         ('3 . 3Q) ('4 . 4Q) ('5 . 5Q) ('6 . 6Q) ('7 . 7Q)))))
     (SET RT (CONS (CDR X) RT))
     (NOT (EQ (CAR L) (QUOTE '.)))
     (RETURN NIL) (BLOCK NIL (SET LF RT) (SET RT NIL)))
    (SET L (CDR L))
    (GO A)
    COMP (IF (NOT RT)
     (RETURN NIL)
     (NOT LF)
     (RETURN (APMOD 0Q RT)) (NOT (EQ C (QUOTE 'S))) (GO C2))
    (SET C LF)
    (SET LF RT)
    (SET RT C)
    C2 (IF (LS (SET I (DIFFERENCE (CAR LF) (CAR RT))) 0)
     (SET I (PLUS 8 I)))
    (IF (GQ (LENGTH LF) (LENGTH RT)) (RETURN (APMOD I LF)))
    (SET LF NIL)
    B (IF (NULL RT) (RETURN (APMOD I LF)))
    (SET LF (CONS (WORDAND 7Q (PLUS I (CAR RT))) LF))
    (SET RT (CDR RT)) (GO B)))
  (FUNCTION (APMOD SYMBCL)
   ((P OCTAL) (A SYMBOL))
   (BLOCK ((X OCTAL))
    (SET X (WORDOR (SHIFT P 18) 377Q2))
    A (IF (NULL A) (RETURN X))
    (SET X (WORDXOR X (SHIFT 2Q4 (MINUS (CAR A)))))
    (SET A (CDR A)) (GO A))))
(PCMAP (FUNCTION (LAPPUSH NOVALUE)
   ((N INTEGER) (B OCTAL))
   (BLOCK NIL (SET PDC (PLUS PDC N))
    (SET PCMAP (NCONC (NDUP N B) PCMAP))))
  (ROUTINE (LAPPOP NOVALUE)
   ((N INTEGER))
   (BLOCK NIL (SET PDC (DIFFERENCE PDC N))
    (SET PDMIN (MIN PDMIN PDC)) (SET PDMAP (NOFF N PDMAP))))
  (FUNCTION (LAPCALL2 NOVALUE)
   NIL (BLOCK ((C INTEGER)
     (M SYMBOL) (H OCTAL) (L SYMBOL) (I INTEGER))
    (SET C PDC)
    (SET M PDMAP)
    A (SET H 0Q)
    (SET I 23)
    B (IF (NULL M) (GO C))
    (SET H (WORDOR H (SHIFT (CAR M) I)))
    (SET M (CDR M))
    (IF (EQ I 0) (GO C))
    (SET I (PLUS I -1))
    (GO B)
    C (SET L (CONS (CONS C H) L))
    (SET C (MAX (DIFFERENCE C 24) 0))
    (IF (GR C PDMIN) (GO A))
    D (IF (LS (CAADR MAPS) C) (GO DD))
    (SET MAPS (CDR MAPS))
    (GO D)
    DD (SET (CAAR MAPS) (MIN (CAAR MAPS) PDMIN))
    (SET PDMIN PDC)
    E (SET I (CDAR MAPS))
    (SET M (CAR L))
    (SET L (CDR L))
    (IF (NULL L) (GO R))
    (REMWORD (SHIFT (PLUS (CAR M) 1) 24) 0 (QUOTE R))
    (SET MAPS (CONS (CONS (CAR M) RLC) MAPS))
    (REMWORD (SHIFT (CDR M) 24) I (QUOTE R))
    (GO E)
    R (SET MAPS (CONS (CONS (CAR M) ILC) MAPS))
    (BPINSTR (SHIFT (CDR M) 24) I (QUOTE R)))))
(BPGEN (FUNCTION (BPINSTR NOVALUE)
   ((C OCTAL) (R INTEGER) (Y SYMBOL)) (BPINST C 0 NIL R Y))
  (FUNCTION (BPINST NOVALUE)
   ((C OCTAL) (L INTEGER) (X SYMBOL) (R INTEGER) (Y SYMBOL))
   (BLOCK ((FL SYMBOL FLUID))
    (SET C (WORDOR C (SHIFT (BPADDR L X ILC (QUOTE L)) 24)
      (BPADDR R Y ILC (QUOTE R))))
    (SET R (BPLOC ILC))
    (SET (CORE R) C)
    (SET ILC (PLUS ILC 1))
    (IF ORGMODE (SET FPP (PLUS FPP 1)))
    (IF OUTLAP (BLOCK ((S SYMBOL (OUTPUT OUTLAP)))
      (PRINOCT R 6) (BLANKS 2) (PRINOCT C 16) (OUTPUT S)))
    (IF FL (LAPNIX (QUOTE RELOC)))))
  (ROUTINE (BPADDR OCTAL)
   ((C INTEGER) (M SYMBOL) (A INTEGER) (H SYMBOL))
   (BLOCK ((U INTEGER))
    (IF (NULL M) (RETURN 0Q) (EQ M (QUOTE A)) (GO R) ORIGIN (GO A))
    (SET U (PLUS (BPLOC FSIZ) (MINUS (IQUOTIENT A 24)) -1))
    (SET (CORE U)
     (WORDOR (CORE U)
      (SHIFT (IF (EQ H (QUOTE L)) 4Q15 2Q15)
       (TIMES -2 (REMAINDER A 24)))))
    A (IF (EQ M (QUOTE F))
     (SET C (PLUS C -1))
     (AND (EQ M (QUOTE R))
      (OR ORIGIN (AND (NOT (LS C 0)) (LS C FSIZ))))
     (SET C (BPLOC C)) (NOT (EQ M (QUOTE S))) (SET FL TRUE))
    R (RETURN (BIT 0 18 C))))
  (ROUTINE (BPLOC INTEGER)
   ((C INTEGER))                                                        0082000
   (IF ORIGIN C FDESC (PLUS (BIT 0 18 (CORE (PLUS (S2O. FDESC) -1)))   0082100
     -1 C) 0))                                                          0082200
  (FUNCTION (REMWORD NOVALUE)                                           0082300
   ((C OCTAL) (R INTEGER) (Y SYMBOL))                                   0082400
   (BLOCK ((ILC INTEGER FLUID RLC))                                     0082500
    (BPINSTR C R Y) (PRINCOM NIL 0) (SET RLC (PLUS RLC 1)))))           0082600
(LAPLIB (FUNCTION (NDUP SYMBOL)                                         0082700
   ((N INTEGER) (X SYMBOL))                                             0082800
   (BLOCK ((U SYMBOL))                                                  0082900
    A (IF (EQ N 0) (RETURN U))                                          0083000
    (SET U (CONS X U)) (SET N (PLUS N -1)) (GO A)))                     0083100
  (FUNCTION (LAPNIX NOVALUE)                                            0083200
   ((M SYMBOL))                                                         0083300
   (BLOCK NIL (SET ERRS TRUE)                                           0083400
    (MESSAGE (CONS (QUOTE UNDEFINED)                                    0083500
      (APPEND (IF (ATOM M) (LIST M) M)                                  0083600
       (APPEND (QUOTE (IN ITEM)) (LIST IT)))))                          0083700
    (MESSAGE (LIST (QUOTE LOCATION)                                     0083800
      (I2O. (BPLOC ILC)) (QUOTE FUNCTION) (CONS FNAME FSEC))))))        0083900
(LAPID (ROUTINE (LAPID SYMBOL)                                          0084000
   ((X SYMBOL))                                                         0084100
   (BLOCK NIL (IF (NOT (CHARP X))                                       0084200
     (SET (BIT 24 18 (CORE (PLUS (S2O. X) 1)))                          0084300
      (I2O. (PLUS 1 (BIT 24 18 (CORE (PLUS (S2O. X) 1)))))))            0084400
    (RETURN X)))                                                        0084500
  (ROUTINE (LAPFREE SYMBOL)                                             0084600
   ((N SYMBOL) (SN SYMBOL))                                             0084700
   (BLOCK ((U SYMBOL (GETFREE N SN)))                                   0084800
    (IF (NULL U) (RETURN NIL))                                          0084900
    (SET (BIT 0 18 (CORE (S2O. U)))                                     0085000
     (I2O. (PLUS 1 (BIT 0 18 (CORE (S2O. U)))))) (RETURN U)))           0085100
  (FUNCTION ((LAPGC . SYS) SYMBOL)                                      0085200
   NIL (BLOCK ((X SYMBOL (MAPCAR LAPSTL FTRANS)))                       0085300
    (SET LAPSTL NIL) (RETURN X)))                                       0085400
  (FUNCTION (PRINCOM NOVALUE)                                           0085500
   ((M SYMBOL) (I INTEGER))                                             0085600
   (BLOCK NIL (IF OUTLAP (BLOCK ((S SYMBOL (OUTPUT OUTLAP)))            0085700
      (BLANKS I) (IF M (PRINT M) (ENDOUT)) (OUTPUT S)))))               0085800
  (FUNCTION (BLANKS NOVALUE)                                            0085900
   ((I INTEGER))                                                        0086000
   (BLOCK NIL (FOR I (STEP I -1 EQ 0) (PRINCH (QUOTE ' )))))            0086100
  (FUNCTION (PRINOCT NOVALUE)                                           0086200
   ((C OCTAL) (I INTEGER))                                              0086300
   (BLOCK NIL (FOR I (STEP I -1 EQ 0)                                   0086400
     (PRINTOKEN (O2I. (BIT 0 3 (SHIFT C (TIMES -3 (PLUS I -1))))))))))) 0086500
(INDEXER DEFINE (((INDEXER (LAMBDA (N L)                                0086600
      (TEDFILER (CONS N (MAPCON L (FUNCTION (LAMBDA (J)                 0086700
           (MAPCON (TEDSEEKER (CAR J))                                  0086800
            (FUNCTION (LAMBDA (J)                                       0086900
              (COND ((OR (ATOM (CAR J))                                 0087000
                 (NOT (MEMBER (CAAR J)                                  0087100
                   (QUOTE (SECTION ROUTINE FUNCTION))))) NIL)           0087200
               (T (LIST (CONS (CAAR J)                                  0087300
                  (CONS (CADAR J)                                       0087400
                   (COND ((NULL (CDDAR J)) NIL)                         0087500
                    (T (LIST (CADDAR J))))))))))))))))))))))            0087600
****END OF FILE DETECTED
```

```
(POSTCOMP (SECTION LAP SYMBOL)
 (DECLARE (ENTRIES SYMBOL OWN ENTRIES))
 (SECTION SYS OCTAL)
 (DECLARE (FPO OWN FPO)
  (FPP OWN FPP)
  (CHO OWN CHO)
  (TRO OWN TRO)
  (TRP OWN TRP)
  (TRM OWN TRM)
  (BPO OWN BPO)
  (BPP OWN BPP)
  (ARO OWN ARO)
  (ARP OWN ARP)
  (LSP OWN LSP)
  (LSO OWN LSO)
  (OBLIST (ARRAY SYMBOL)
   OWN (CONS (QUOTE *SYMBOL)
    (MAPCAR OBLIST (FUNCTION (LAMBDA (J)
       (COND ((NULL (CDR J)) NIL)
        ((MEMBER (CAAR J) (QUOTE (TRUE FALSE)))
         (LIST (QUOTE *IDENTIFIER) (CAAR J))) (T (CAAR J))))))))
  (OBLSIZ INTEGER OWN OBLSIZ) (TRL OWN)))
****END OF FILE DETECTED
```

```
(SECTION (SECTICN (CCMPIL SUPV SYS LISP) SYMBOL))
(HELP (DECLARE (SLIST SYMBCL FLUID)
   (SNAME SYMBCL FLUIC)
   (STYPE SYMBCL FLUIC)
   ((DEBUG . LISP) BCCLEAN FLUID)
   (FKIND SYMBCL FLUIC)
   (FORG SYMBCL FLLID)
   (SCLASS SYMBCL FLUID)
   (PCLASS SYMBCL FLUID)
   (TGO SYMBCL FLUID)
   (FGO SYMBCL FLUID)
   (XGO SYMBCL FLUID)
   (TERGO SYMBCL FLUIC)
   (ALIST SYMBCL FLUIC)
   (APLIST SYMBCL FLUID)
   (IRLIST SYMBCL FLUID)
   (GOLIST SYMBCL FLUID)
   (LABELS SYMBCL FLUID)
   (EXP SYMBCL FLUID)
   (TERMINS SYMBCL FLUIC)
   (LISTING SYMBCL FLUIC)
   (REMOTES SYMBCL FLUIC)
   (REFLIST SYMBCL FLUIC)
   (FNAME SYMBCL FLUID)
   (FTYPE SYMBCL FLUID)
   (VCLASS SYMBCL FLUID)
   (VTYPE SYMBCL FLUIC)
   (VREG SYMBOL FLLID)
   (VADDR SYMBCL FLUID)
   (VIND SYMBOL FLLID)
   (VINV SYMBOL FLLIC)
   (VBYTE SYMBCL FLUID)
   (VBLOT SYMBCL FLUID)
   (XTYPE SYMBCL FLUID)
   (XREG SYMBOL FLLID)
   (XLOC SYMBOL FLLID)
   (XBYTE SYMBCL FLUID)
   (CV SYMBOL FLLIC)
   (CT SYMBOL FLUIC)
   (CM SYMBOL FLUIC)
   (CF SYMBOL FLUIC)
   (CL SYMBOL FLUIC)
   (CI SYMBOL FLUIC)
   (FTLIST SYMBCL FLUID)
   (CRGP SYMBOL FLLID) (INSTRUCTION SYMBOL FLUID))
  (FLNCTICN (CCMER2 SYMBCL)
   ((X SYMBOL) (Y SYMBCL)) (COMERR (CONS X Y)))
  (FLNCTICN COMERR (J)
   (BLOCK NIL (SET ERRFLG TRUE)
    (SET VADDR (GENID))
    (SET VCLASS (QLCTE LCC))
    (SET VTYPE (QUCTE SYMBCL))
    (RETURN (SLPCTY (APPENC (QUOTE (ERROR.. IN FUNC))
       (LIST FNAME J))))))
  (FLNCTICN ((CEBUGGING . COMPIL) BCCLEAN)
   NIL (ANC (NCT (FKIND . COMPIL)) (CEBUG . LISP)))
  (FLNCTICN (ATTACH SYMBCL)
   ((L SYMBOL)) (SET LISTING (CONS L LISTING)))
  (FLNCTICN (ATTACHGC SYMBCL)
   ((L SYMBOL))
   (BLOCK NIL (IF (LASTBRANCH) (RETLRN NIL))
    (ATTACH (LIST (QUCTE BUC) (LABELER L)))
    (IF (NCT (MEMBER L LABELS))
     (SET GOLIST (CONS (CONS (QUOTE GO) LISTING) GOLIST)))))
  (FUNCTION (ATTACHLAB SYMBOL)
   ((L SYMBOL)) (BLOCK NIL (ATTACH L) (SET LABELS (CONS L LABELS))))
  (FUNCTION (REMOTE SYMBOL)
   ((L SYMBOL)) (SET REMOTES (CONS L REMOTES)))
  (FUNCTION (BLOTTO SYMBOL)
   NIL (SET VBLOT (QUOTE (AC L B X1 X2 X3 X4))))
  (FUNCTION (BLOTCH SYMBOL)
   ((X SYMBOL))
   (IF (MEMBER X VBLOT) NIL (SET VBLOT (CONS X VBLOT))))
  (FUNCTION (UNION SYMBOL)
   ((A SYMBOL) (B SYMBOL))
   (BLOCK NIL L (IF (NULL B)
     (RETURN A) (NOT (MEMBER (CAR B) A)) (SET A (CONS (CAR B) A)))
    (SET B (CDR B)) (GO L)))
  (FUNCTION (FVTYPE SYMBOL)
   ((X SYMBOL))
   (IF (NUMBP X)
    (IF (FIXP X)
     (IF (EQUALN X (WORDOR X 0Q)) (QUOTE OCTAL) (QUOTE INTEGER))
     (QUOTE REAL))
    (MEMBER X (QUOTE (TRUE FALSE NIL)))
    (QUOTE BOOLEAN) (QUOTE SYMBOL)))
  (FUNCTION (LABELER SYMBOL) ((X SYMBOL)) (LIST (QUOTE LABEL) X))
  (FUNCTION (GETN SYMBOL)
   ((L SYMBOL) (P SYMBOL))
   (BLOCK NIL A (IF (NULL L)
     (RETURN L) (EQUALN (CAR L) P) (RETURN (CADR L)))
    (SET L (CDR L)) (GO A)))
  (FUNCTION (ITYPE SYMBOL)
   ((J SYMBOL))
   (IF (EQN J (QUOTE NUMBER))
    0.0 (EQN J (QUOTE OCTAL))
    0Q (EQN J (QUOTE INTEGER))
    0 (EQN J (QUOTE REAL))
    0.0 (EQN J (QUOTE FUNCTIONAL)) (QUOTE (FMTRAP . SYS)) NIL)))
(TCP1 (FUNCTION (GENID2 SYMBOL) NIL (GENID))
  (DECLARE (INST1 SYMBOL FLUID)
   (INST2 SYMBOL FLUID) (INST3 SYMBOL FLUID))
  (FUNCTION (NEXLST SYMBOL)
   NIL (BLOCK ((X SYMBOL))
    (RETURN (IF (CDDDR LISTING)
      (IF (SET X (CAR LISTING))
       (BLOCK NIL (SET LISTING (CDR LISTING)) (RETURN X))
       (BLOCK NIL (SET LISTING (CDR LISTING))
        (RETURN (NEXLST)))) NIL))))
  (FUNCTION (REVLST SYMBOL)
   NIL (BLOCK ((INST1 SYMBOL)
     (INST2 SYMBOL) (INST3 SYMBOL) (LST SYMBOL))
    (SET INST1 (CAR LISTING))
    (SET INST2 (CADR LISTING))
    (SET INST3 (CADDR LISTING))
    (SET LISTING (CDDDR LISTING))
    A (OR (REVACR) (REVBUC) (REVLDA) (REVSTZ) (REVWRD) (REVZER))
    (IF INST1 (SET LST (CONS INST1 LST)))
    (SET INST1 INST2)
    (SET INST2 INST3)
    (IF (SET INST3 (NEXLST)) (GO A))
    (RETURN (NCONC (NCONC (REVERSE LISTING)
       (LIST INST2 INST1)) LST))))
  (FUNCTION (REVTST BOOLEAN)
   ((A SYMBOL) (B SYMBOL))
   (AND (SIM (QUOTE (STF . S.)) A)
    (SIM (QUOTE (LDA . S.)) B) (EQ (CDR A) (CDR B))))
  (FUNCTION (AORTST BOOLEAN)
   ((L SYMBOL))
   (OR (SIM (QUOTE (OR. ((NUMBER 1) S) (1 (L567.7 R S)))) L)
    (BLOCK NIL (SET VINV (NOT VINV)))
    (SIM (QUOTE (OR. ((NUMBER -1) S) (-1 (L567.7 R S)))) L)))
  (FUNCTION (REVZER BOOLEAN)
   NIL (IF (SIM (QUOTE (XOR (NUMBER (OR. 0Q 777777777777777Q))))
     INST1)
    (BLOCK NIL (IF (EQUALN (CADADR INST1) 0Q)
      (GO X)
      (IF (SIM (QUOTE ((OR. LDA LDC) . S.)) INST2)
       (BLOCK NIL (SET INST2 (CONS (CDR (FINDN (CAR INST2)
            (QUOTE ((LDA . LDC) (LDC . LDA))))) (CDR INST2)))
        X (SET INST1 NIL)) (SET INST1 (QUOTE (LDC A.)))))
     (RETURN TRUE)) FALSE))
  (FUNCTION (REVSTZ BOOLEAN)
   NIL (IF (NOT (SIM (QUOTE (BUC (LABEL ID.))) INST1))
    NIL (EQ (QUOTE (STZ A.)) INST2)
    (REVENT (QUOTE (ENTRY STZENT)))
    (SIM (QUOTE (LDA 1 ((OR. L4567.7 L567.7) R))) INST2)
    (REVENT (QUOTE (ENTRY ONENT))) NIL))
  (FUNCTION (REVENT BOOLEAN)
   ((X SYMBOL))
   (BLOCK NIL (SET INST2 (LIST (QUOTE BSX) X 4 (CADR INST1)))
    (SET INST1 NIL) (RETURN TRUE)))
  (FUNCTION (REVAOR BOOLEAN)
   NIL (BLOCK ((VINV SYMBOL))
    (IF (AND (NOT (ATOM INST2))
      (CDR INST2)
      (REVTST INST1 INST3)
      (AORTST (CDR INST2))
      (OR (EQN (CAR INST2) (QUOTE ADD))
       (AND (EQN (CAR INST2) (QUOTE SUB))
        (BLOCK NIL (SET VINV (NOT VINV)) (RETURN TRUE)))))
     (BLOCK NIL (SET INST1 (SET INST2 NIL))
      (SET INST3 (CONS (IF VINV (QUOTE SOR) (QUOTE AOR))
        (CDR INST3))) (RETURN TRUE)))))
  (FUNCTION (REVBUC BOOLEAN)
   NIL (AND (SIM (QUOTE (BUC (LABEL ID.))) INST2)
    (EQN (CADADR INST2) INST1)
    (BLOCK NIL (SET INST2 INST1) (SET INST1 NIL) (RETURN TRUE))))
  (FUNCTION (REVLDA BOOLEAN)
   NIL (AND (OR (REVTST INST1 INST2) (REVTST INST2 INST1))
    (BLOCK NIL (SET INST1 NIL) (RETURN TRUE))))
  (FUNCTION (REVWRD BOOLEAN)
   NIL (BLOCK NIL (IF (AND (SIM (QUOTE ((OR. ANA ORA XOR) N. L.))
       INST1) (MEMBER (QUOTE R) (CADDR INST1)))
     (SET INST1 (LIST (CAR INST1)
       (LIST (QUOTE NUMBER) (CADR INST1))))) (RETURN NIL))))
(DECL (FUNCTION (FUNCTIO SYMBOL)
  ((EXP SYMBOL))
   (BLOCK ((FN SYMBOL (FNAMER)))
    (RETURN (BLOCK ((FKIND SYMBOL (EQ (QUOTE ROUTINE) (CAR EXP)))
       (FNAME SYMBOL (CAR FN))
       (FTYPE SYMBOL (CADR FN)) (EXPR SYMBOL (CADDDR EXP)))
      (RETURN (BLOCK ((XTYPE SYMBOL (IF (NQ (QUOTE NOVALUE) FTYPE)
           FTYPE (BLOCK NIL (SET FTYPE STYPE)))))
        (RETURN (BLOCK ((LISTING SYMBOL)
           (ALIST SYMBOL)
           (FORG SYMBOL)
           (OROP SYMBOL)
           (FTLIST SYMBOL)
```

```
         (REMOTES SYMBOL) (REFLIST SYMBOL) (VBLOT SYMBOL))
        (FNDEC EXP)
        (ATTACH (CAR EXP))
        (ATTACH FN)
        (FNBIND)
        (COMLCK 4)
        (COMVAL (IF (NULL XTYPE)
          (LIST (QUOTE BLOCK) NIL EXPR) EXPR)
         XTYPE (QUOTE VALUE) (QUOTE AC))
        (IF FORG (BLOCK NIL (SET (CDR ORGP)
           (CONS (CAR ORGP) (CDR ORGP))) (SET (CAR ORGP) FORG)))
        (ATTACH (QUOTE (END)))
        (ATTACH (QUOTE (RETURN)))
        (LSTLST REMOTES)
        (RETURN (LIST (QUOTE LAP) (REVLST) REFLIST SNAME)))))))))
(FUNCTION (FNDEC SYMBOL)
 ((EXP SYMBOL))
 (BLOCK ((FNA SYMBOL (FNAMER)))
  (RETURN (BLOCK ((FNAME SYMBOL (CAR FNA)))
    (RETURN (BLOCK ((LISTING SYMBOL)
       (ALIST SYMBOL)
       (REFLIST SYMBOL)
       (FTLIST SYMBOL) (ORGP SYMBOL) (VBLOT SYMBOL))
      (FNBIND)
      (MAKEFREE (CAR FNAME)
       (CDR FNAME)
       (CAR EXP)
       (CONS (QUOTE FUNCTIONAL) (CADR FNA) (REVERSE FTLIST))
       (QUOTE VALUE)) (RETURN FNAME)))))))
(FUNCTION (MAKEFREE . COMPIL)
 (A B C D E) ((KEEPER . SUPV) ((MAKEFREE . SYS) A B C D E)))
(FUNCTION DECL1 (J)
 (BLOCK ((DV FLUID) (DT FLUID) (DF FLUID) (DL FLUID) (DM FLUID) K)
  (IF (NULL (SET K (ORDER J)))
   (RETURN NIL)
   (EQ DF (QUOTE LEXICAL))
   (GO ERR)
   (MEMBER K (QUOTE (NORMAL OWN)))
   (BLOCK NIL (MAKEFREE (GETVAR DV)
     (GETSEC DV) (IF DF DF (QUOTE FREE)) (GETYPE DT) (GETLOC DL))
    (IF (AND (EQ PASS 2) DI)
     (EVAL (LIST (QUOTE SET) DV (CAR DI)))) (RETURN DV))
   (EQ K (QUOTE MEANS))
   (BLOCK ((X (BLOCK ((DV FLUID DI) (REFLIST FLUID))
       (IF (GETGLB DV) (RETURN DV)))))
    (RETURN (IF (NOT X)
      (COMER2 J (QUOTE (NO PRIOR DECLARATION)))
      (BLOCK NIL (IF (EQ DM (SET DV (CONS (GETVAR DV)
           (GETSEC DV)))) (SET X DM))
       (MAKEFREE (CAR DV) (CDR DV) (QUOTE MEANS) (CAR X) (CDR X))
       (RETURN DV)))))
   (LABEL ERR (COMER2 J (QUOTE (ILLEGAL TOP DECLARATION)))))))
(FUNCTION FVLIS1 (X)
 (BLOCK ((J (FVLIST X)))
  (RETURN (IF (MEMBER (CAR J) (QUOTE (MACRO INSTRUCTIONS)))
    (LIST (CAR J) (CADR J) X) J))))
(FUNCTION (GETSEC SYMBOL)
 ((V SYMBOL))
 (IF (ATOM V) SNAME (SIM (QUOTE (ID. . ID.)) V) (CDR V) SNAME))
(FUNCTION (GETVAR SYMBOL) ((V SYMBOL)) (IF (ATOM V) V (CAR V)))
(FUNCTION (DEFAULT SYMBOL) ((X SYMBOL)) (SECSET SLIST X))
(FUNCTION (SECSET SYMBOL)
 ((N SYMBOL) (TYP SYMBOL))
 (BLOCK NIL (IF (NOT (FTYPP TYP))
   (COMER2 TYP (QUOTE (INVALID DEFAULT TYPE)))
   (RETURN (CONS (SET SNAME (CAR (SET SLIST (IF (MEMBER (QUOTE LISP)
          (IF (ATOM N) (SET N (LIST N)) N))
         N (APPEND N (QUOTE (LISP))))))) (SET STYPE TYP))))))
(FUNCTION (ANYVARS SYMBOL)
 ((D SYMBOL))
 (NOT (SIM (QUOTE ((C. 0 1000 (IC. SWITCH . S.)))) D)))
(FUNCTION (GETLEX SYMBOL)
 ((V SYMBOL))
 (BLOCK ((A SYMBOL))
  (IF (NULL (SET A (FIND V ALIST))) (GO L))
  (SET DV V)
  (RETURN (CONS (QUOTE LEXICAL) (CDR A)))
  L (IF (NULL (SET A (FIND V APLIST))) (RETURN NIL))
  (SET DV V)
  (RETURN (CONS (QUOTE LEXICAL)
    (IF (MEMBER A IRLIST)
     (LIST (CADR A) (QUOTE LOC))
     (BLOCK NIL (SET IRLIST (CONS A IRLIST))
      (RETURN (CDR A))))))))
(FUNCTION (GETGLB SYMBOL)
 ((V SYMBOL))
 (IF (NOT (ATOM V))
  (GETDEC (CAR V) (CDR V))
  (BLOCK ((SLIST SYMBOL SLIST))
   (RETURN (BLOCK ((A SYMBOL))
     L (IF (NULL SLIST)
      (IF (AND (NQ (O2S. (BIT 24 18 (CORE (S2O. V)))) V)
        (EQ (O2S. (BIT 0 18 (CORE (PLUS 1 (BIT 24 18 (CORE (S2O. V))
             ))))) V))
       (BLOCK (ERRFLG)
        (COMERR (APPEND (QUOTE (WILL USE DECLARATION FOR))
          (LIST (CONS V (SET A (O2S. (BIT 24 18 (CORE (BIT 24 18
                (CORE (S2O. V))))))))))))) (RETURN (GETDEC V A)))
       (RETURN NIL)) (SET A (GETDEC V (CAR SLIST))) (RETURN A))
     (SET SLIST (CDR SLIST)) (GO L))))))
(FUNCTION (GETFRV SYMBOL)
 ((V SYMBOL))
 (BLOCK ((A SYMBOL (IF (ATOM V) (GETLEX V) NIL)))
  (RETURN (IF A A (GETGLB V)))))
(FUNCTION (GETDEC SYMBOL)
 ((V SYMBOL) (S SYMBOL))
 (BLOCK ((A SYMBOL))
  (IF (SET A (GETDC V S)) (ADREF A)) (RETURN A)))
(FUNCTION (GETDC SYMBOL)
 ((V SYMBOL) (S SYMBOL))
 (BLOCK ((A SYMBOL))
  (RETURN (IF (NULL (SET A (FVLIST (GETFREE V S))))
    (IF (AND (EQN S (QUOTE LISP))
      (SIM (QUOTE (C (C. 1 10000 (CR. A D)) R))
       (SET A (EXPLODE V)))) (LIST (QUOTE MACRO) NIL A) NIL)
    (EQN (CAR A) (QUOTE MEANS))
    (BLOCK NIL (IF (NOT DM) (SET DM (CONS V S)))
     (RETURN (GETDC (CADR A) (CADDR A))))
    (BLOCK NIL (SET DV (CONS V S)) (RETURN A))))))
(FUNCTION (ADREF SYMBOL)
 ((A SYMBOL))
 (BLOCK NIL (IF (AND (NOT (MEMBER (CAR A)
      (QUOTE (INSTRUCTIONS MACRO)))) (NOT (FIND DV REFLIST)))
   (SET REFLIST (CONS (CONS DV A) REFLIST)))))
(FUNCTION (FNBIND SYMBOL)
 NIL (BLOCK ((VADDR SYMBOL) (VCLASS SYMBOL) (VTYPE SYMBOL))
  (RETURN (BIND (CADDR EXP)
    (FUNARG NOVALUE ((X SYMBOL))
     (BLOCK NIL (SET ORGP (ATTACH X))
      (IF X (ATTACH (QUOTE (STF TOP.))))
      (ATTACH (QUOTE (BEGIN)))))
    (FUNARG SYMBOL ((X SYMBOL))
     (BLOCK NIL (IF (NOT (EQN X (QUOTE NORMAL)))
       (GO ER2) DI (GO ER1))
      (SET FTLIST (CONS (IF (EQN DL (QUOTE LOC))
         (LIST (FTYPER DT) DL) (FTYPER DT)) FTLIST))
      (RETURN X)
      ER1 (SET X (QUOTE PRESET))
      ER2 (COMER2 X (QUOTE (ILLEGAL IN FUNC DEC))))))))
(FUNCTION (BKBIND SYMBOL)
 NIL (BLOCK ((ANY BOOLEAN (ANYVARS (CADR EXP))))
  (IF ANY (ATTACH (QUOTE (BLOCK))))
  (BLOCK ((P (BIND (CADR EXP)
      (FUNARG NOVALUE ((X SYMBOL))
       (ATTACH (CONS (QUOTE DECLARE) X)))
      (FUNARG SYMBOL ((X SYMBOL))
       (BLOCK ((Y SYMBOL) (Z SYMBOL))
        (IF (AND (EQN DL (QUOTE LOC)) (NULL DI))
         (SET X (QUOTE (LOC WITHOUT PRESET)))
         (EQN X (QUOTE NORMAL))
         (GO NORMAL)
         (EQN X (QUOTE SWITCH))
         (GO SWITCH) (EQN X (QUOTE ASSIGNED)) (GO ASSIGNED))
        (COMER2 X (QUOTE (ILLEGAL IN BLOCK DEC)))
        (RETURN NIL)
        SWITCH (COMSWITCH DI DV)
        (RETURN NIL)
        NORMAL (SET Y (FTYPER DT))
        ASSIGNED (SET Z (COMPUSH (IF DI (CAR DI) (ITYPE Y))
          Y (GETLOC DL)))
        (IF (NULL Y)
         (SET DT (IF (EQN Z (QUOTE FUNCTIONAL))
           (QUOTE SYMBOL) Z))) (RETURN X))))))
   (RETURN (IF ANY P TRUE)))))
(FUNCTION (BIND SYMBOL)
 ((L SYMBOL)
  (F1 (FUNCTIONAL NOVALUE SYMBOL))
  (F2 (FUNCTIONAL SYMBOL SYMBOL)))
 (BLOCK ((A SYMBOL)
   (H SYMBOL)
   (FL SYMBOL)
   (TEMP SYMBOL)
   (X SYMBOL)
   (Y SYMBOL)
   (DV SYMBOL) (DT SYMBOL) (DF SYMBOL) (DL SYMBOL) (DI SYMBOL))
  A (IF (NULL L)
   (GO D)
   (NULL (SET X (ORDER (CAR L))))
   (GO C)
   (EQN X (QUOTE NORMAL)) (GO NEXT) (NULL (F2 X)) (GO C))
  NEXT (IF (EQN DF (QUOTE LEXICAL))
   (GO LEXICAL)
   (AND (SET TEMP (GETCC (GETVAR (SET Y DV)) (GETSEC DV)))
    (EQN (CAR TEMP) (QUOTE FLUID)))
   (GO VERIFY2)
   (NULL DF)
   (GO LEXCAL)
   (BLOCK NIL (SET DF (QUOTE FREE)) (RETURN TEMP)) (GO VERIFY1))
  (GENDTCL)
```

```
   (DECL1 (LIST DV DT DF DL))
   (GO NEXT)
   VERIFY1 (IF (NOT (OR (EQN (CAR TEMP) (QUOTE FREE))
      (EQN (CAR TEMP) (QUOTE FLUID))))
    (COMER2 DV (QUOTE (BAD REDEF))))
   VERIFY2 (SET DF (CAR TEMP))
   (IF (NULL DT)
    (SET DT (CADR TEMP))
    (NOT (EQ DT (CADR TEMP))) (COMER2 DV (QUOTE (TYPE MISMATCH))))
   (IF (NULL DL)
    (SET DL (CADDR TEMP))
    (NOT (EQN DL (CADDR TEMP)))
    (COMER2 DV (QUOTE (TRANS MODE MISMATCH))))
   (ADREF TEMP)
   ONWARD (IF (AND (EQN X (QUOTE NORMAL)) (NULL (F2 X))) (GO C))
   (SET H (CONS (LIST DV DT DF DL) H))
   (IF (NOT (ATOM DV))
    (SET FL (CONS DV FL)) (SET A (CONS (LIST DV DT DL) A)))
   C (SET L (CDR L))
   (GO A)
   LEXCAL (SET DV Y)
   (SET DF (QUOTE LEXICAL))
   LEXICAL (GENDTDL)
   (GO ONWARD)
   D (F1 (REVERSE H))
   (SET ALIST (NCONC A ALIST))
   (RETURN (IF FL (BLOCK NIL (ATTACH (CONS (QUOTE FLBIND)
        (SET FL (REVERSE FL)))) (BLCTTO) (RETURN FL)) NIL))))
 (FUNCTION (GENDTDL SYMBOL)
  NIL (BLOCK NIL (SET DT (GETYPE DT)) (SET DL (GETLOC DL))))
 (FUNCTION (GETYPE SYMBOL) ((X SYMBOL)) (IF X X STYPE))
 (FUNCTION (GETLOC SYMBOL) ((X SYMBOL)) (IF X X (QUOTE VALUE)))
 (FUNCTION (ORDER SYMBOL)
  ((L SYMBOL))
  (BLOCK ((M SYMBOL (QUOTE NORMAL)))
   (RETURN (BLOCK NIL (SET DT (SET DF (SET DL (SET DI NIL))))
     (IF (SIM (QUOTE V.) L)
      (SET L (LIST L))
      (NOT (LISTP L))
      (RETURN (BLOCK NIL (COMER2 L (QUOTE (IMPROPER DECLARATION))))))
     (IF (NOT (SIM (QUOTE V.) (SET DV (CAR L))))
      (RETURN (BLOCK NIL (COMER2 DV (QUOTE (IMPROPER VARIABLE)))))
      (NULL (SET L (CDR L)))
      (RETURN (BLOCK NIL (TESLEX) (RETURN M)))
      (TYPEP (CAR L))
      (SET DT (STANTP (CAR L)))
      (EQN (CAR L) (QUOTE MEANS))
      (GO MEANS)
      (EQN (CAR L) (QUOTE ASSIGNED))
      (GO ASSIGNED)
      (EQN (CAR L) (QUOTE SWITCH)) (GO SWITCH) (GO TRY2))
     (IF (NULL (SET L (CDR L))) (RETURN M))
     TRY2 (IF (NOT (MEMBER (CAR L)
        (QUOTE (FLUID FREE LEXICAL OWN))))
      (GO TRY3) (EQN (SET DF (CAR L)) (QUOTE OWN)) (SET M DF))
     TRYA (TESLEX)
     (IF (NULL (SET L (CDR L))) (RETURN M))
     TRY3 (TESLEX)
     (IF (NOT (TMODEP (CAR L))) (GO TRY4))
     (SET DL (CAR L))
     LESS1 (IF (NULL (SET L (CDR L))) (RETURN M))
     TRY4 (IF (NOT (EQ (LENGTH (SET DI L)) 1))
      (COMER2 (CAR L) (QUOTE (IGNORED IN DEC))))
    (GO LESS1)
    MEANS (IF (SIM (QUOTE (MEANS V.)) L) (GO COMMON))
    ERROR (COMER2 (CAR L) (QUOTE (DEC FORMAT ERROR)))
    (RETURN NIL)
    ASSIGNED (IF (NOT (SIM (QUOTE (ASSIGNED (ANY. LOC VALUE) S.)) L)
     ) (GO ERROR))
    (SET DF (QUOTE LEXICAL))
    (SET M (QUOTE ASSIGNED))
    (GO TRYA)
    COMMON (SET DI (CADR L))
    (RETURN (CAR L)) SWITCH (SET DI (CDR L)) (RETURN (CAR L))))))
(FUNCTION (TESLEX SYMBOL)
 NIL (IF (NOT (ATOM DV))
  (BLOCK NIL (IF (EQN DF (QUOTE LEXICAL))
    (COMER2 DV (QUOTE (INVALID AS LEXICAL))) DF (RETURN NIL))
   (SET DF (QUOTE FREE))) NIL))
(FUNCTION (FPNAME SYMBOL)
 ((N SYMBOL) (TY SYMBOL)) (LIST (CONS (GETVAR N) (GETSEC N)) TY))
(FUNCTION (FNAMER SYMBOL)
 NIL (BLOCK ((N SYMBOL (CADR EXP)))
  (RETURN (IF (SIM (QUOTE V.) N)
    (FPNAME N STYPE)
    (SIM (QUOTE (V. V.)) N)
    (BLOCK ((TY SYMBOL (CADR N)))
     (RETURN (IF (VTYPEP TY)
       (FPNAME (CAR N) TY)
       (BLOCK NIL (COMER2 TY (QUOTE (ILL FUNC TYPE)))
        (RETURN (FPNAME (CAR N) STYPE))))))
    (BLOCK NIL (COMER2 N (QUOTE (BAD FUN NAME)))
     (RETURN (FPNAME (GENID) STYPE)))))))
IDOLE (FUNCTION (COMTERM SYMBOL)
 ((X SYMBOL))
 (BLOCK ((SCLASS SYMBOL)
   (PCLASS SYMBOL)
   (VCLASS SYMBOL)
   (VTYPE SYMBOL)
   (VREG SYMBOL)
   (VADDR SYMBOL) (VIND SYMBOL) (VINV SYMBOL) (VBYTE SYMBOL))
  (COMEXP X)
  (ATTACH (VLIST)) (SET TERMINS (CONS LISTING TERMINS))))
(FUNCTION (COMSTAT SYMBOL)
 ((X SYMBOL))
 (BLOCK ((SCLASS SYMBOL)
   (VCLASS SYMBOL)
   (VTYPE SYMBOL)
   (VREG SYMBOL)
   (VADDR SYMBOL) (VIND SYMBOL) (VINV SYMBOL) (VBYTE SYMBOL))
  (SET SCLASS (QUOTE TRUE)) (COMEXP X)))
(FUNCTION (COMVAL SYMBOL)
 ((X SYMBOL) (XTYPE SYMBOL) (XLOC SYMBOL) (XREG SYMBOL))
 (BLOCK ((SCLASS SYMBOL)
   (PCLASS SYMBOL)
   (VCLASS SYMBOL)
   (VTYPE SYMBOL)
   (VREG SYMBOL)
   (VADDR SYMBOL)
   (VIND SYMBOL) (VINV SYMBOL) (VBYTE SYMBOL) (TERGO SYMBOL))
  (COMEXP X)
  (SET X VTYPE)
  (IF (EQN (GETLOC XLOC) (QUOTE LOC))
   (BLOCK NIL (IF (NOT (EQN XTYPE VTYPE))
     (COMERR (QUOTE (NO TYPES FOR LOC ARG))))
    (MAKELOC) (SET XTYPE VTYPE) G01568))
  (MOVACTIVE (MAKTYP) XREG NIL) (RETURN X)))
(FUNCTION (COMPUSH SYMBOL)
 ((X SYMBOL) (XTYPE SYMBOL) (XLOC SYMBOL))
 (BLOCK ((SCLASS SYMBOL)
   (PCLASS SYMBOL)
   (VCLASS SYMBOL)
   (VTYPE SYMBOL)
   (VREG SYMBOL)
   (VADDR SYMBOL)
   (VIND SYMBOL) (VINV SYMBOL) (VBYTE SYMBOL) (TERGO SYMBOL))
  (COMEXP X)
  (SET X VTYPE)
  (IF (EQN (GETLOC XLOC) (QUOTE LOC))
   (BLOCK NIL (IF (AND XTYPE (NOT (EQN XTYPE VTYPE)))
     (COMERR (QUOTE (NO TYPES FOR LOC ARG))))
    (MAKELOC) (SET XTYPE VTYPE) G01569))
  (MOVPDS (IF (NULL XTYPE)
    VTYPE (EQN XTYPE (QUOTE NUMBER))
    (IF (MEMBER VTYPE (QUOTE (REAL INTEGER OCTAL)))
     VTYPE (QUOTE REAL)) XTYPE) NIL) (RETURN X)))
(FUNCTION (MAKTYP SYMBOL)
 NIL (IF (NULL XTYPE)
  VTYPE (EQN XTYPE (QUOTE NUMBER))
  (IF (MEMBER VTYPE (QUOTE (REAL INTEGER OCTAL)))
   VTYPE (QUOTE REAL)) XTYPE))
(FUNCTION (COMTERMINS SYMBOL)
 NIL (BLOCK ((W SYMBOL) (X SYMBOL))
  (IF (NULL (SET W TERMINS))
   (RETURN (COMERR (QUOTE (NO EXIT GIVEN)))) (SET X XTYPE) (GO C))
  (SET X (GVTYPE (CAAR W)))
  A (IF (NULL (SET W (CDR W)))
   (GO C) (EQN X (GVTYPE (CAAR W))) (GO A))
  (SET X (QUOTE SYMBOL))
  C (SET W TERMINS)
  (BLOCK ((LISTING SYMBOL)
    (VCLASS SYMBOL)
    (VTYPE SYMBOL)
    (VREG SYMBOL)
    (VADDR SYMBOL) (VIND SYMBOL) (VINV SYMBOL) (VBYTE SYMBOL))
   L (IF (NULL W) (GO G01570))
   (INHERIT (CAAR W))
   (IF (NULL VCLASS) (GO NL))
   (SET LISTING NIL)
   (MOVACTIVE X (QUOTE AC) NIL)
   (IF (NULL LISTING) (GO NL))
   (SET (CAR (CAR W)) (CAR LISTING))
   (IF (CDR LISTING)
    (SET (CDR (CAR W)) (NCONC (CDR LISTING) (CDAR W))))
   I (SET W (CDR W))
   (GO L) NL (SET (CAR (CAR W)) NIL) (GO I) G01570)
  (SET VTYPE X)
  (SET VREG (QUOTE AC)) (SET VCLASS (QUOTE ACTIVE))))
(FUNCTION (COMEXP SYMBOL)
 ((EXP SYMBOL))
 (IF (SIM (QUOTE V.) EXP)
  (COMVAR EXP)
  (ATOM EXP)
  (COMDATUM)
  (NOT (SIM (QUOTE V.) (CAR EXP)))
  (COMER2 (CAR EXP) (QUOTE (ILLEGAL FORM NAME)))
  (BLOCK ((D SYMBOL) (DV SYMBOL))
   (RETURN (IF (NULL (SET D (GETFRV (CAR EXP))))
     (COMER2 (CAR EXP) (QUOTE (NO DEC YET)))
    (EQN (CAR D) (QUOTE INSTRUCTIONS))
    (BLOCK ((Y (FUNCTIONAL SYMBOL)))
     (SET Y (CADDR D)) (RETURN (Y)))
    (EQN (CAR D) (QUOTE MACRO))
    (IF (NOT (ATOM (CADDR D)))
     (COMEXP (MAKECARCDR (CDADDR D)))
     (BLOCK ((Z (FUNCTIONAL SYMBOL SYMBOL)))
      (SET Z (CADDR D)) (RETURN (COMEXP (Z EXP)))))
    (ATYPEP (CADR D))
    (COMSUB D)
    (AND FKIND (NOT (EQN (CAR D) (QUOTE ROUTINE))))
    (COMER2 DV (QUOTE (USED IN ROUTINE)))
    (FUNTYP (CADR D))
    (COMFUNC DV D EXP)
    (COMERR (APPEND (QUOTE (I DO NOT BELIEVE)) (LIST DV))))))))
(FUNCTION (COMPAR SYMBOL)
 ((DEC SYMBOL) (E SYMBOL))
 (BLOCK ((VT SYMBOL) (VL SYMBOL))
  L (IF (AND (NULL E) (NULL DEC))
   (RETURN NIL) (NOT (AND E DEC)) (RETURN (COMLCK 0)))
  (SET VT (IF (ATOM (SET VL (CAR DEC))) VL (CAR VL)))
  (SET VL (IF (ATOM VL) (QUOTE VALUE) (CADR VL)))
  (IF (SET DEC (CDR DEC))
   (COMPUSH (CAR E) VT VL) (COMVAL (CAR E) VT VL (QUOTE AC)))
  (SET E (CDR E)) (GO L)))
(FUNCTION (COMFUNC SYMBOL)
 ((NAM SYMBOL) (DECL SYMBOL) (X SYMBOL))
 (BLOCK NIL (IF (AND (DEBUGGING)
    (NOT (EQ (CAR DECL) (QUOTE ROUTINE)))
    (NOT (EQ (CAR DECL) (QUOTE FUNCTION))))
   (COMSTAT (LIST (QUOTE (FUNCHK . DEBUG))
     (LIST (QUOTE QUOTE) (CADR DECL)) NAM)))
  (ATTACH (QUOTE (ARGS)))
  (COMPAR (CDDADR DECL) (CDR X))
  (ATTACH (LIST (QUOTE CALL)
    (IF (OR (EQN (CAR DECL) (QUOTE ROUTINE))
      (EQN (CAR DECL) (QUOTE FUNCTION)))
     NAM (BLOCK NIL (ATTACH (LIST (QUOTE LDB) NAM (MAKIND DECL)))
      (ATTACH (QUOTE (STB (FMCALL . SYS))))
      (RETURN (QUOTE (FMCALL . SYS)))))))
  (BLOTTO)
  (IF (EQN (SET VTYPE (CADADR DECL)) (QUOTE NOVALUE))
   (BLOCK NIL (SET VTYPE (QUOTE BOOLEAN))
    (SET VCLASS (QUOTE DATUM)) G01571)
   (BLOCK NIL (SET VCLASS (QUOTE ACTIVE))
    (SET VREG (QUOTE AC)) G01572))))
(FUNCTION (COMVAR SYMBOL)
 ((DV SYMBOL))
 (BLOCK ((D SYMBOL))
  (IF (OR (NULL (SET D (GETFRV DV)))
    (MEMBER (CAR D) (QUOTE (MACRO INSTRUCTIONS ROUTINE))))
   (RETURN (COMER2 DV (QUOTE (ILLEGALLY USED))))
   (EQN (CAR D) (QUOTE FUNCTION)) (GO FUNC))
  (SET VCLASS (QUOTE LOC))
  (SET VADDR DV)
  (SET VIND (OR (EQN (CADDR D) (QUOTE LOC))
    (NOT (MEMBER (CAR D) (QUOTE (LEXICAL OWN))))))
  (GO L)
  FUNC (ATTACH (LIST (QUOTE LDA) DV (QUOTE (2Q1 R L4567.7))))
  (BLOTCH (SET VREG (QUOTE AC)))
  (SET VCLASS (QUOTE ACTIVE))
  L (RETURN (SET VTYPE (FTYPER (CADR D))))))
(FUNCTION (COMDATUM SYMBOL)
   NIL (BLOCK NIL (SET VCLASS (QUOTE DATUM))
    (SET VTYPE (FVTYPE EXP)) (SET VADDR EXP)))
  (FUNCTION (COMSUB SYMBOL)
   ((D SYMBOL))
   (BLOCK NIL (COMVAL (IF (DEBUGGING)
      (CONS (QUOTE (ARYCHK . DEBUG))
       DV (LIST (QUOTE QUOTE) (CADADR D)) (CDR EXP)) (CADR EXP))
     (QUOTE INTEGER) NIL (QUOTE AC))
    (ATTACH (LIST (QUOTE ADD)
      DV (LIST (QUOTE T) (MAKIND D) (QUOTE L01234567.3))))
    (SET VCLASS (QUOTE LOC))
    (SET VTYPE (CADADR D)) (SET VREG (QUOTE AC)) (SET VADDR D)))
  (FUNCTION (MAKIND SYMBOL)
   ((D SYMBOL))
   (IF (OR (EQN (CADDR D) (QUOTE LOC))
     (MEMBER (CAR D) (QUOTE (FREE FLUID)))) (QUOTE I) 0))
  (FUNCTION (MAKECARCDR SYMBOL)
   ((J SYMBOL))
   (IF (NULL (CDR J))
    (IF (COMLOK 2) NIL (CADR EXP))
    (LIST (IF (EQN (CAR J) (QUOTE A)) (QUOTE CAR) (QUOTE CDR))
     (MAKECARCDR (CDR J)))))
  (MACRO ((ORG . LISP) SYMBOL)
   ((X SYMBOL))
   (IF (NOT (SIM (QUOTE (V. (OR. N. NIL) S.)) X))
    (BLOCK NIL (COMERR (QUOTE (BAD ORG))))
    (BLOCK NIL (SET FORG (CONS (QUOTE ORG)
       (IF (CADR X) (LIST (CADR X)) NIL))) (RETURN (CADDR X))))))
(PREC (INSTRUCTIONS ((IF . LISP) NOVALUE)
   NIL (BLOCK NIL (SET EXP (CDR EXP))
    (RETURN (IF (OR (NULL EXP) (NULL (CDR EXP)))
      (COMERR (QUOTE (BAD IF)))
      SCLASS (IFST EXP XGO)
      PCLASS (IFPRED EXP) TERGO (IFEXPT EXP) (IFEXP EXP)))))
  (FUNCTION (IFPRED SYMBOL)
   ((X SYMBOL))
   (BLOCK ((GEN SYMBOL) (TG SYMBOL) (FG SYMBOL))
    (SET VCLASS (QUOTE PREDICATE))
    (SET TG (IF TGO TGO (GENID)))
    (SET FG (IF FGO FGO (GENID)))
    A (IF (NULL X) (GO N) (NULL (CDR X)) (GO NO))
    (SET GEN NIL)
    (IF (EQN (CADR X) (QUOTE TRUE))
     (COMPACT (CAR X) TG NIL)
     (EQN (CADR X) (QUOTE FALSE))
     (COMPACT (CAR X) FG NIL)
     (BLOCK NIL (SET GEN (GENID))
      (COMPACT (CAR X) NIL GEN)
      (COMPACT (CADR X) TG FG) (ATTACHLAB GEN) G01573))
    (SET X (CDDR X))
    (GO A)
    NO (COMPACT (CAR X) TGO FGO)
    (GO R)
    N (IF GEN (CALERR))
    R (IF (NOT TGO) (ATTACHLAB TG) (NOT FGO) (ATTACHLAB FG))))
  (FUNCTION (IFST SYMBOL)
   ((EXP SYMBOL) (NAME SYMBOL))
   (BLOCK ((GEN SYMBOL) (XGO SYMBOL) (Z SYMBOL))
    (SET XGO NAME)
    A (IF (NULL EXP)
     (GO E)
     (NULL (CDR EXP))
     (GO L1)
     (AND (CDDR EXP) (NULL (CDDDR EXP)) (IFGO (CADDR EXP)))
     (GO L3) (IFGO (CADR EXP)) (GO B) (NULL XGO) (SET XGO (GENID)))
    (IF (NULL (CDDR EXP)) (GO L2))
    (SET GEN (GENID))
    (COMPRED (CAR EXP) NIL GEN)
    (COMSTAT (CADR EXP))
    (IF (NOT (LASTBRANCH)) (ATTACHGO XGO))
    (ATTACHLAB GEN)
    I (SET EXP (CDDR EXP))
    (GO A)
    L2 (COMPRED (CAR EXP) NIL XGO)
    (COMSTAT (CADR EXP))
    (GO E)
    L3 (COMPRED (CAR EXP)
     (IF (SET Z (IFGO (CADR EXP))) (CADADR EXP) NIL)
     (CADR (CADDR EXP)))
    (IF (NOT Z) (COMSTAT (CADR EXP)))
    (GO E)
    B (COMPRED (CAR EXP) (CADADR EXP) NIL)
    (GO I)
    L1 (COMSTAT (CAR EXP))
    E (IF (AND XGO (NOT NAME)) (ATTACHLAB XGO))))
(FUNCTION (IFEXPT SYMBOL)
 ((X SYMBOL))
 (BLOCK ((GEN SYMBOL))
    A (IF (NULL (CDR X)) (GO B))
    (SET GEN (GENID))
    (IF (NULL (CDDR X)) (GO C))
    (COMPRED (CAR X) NIL GEN)
    (COMTERM (CADR X))
    (SET X (CDR X))
    D (ATTACHGO TERGO)
    (ATTACHLAB GEN)
    I (SET X (CDR X))
    (GO A)
    C (COMPRED (CAR X) GEN NIL)
    (CALERR) (GO D) B (COMTERM (CAR X))))
(FUNCTION (IFEXP SYMBOL)
 ((EXP SYMBOL))
 (BLOCK ((TERMINS SYMBOL)
     (LABELS SYMBOL) (GOLIST SYMBOL) (TERGO SYMBOL) (X SYMBOL))
    (SET TERGO (GENID))
    (IFEXPT EXP)
    (ATTACHLAB TERGO)
    (COMTERMINS)
    (IF (SET X (COMGOES (QUOTE TRUE)))
     (COMER2 (QUOTE (UNDEFINED LABELS)) X))))
(FUNCTION (IFGO SYMBOL) ((X SYMBOL)) (SIM (QUOTE (GO ID.)) X))
(FUNCTION (COMPRED SYMBOL)
 ((X SYMBOL) (TG SYMBOL) (FG SYMBOL))
 (BLOCK ((SCLASS SYMBOL) (PCLASS SYMBOL) (XTYPE SYMBOL))
    (SET PCLASS (QUOTE TRUE)) (COMPACT X TG FG)))
(FUNCTION (COMPACT SYMBOL)
 ((X SYMBOL) (TGO SYMBOL) (FGO SYMBOL))
 (BLOCK ((XTYPE SYMBOL)
     (VCLASS SYMBOL)
     (VTYPE SYMBOL)
     (VREG SYMBOL)
     (VADDR SYMBOL) (VIND SYMBOL) (VINV SYMBOL) (VBYTE SYMBOL))
    (SET XTYPE (QUOTE BOOLEAN))
    (COMEXP X)
    (IF (EQN VCLASS (QUOTE PREDICATE))
     (GO P)
```

```
      (NULL VCLASS)
      (RETURN (COMERR (QUOTE (NOVALUE PREDICATE))))
      (EQN VCLASS (QUOTE DATUM))
      (IF (MEMBER VADDR (QUOTE (NIL FALSE))) (GO A) NIL)
      (MEMBER VTYPE (QUOTE (SYMBOL BOOLEAN))) (GO B))
     (BRANCHER (QUOTE ((TGO (BUC)))))
     (GO P)
     A (BRANCHER (QUOTE ((FGO (BUC)))))
     (GO P)
     B (MOVACTIVE (QUOTE SYMBOL) (QUOTE AC) NIL)
     (BRANCHER (QUOTE ((FGO (BOZP) (BNZP)))))
     P (IF (AND TGO FGO (NOT (LASTBRANCH)))
      (BRANCHER (QUOTE ((TGO (BUC)))))))))
   (FUNCTION (MAKEPRED SYMBOL)
    NIL (COMEXP (LIST (QUOTE IF) EXP (QUOTE TRUE) NIL)))
   (FUNCTION (COMBOOL SYMBOL)
    ((MODE SYMBOL))
    (BLOCK ((H SYMBOL) (G SYMBOL) (I SYMBOL))
     (IF (OR SCLASS (NOT PCLASS))
      (RETURN (MAKEPRED)) (NULL (CDR EXP)) (GO D))
     (SET H (CDR EXP))
     A (IF (NULL (CDR H)) (GO B))
     (SET I (IF MODE FGO TGO))
     (SET I (IF I I G G (SET G (GENID))))
     (COMPACT (CAR H) (IF MODE NIL I) (IF MODE I NIL))
     (SET H (CDR H))
     (GO A)
     B (COMPACT (CAR H) TGO FGO)
     (SET VCLASS (QUOTE PREDICATE))
     (IF (NULL G) (GO C))
     (ATTACHLAB G)
     C (RETURN NIL) D (SET VCLASS (QUOTE DATUM)) (SET VADDR MODE)))
   (FUNCTION (BRANCHER SYMBOL)
    ((BLIST SYMBOL))
    (BLOCK ((I SYMBOL)
      (B SYMBOL)
      (Z1 SYMBOL) (Z2 SYMBOL) (X SYMBOL) (Z SYMBOL) (DGO SYMBOL))
     A (IF (NULL BLIST) (GO R))
     (SET I (SET B (CAR BLIST)))
     (SET BLIST (CDR BLIST))
     B (IF (EQN (CAR B) (QUOTE TGO))
      (GO TG)
      (EQN (CAR B) (QUOTE FGO))
      (GO FG) (MEMBER (CADDR B) (QUOTE (TGO FGO))) (SET B (CDDR B)))
     (GO B)
     TG (SET Z1 TGO)
     (SET Z2 FGO)
     (GO C)
     FG (SET Z1 FGO)
     (SET Z2 TGO)
     C (SET Z (IF Z1 Z1 BLIST (IF (NULL DGO) (SET DGO (GENID)) DGO)
       (CDDR B) (BLOCK NIL (SET B (CDR B)) (RETURN Z2)) NIL))
     (IF (NOT Z) (GO A))
     (SET X (CONS (CAADR B) (LABELER Z) (CDADR B)))
     (ATTACH X)
     M (IF (AND BLIST (NOT Z1)) (GO A))
     (SET Z (IF (EQN (CAR X) (QUOTE BUC)) (QUOTE GO) (QUOTE ADDR)))
     (SET GOLIST (CONS (CONS Z LISTING) GOLIST))
     (GO A) R (IF DGO (ATTACH DGO))))
   (FUNCTION (COMREL SYMBOL)
    ((J SYMBOL))
    (BLOCK NIL (IF (OR SCLASS (NOT PCLASS)) (RETURN (MAKEPRED)))
     (COMVAL (CONS (QUOTE DIFFERENCE) (CDR EXP))
     (QUOTE NUMBER) NIL (QUOTE AC))                                     0082000
    (BRANCHER J) (SET VCLASS (QUOTE PREDICATE))))                       0082100
  (FUNCTION (NOTF SYMBOL)                                               0082200
   NIL (BLOCK ((J SYMBOL))                                              0082300
    (IF (OR SCLASS (NOT PCLASS)) (RETURN (MAKEPRED)))                   0082400
    (SET J TGO) (SET TGO FGO) (SET FGO J) (COMEXP (CADR EXP))))         0082500
  (FUNCTION (CALERR NOVALUE)                                            0082600
   NIL (BLOCK NIL (ATTACH (QUOTE (ARGS)))                               0082700
    (CALCOMP (QUOTE CONDERR))))                                         0082800
  (INSTRUCTIONS ((AND . LISP) NOVALUE) NIL (COMBOOL (QUOTE TRUE)))      0082900
  (INSTRUCTIONS ((OR . LISP) NOVALUE) NIL (COMBOOL NIL))                0083000
  (INSTRUCTIONS ((NULL . LISP) NOVALUE) NIL (NOTF))                     0083100
  (INSTRUCTIONS ((NOT . LISP) NOVALUE) NIL (NOTF))                      0083200
  (INSTRUCTIONS ((QUOTE . LISP) NOVALUE)                                0083300
   NIL (BLOCK NIL (IF (COMLCK 2) (RETURN NIL))                          0083400
    (SET VCLASS (QUOTE DATUM))                                          0083500
    (SET VTYPE (FVTYPE (SET VADDR (CADR EXP)))))))                      0083600
(BRANCH (INSTRUCTIONS ((GO . LISP) NOVALUE)                             0083700
   NIL (BLOCK ((A SYMBOL))                                              0083800
    (IF (OR (NOT SCLASS) (COMLCK 2))                                    0083900
     (RETURN (COMER2 EXP (QUOTE (ILLEGAL GO))))                         0084000
     (NOT (ATOM (CADR EXP))) (GO SW))                                   0084100
    (SET A (CADR EXP))                                                  0084200
    (GO C)                                                              0084300
    SW (SET A (CAADR EXP))                                              0084400
    (COMVAL (CADADR EXP) (QUOTE INTEGER) NIL (QUOTE AC))                0084500
    C (ATTACHGO A)))                                                    0084600
  (INSTRUCTIONS ((RETURN . LISP) NOVALUE)                               0084700
   NIL (IF (OR (NOT SCLASS) (COMLCK 2))                                 0084800
    (COMER2 EXP (QUOTE (ILLEGAL RETURN)))                               0084900
    (AND (LASTBRANCH) TERMINS)                                          0085000
    NIL (BLOCK ((SCLASS SYMBOL))                                        0085100
     (IF PCLASS (RETURN (COMPACT (CADR EXP) TGO FGO)))                  0085200
     (COMTERM (CADR EXP))                                               0085300
     (IF (NOT (LASTBRANCH)) (ATTACHGO TERGO)))))                        0085400
  (INSTRUCTIONS ((LABEL . LISP) NOVALUE)                                0085500
   NIL (BLOCK NIL (IF (COMLCK 3)                                        0085600
     (RETURN NIL)                                                       0085700
     (AND (IDP (CADR EXP)) (OR SCLASS PCLASS))                          0085800
     (ATTACHLAB (CADR EXP)) (COMER2 (CADR EXP) (QUOTE (BAD LABEL))))    0085900
    (COMEXP (CADDR EXP))))                                              0086000
  (FUNCTION (COMSWITCH SYMBOL)                                          0086100
   ((L SYMBOL) (V SYMBOL))                                              0086200
   (BLOCK NIL (REMOTE V)                                                0086300
    (REMOTE (LIST (QUOTE BUC) (LIST (QUOTE LABEL) V) (QUOTE A)))        0086400
    A (IF (NULL L) (GO B))                                              0086500
    (REMOTE (LIST (QUOTE BUC) (LABELER (CAR L))))                       0086600
    (SET GOLIST (CONS (CONS (QUOTE GO) REMOTES) GOLIST))                0086700
    (SET L (CDR L)) (GO A) B (SET LABELS (CONS V LABELS))))             0086800
  (FUNCTION (COMGOES SYMBOL)                                            0086900
   ((P SYMBOL))                                                         0087000
   (BLOCK ((L SYMBOL)                                                   0087100
     (M SYMBOL) (X SYMBOL) (Y SYMBOL) (G1 SYMBOL) (G2 SYMBOL))          0087200
    A (IF (NULL GOLIST)                                                 0087300
     (GO B)                                                             0087400
     (NOT (MEMBER (CADAR (GOGET (CAR GOLIST))) LABELS))                 0087500
     (SET L (CONS (CAR GOLIST) L)))                                     0087600
    (SET GOLIST (CDR GOLIST))                                           0087700
    (GO A)                                                              0087800
    B (IF (AND L (NOT (ATOM P)))                                        0087900
     (GO E) (OR (NULL P) (NOT (ATOM P))) (ATTACH (QUOTE (END))))        0088000
    (RETURN L)                                                          0088100
    E (SET X (LASTBRANCH))                                              0088200
    (SET G1 (LABELER (GENID)))
    (IF X (GO G))
    (SET G2 (LABELER (GENID)))
    (ATTACH (LIST (QUOTE BSX) G1 4 G2))
    G (ATTACH (CADR G1))
    (ATTACH (QUOTE (END)))
    (ATTACH (QUOTE (BUC 0 4)))
    (IF (NOT X) (ATTACH (CADR G2)))
    C (IF (NULL L) (RETURN GOLIST))
    (SET X (GOGET (CAR L)))
    (SET Y (LIST (QUOTE BSX) G1 4 (CAR X)))
    (IF (EQN (CAAR L) (QUOTE GO))
     (GO H) (SET G2 (GOMEMBER Y GOLIST)) (GO P))
    (SET G2 (LABELER (GENID)))
    (REMOTE (CADR G2))
    (REMOTE Y)
    (SET GOLIST (CONS (CONS (QUOTE DECR) REMOTES) GOLIST))
    P (SET (CAR X) G2)
    (GO Q)
    H (SET (CAR (CDAR L)) Y)
    (SET GOLIST (CONS (CONS (QUOTE DECR) (CDAR L)) GOLIST))
    Q (SET L (CDR L)) (GO C)))
  (FUNCTION (LASTBRANCH SYMBOL)
   NIL (AND (NOT (ATOM (CAR LISTING)))
    (MEMBER (CAAR LISTING) (QUOTE (BUC BSX BAX)))))
  (FUNCTION (GOGET SYMBOL)
   ((L SYMBOL))
   (BLOCK ((X SYMBOL) (Y SYMBOL))
    (SET X (CAR L))
    (SET Y (CDADR L))
    (RETURN (IF (EQN X (QUOTE GO))
      Y (EQN X (QUOTE ADDR)) Y (EQN X (QUOTE DECR)) (CDDR Y) NIL))))
  (FUNCTION (GOMEMBER SYMBOL)
   ((X SYMBOL) (L SYMBOL))
   (BLOCK ((P SYMBOL))
    A (IF (NULL L) (RETURN NIL))
    (SET P (CDAR L))
    (IF (NOT (EQ (CAR P) X)) (GO B) (ATOM (CADR P)) (GO R))
    (SET (CDR P) (CONS (GENID) (CDR P)))
    R (RETURN (LABELER (CADR P))) B (SET L (CDR L)) (GO A))))
(COMPARE (INSTRUCTIONS ((EQN . LISP) NOVALUE)
   NIL (IF (COMLCK 3)
    NIL (OR SCLASS (NOT PCLASS))
    (MAKEPRED)
    (BLOCK ((A SYMBOL) (B SYMBOL))
     (SET A (COMTOP (QUOTE SYMBOL) (CADR EXP)))
     (SET B (COMTOP (QUOTE SYMBOL) (CADDR EXP)))
     (IF (AND (EQN (GVCLAS A) (QUOTE DATUM)) (IDP (GVADDR A)))
      (EQBXE A B)
      (AND (EQN (GVCLAS B) (QUOTE DATUM)) (IDP (GVADDR B)))
      (EQBXE B A) (EQXOR (LIST A B NIL))))))
  (INSTRUCTIONS ((EQ . LISP) NOVALUE) NIL (EQHLP (QUOTE EQUAL.)))
  (INSTRUCTIONS ((EQUALN . LISP) NOVALUE)
   NIL (EQHLP (QUOTE EQUALN.)))
  (INSTRUCTIONS ((NQ . LISP) NOVALUE)
   NIL (COMEXP (LIST (QUOTE NOT) (CONS (QUOTE EQUAL) (CDR EXP)))))
  (INSTRUCTIONS ((LS . LISP) NOVALUE)
   NIL (COMREL (QUOTE ((FGO (BOZ)) (FGO (BOP) (BOM))))))
  (INSTRUCTIONS ((GR . LISP) NOVALUE)
   NIL (COMREL (QUOTE ((FGO (BOZ)) (FGO (BOM) (BOP))))))
  (INSTRUCTIONS ((LQ . LISP) NOVALUE)
   NIL (COMREL (QUOTE ((TGO (BOZ)) (FGO (BOP) (BOM))))))
  (INSTRUCTIONS ((GQ . LISP) NOVALUE)
 NIL (COMREL (QUOTE ((TGO (BOZ)) (FGO (BOM) (BOP))))))
(MACRO ((EQUAL . LISP) SYMBOL)
 ((X SYMBOL)) (CONS (QUOTE EQ) (CDR X)))
(FUNCTION (COMGLITCH SYMBOL)
 ((L SYMBOL))
 (BLOCK ((VCLASS SYMBOL)
   (VTYPE SYMBOL)
   (VREG SYMBOL)
   (VADDR SYMBOL)
   (LISTING SYMBOL)
   (VBYTE SYMBOL) (VBLOT SYMBOL) (VIND SYMBOL) (VINV SYMBOL))
  (RESTORE L)
  (IF (AND (EQN VCLASS (QUOTE DATUM)) (NOT (NUMBP VADDR)))
   (MOVACTIVE VTYPE (QUOTE AC) NIL))
  (SET VTYPE (QUOTE OCTAL)) (RETURN (CLUNK))))
(FUNCTION (EQXOR SYMBOL)
 ((L SYMBOL))
 (BLOCK ((INSTRUCTION SYMBOL))
  (SET INSTRUCTION (QUOTE XOR))
  (INHERIT (CAR (SET L (WRDHLP (LIST (COMGLITCH (CAR L))
       (COMGLITCH (CADR L)) NIL)))))
  (LSTLST (CADR L))
  (MOVACTIVE (QUOTE OCTAL) (QUOTE AC) NIL)
  (BRANCHER (QUOTE ((FGO (BNZP) (BOZP)))))
  (SET VCLASS (QUOTE PREDICATE))))
(FUNCTION (EQBXE SYMBOL)
 ((A SYMBOL) (B SYMBOL))
 (BLOCK NIL (RESTORE A)
  (RESTORE B)
  (MOVACTIVE (QUOTE SYMBOL) (QUOTE AC) NIL)
  (BRANCHER (SUBST (GVADDR A)
    (QUOTE Z) (QUOTE ((TGO (BXE AC (ID Z))) (FGO (BUC))))))
  (SET VCLASS (QUOTE PREDICATE))))
(FUNCTION (EQNIL SYMBOL)
 ((L SYMBOL))
 (BLOCK NIL (LSTLST (LAST (CAR L)))
  (LSTLST (LAST (CADR L)))
  (INHERIT (QUOTE (DATUM BOOLEAN NIL NIL NIL NIL NIL NIL)))))
(FUNCTION (EQSUB SYMBOL)
 ((L SYMBOL))
 (BLOCK ((X SYMBOL))
  (SET X (CDDDDR (CDDDR (IF (FULLW (GVBYTE (CAR L)))
      (CAR L)
      (FULLW (GVBYTE (CADR L)))
      (CADR L)
      (CAR (SET L (CONS (BLOCK ((VCLASS SYMBOL)
           (VTYPE SYMBOL)
           (VREG SYMBOL)
           (VADDR SYMBOL)
           (VBYTE SYMBOL)
           (VBLOT SYMBOL)
           (VIND SYMBOL) (VINV SYMBOL) (LISTING SYMBOL))
          (RESTORE (CAR L))
          (MOVACTIVE VTYPE (QUOTE AC) NIL) (RETURN (CLUNK)))
         (CDR L))))))))
  (SET (CAR X)
   (IF (MEMBER (QUOTE MINUS) (CAR X))
    (DELETEL (QUOTE MINUS) (CAR X)) (CONS (QUOTE MINUS) (CAR X))))
  (COMARI 0 L (SPLUS . SYS)
   PLIALG PLSMOV PLIMVP PLSPDL PLRALG PLSMOV PLRMVP PLSPDL)
  (IF VCLASS (BLOCK NIL (MOVACTIVE VTYPE (QUOTE AC) NIL)
    (BRANCHER (QUOTE ((FGO (BNZ) (BOZ)))))
    (SET VCLASS (QUOTE PREDICATE)) G01574)
```

```
     (INHERIT (QUOTE (DATUM BOOLEAN NIL TRUE NIL NIL NIL NIL))))))
  (FUNCTION (EQHLP SYMBOL)
   ((FCN SYMBOL))
   (BLOCK ((TA SYMBOL) (TB SYMBOL) (A SYMBOL) (B SYMBOL) (X SYMBOL))
    (IF (COMLCK 3)
     (RETURN NIL) (OR SCLASS (NOT PCLASS)) (RETURN (MAKEPRED)))
    (SET X (COMARGS))
    (SET TA (GVTYPE (SET A (CAR X))))
    (SET TB (GVTYPE (SET B (CADR X))))
    (IF (OR (AND (EQN TA TB)
       (MEMBER TA (QUOTE (OCTAL FUNCTIONAL BOOLEAN))))
      (AND (EQN TA (QUOTE BOOLEAN)) (EQN TB (QUOTE SYMBOL)))
      (AND (EQN TA (QUOTE SYMBOL)) (EQN TB (QUOTE BOOLEAN))))
     (RETURN (EQXOR X))
     (AND (MEMBER TA (QUOTE (OCTAL INTEGER REAL)))
      (MEMBER TB (QUOTE (OCTAL INTEGER REAL))))
     (RETURN (IF (AND (NOT (EQN TA TB)) (EQN FCN (QUOTE EQUALN.)))
       (EQNIL X) (EQSUB X)))
     (NOT (OR (EQN (QUOTE SYMBOL) TA) (EQN (QUOTE SYMBOL) TB)))
     (RETURN (EQNIL X))
     (AND (EQN TB (QUOTE SYMBOL))
      (OR (NOT (EQN TA (QUOTE SYMBOL)))
       (NOT (EQN (GVCLAS A) (QUOTE DATUM))) (NOT (IDP (GVADDR A)))))
     (BLOCK NIL (SET A B)
      (SET TA TB) (SET TB (GVTYPE (SET B (CAR X)))) G01575))
    (IF (AND (IDP (GVADDR A)) (EQN (GVCLASS A) (QUOTE DATUM)))
     (RETURN (EQBXE A B)))
    (ATTACH (QUOTE (ARGS)))
    (RESTORE A)
    (MOVPDS (QUOTE SYMBOL) NIL)
    (RESTORE B)
    (MOVACTIVE (QUOTE SYMBOL) (QUOTE AC) NIL)
    (CALCOMP FCN) (SET VTYPE (QUOTE BOOLEAN)))))
(BLOCK (INSTRUCTIONS ((CODE . LISP) NOVALUE)
   NIL (BLOCK NIL (BLOTTO)
    (INHERIT (QUOTE (ACTIVE OCTAL AC NIL NIL NIL NIL NIL)))
    (LSTLST (REVERSE (CDR EXP)))))
  (INSTRUCTIONS ((BLOCK . LISP) NOVALUE)
   NIL (BLOCK ((P SYMBOL) (X SYMBOL))
    (IF (AND SCLASS (NOT (ANYVARS (CADR EXP))))
     (RETURN (BLOCK NIL (MAPCAR (CADR EXP)
        (FUNARG SYMBOL ((J SYMBOL)) (COMSWITCH (CDDR J) (CAR J))))
       (RETURN (COMBLOCK (CDDR EXP))))))
    (SET X ALIST)
    (BLOCK ((ALIST SYMBOL) (LABELS SYMBOL) (GOLIST SYMBOL))
     (SET ALIST X)
     (SET P (IF (NULL (CADR EXP)) (QUOTE TRUE) (BKBIND)))
     (IF SCLASS (COMBLOCK (CDDR EXP))
      PCLASS (BLOCK ((X SYMBOL))
       (IF (NULL TGO) (SET X (SET TGO (GENID))))
       (IF (NULL FGO) (SET X (SET FGO (IF X X (GENID)))))
       (COMBLOCK (CDDR EXP))
       (SET VCLASS (QUOTE PREDICATE))
       (IF (NULL X) (GO G01578))
       (ATTACHLAB X) (SET TGO NIL) G01578)
      TERGO (COMBLOCK (CDDR EXP))
      (BLOCK ((TERMINS SYMBOL) (TERGO SYMBOL))
       (SET TERGO (GENID))
       (COMBLOCK (CDDR EXP)) (ATTACHLAB TERGO) (COMTERMINS) G01579))
     (SET X (COMGOES P)) G01577)
    (IF (OR SCLASS PCLASS TERGO)
     (SET GOLIST (NCONC X GOLIST))
     X (COMER2 (QUOTE (UNDEFINED LABELS))
      (MAPCAR X (FUNARG SYMBOL ((J SYMBOL)) (CADAR (GOGET J))))))))
  (FUNCTION (COMBLOCK SYMBOL)
   ((L SYMBOL))
   (BLOCK ((X SYMBOL) (XGO SYMBOL))
    A (IF (NULL L)
     (GO B)
     (NOT (ATOM (SET X (CAR L))))
     (BLOCK ((XGO SYMBOL (IF (SIM (QUOTE (S. (GO ID.) . S.)) L)
         (CADADR L) NIL))) (COMSTAT X))
     (IDP X)
     (ATTACHLAB X) (NULL X) NIL (COMER2 X (QUOTE (NOT A LABEL))))
    (SET L (CDR L))
    (GO A)
    B (IF (AND (NOT SCLASS) (NOT (LASTBRANCH)))
     (COMSTAT (LIST (QUOTE RETURN) (ITYPE XTYPE))))))
  (MACRO ((ORDER . LISP) SYMBOL)
   ((EXP SYMBOL))
   (IF (COMLCK 2)
    NIL (ATOM (CADR EXP))
    (CADR EXP)
    (BLOCK ((X SYMBOL) (Y SYMBOL))
     (SET X (MAPCAR (CDADR EXP)
       (FUNARG SYMBOL ((J SYMBOL))
        (LIST (GENID) (QUOTE ASSIGNED) J))))
     (SET Y (CONS (CAADR EXP)
       (MAPCAR X (FUNARG SYMBOL ((J SYMBOL)) (CAR J)))))
     (RETURN (LIST (QUOTE BLOCK)
       (IF SCLASS Y (LIST (QUOTE RETURN) Y))))))))
(CASE (MACRO ((CASE . LISP) SYMBOL)
   ((X SYMBOL))
   (IF (LS (LENGTH X) 3)
    (BLOCK NIL (COMERR (QUOTE (BAD CASE))) (RETURN (ITYPE XTYPE)))
    (BLOCK (LABELS)
     (RETURN (BLOCK ((M (MAPCAR (CDDR X)
          (FUNARG SYMBOL ((J SYMBOL))
           (IF (SIM (QUOTE (GO ID.)) J)
            (BLOCK NIL (SET LABELS (CONS (CADR J) LABELS)))
            (BLOCK NIL (SET LABELS (CONS (GENID) LABELS))
             (RETURN (LIST (QUOTE BLOCK)
               NIL (CAR LABELS)
               (IF SCLASS J (LIST (QUOTE RETURN) J))))))))))
      (RETURN (APPEND (QUOTE (BLOCK NIL))
        (CONS (CONS (QUOTE CASEGO)
          (CADR X) (REVERSE LABELS)) M))))))))
  (INSTRUCTIONS ((CASEGO . LISP) NOVALUE)
   NIL (BLOCK ((L (LIST (LIST (QUOTE LABEL) (GENID)) -1)))
    (COMVAL (CADR EXP) (QUOTE INTEGER) NIL (QUOTE AC))
    (LSTLST (SUBST L (QUOTE L)
      (SUBST (LENGTH (CDDR EXP))
       (QUOTE N)
       (QUOTE ((BUC L A)
         (BOP L) (SUB N (L567.7 R S)) (BOZ L) (BOM L))))))
    (MAP (CDDR EXP)
     (FUNARG SYMBOL ((J SYMBOL))
      (BLOCK NIL (ATTACH (LIST (QUOTE BUC) (LABELER (CAR J))))
       (IF (NOT (MEMBER (CAR J) LABELS))
        (SET GOLIST (CONS (CONS (QUOTE GO) LISTING) GOLIST))))))
    (ATTACH (CADAR L)))))
(ZAPZAP (MACRO ((LIST . LISP) SYMBOL)
   ((EXP SYMBOL))
   (IF (GR (LENGTH EXP) 2)
    (LISTX (CDR EXP))
    (CDR EXP) (CONS (QUOTE (LIST1 . SYS)) (CDR EXP)) NIL))
```

```
(MACRO ((CONS . LISP) SYMBOL)
 ((EXP SYMBOL))
 (IF (LS (LENGTH EXP) 3)
  (BLOCK NIL (COMLCK 3)) (LISTX (CDR EXP))))
(INSTRUCTIONS ((SHIFT . LISP) NOVALUE) NIL (SHIFTER (QUOTE CYC)))
(INSTRUCTIONS ((SCALE . LISP) NOVALUE) NIL (SHIFTER (QUOTE SFA)))
(INSTRUCTIONS ((CYCLE . LISP) NOVALUE) NIL (SHIFTER (QUOTE CYA)))
(FUNCTION (LISTX SYMBOL)
 ((X SYMBOL))
 (BLOCK ((L SYMBOL))
  (SET L (LENGTH X))
  (RETURN (IF (GR L 4)
    (LIST (QUOTE (CONS4 . SYS))
     (CAR X) (CADR X) (CADDR X) (LISTX (CDDDR X)))
    (CONS (CDR (FIND L (CDR (FIND (CAR EXP)
         (QUOTE ((LIST (2 LIST2 . SYS)
            (3 LIST3 . SYS) (4 LIST4 . SYS))
           (CONS (2 CONS2 . SYS)
            (3 CONS3 . SYS) (4 CONS4 . SYS)))))))) X)))))
(FUNCTION (SHIFTER SYMBOL)
 ((C SYMBOL))
 (IF (COMLCK 3)
  NIL (BLOCK ((X SYMBOL))
   (COMEXP1 (CADDR EXP))
   (SET VINV (IF (MEMBER (QUOTE MINUS) VINV) NIL (QUOTE (MINUS))))
   (IF (EQN VCLASS (QUOTE DATUM))
    (BLOCK NIL (VSET (COMDAT (CLUNK)))
     (SET X (CNVDATM VTYPE VADDR (QUOTE INTEGER))) G01583)
    (MOVPDS (QUOTE INTEGER) NIL))
   (COMTYP (QUOTE OCTAL) (CADR EXP))
   (MOVACTIVE (QUOTE OCTAL) (QUOTE AC) NIL)
   (IF (EQN C (QUOTE CYC)) (ATTACH (QUOTE (STZ B.))))
   (ATTACH (CONS C (IF X (LIST X (QUOTE R)) (QUOTE (POP.)))))))))
(FUNARG (MACRO ((FUNCTION . LISP) SYMBOL) ((X SYMBOL)) (COMPILER X))
  (INSTRUCTIONS ((FUNARG . LISP) NOVALUE)
   NIL (IF (NOT (AND (SIM (QUOTE (V. A. (OR. L. NIL) S.)) EXP)
      (OR (VTYPEP (CADR EXP)) (NOT (CADR EXP)))))
    (COMER2 EXP (QUOTE (NOT LEGAL FUNARG)))
    (OR SCLASS PCLASS)
    (BLOCK NIL (SET VCLASS (QUOTE DATUM))
     (SET VTYPE (QUOTE BOOLEAN)) (SET VADDR (QUOTE TRUE)))
    (BLOCK ((APLIST SYMBOL (APPEND ALIST APLIST))
      (FEXP SYMBOL (LIST (QUOTE FUNCTION)
        (LIST (GENID) (IF (CADR EXP) (CADR EXP) STYPE))
        (CADDR EXP) (CADDDR EXP))))
     (RETURN (BLOCK ((IR SYMBOL) (C SYMBOL))
       (BLOCK ((IRLIST SYMBOL))
        (SET C (FUNCTIO FEXP)) (SET IR IRLIST) G01584)
       (IF ERRFLG (RETURN NIL)
        (NOT IR) (RETURN (COMEXP (COMPILER FEXP))))
       (BLOCK ((IRLIST SYMBOL))
        (ATTACH (LIST (QUOTE LDA)
          (FUNCTIO (BLOCK NIL (SET (CAR (CDDDR FEXP))
             (LIST (QUOTE BLOCK)
              NIL (QUOTE (CODE (LDX (FMCALL . SYS) L 7)))
              (LIST (QUOTE BLOCK)
               (MAPCAR IR (FUNCTION (G01586 SYMBOL)
                 ((J SYMBOL))
                 (LIST (CAR J) (CADR J) (QUOTE LOC) (CAR J))))
               (LIST (QUOTE RETURN) (CADDDR FEXP)))))
            (RETURN FEXP))) (QUOTE (2Q1 R L4567.7))))) G01585)
       (ATTACH (QUOTE (STX A. L 8)))
       (INHERIT (QUOTE (ACTIVE FUNCTIONAL AC NIL NIL NIL (AC) NIL)))
       L (IF (NULL IR)
        (RETURN NIL)
        (NOT (OR (MEMBER (SET C (CAR IR)) ALIST) (MEMBER C IRLIST)))
        (SET IRLIST (CONS C IRLIST))) (SET IR (CDR IR)) (GO L)))))))
(TRY (MACRO ((TRY . LISP) SYMBOL)
   ((EXP SYMBOL))
   (IF (COMLCK 4)
    NIL (BLOCK ((G1 SYMBOL) (G2 SYMBOL))
     (RETURN (APPEND (SUBST (SET G1 (GENID))
        (QUOTE X)
        (QUOTE (BLOCK (((TRYPT . SYS)
            OCTAL FLUID (CODE (LDA ((LABEL X) (MINUS ORG.))
              (L567.7 R))))))))
       (LIST (CADDDR EXP)
        (LIST (QUOTE GO) (SET G2 (GENID)))
        G1 (CONS (QUOTE SET)
         (LIST (CADR EXP) (QUOTE (C2S. (CODE)))))
        (LIST (QUOTE GO) (CADDR EXP)) G2)))))))
(RELATE (MACRO ((RELATION . LISP) SYMBOL)
   ((EXP SYMBOL))
   (BLOCK ((L SYMBOL (LENGTH EXP)))
    (RETURN (IF (OR (LS L 4) (NOT (EQ (REMAINDER L 2) 0)))
      (COMLCK 0) (LIST (QUOTE AND) (REL. (CADR EXP) (CDDR EXP)))))))
  (FUNCTION (REL. SYMBOL)
   ((A SYMBOL) (R SYMBOL))
   (IF (NULL R)
    (QUOTE TRUE)
    (AND (OKREL. (CADR R)) (OR (NULL (CDDR R)) (OKREL. (CADDDR R))))
    (RELCON A (CADR R) R)
    (BLOCK ((G SYMBOL (GENID)))
     (RETURN (LIST (QUOTE BLOCK)
       (LIST (LIST G (QUOTE ASSIGNED) (CADR R)))
       (LIST (QUOTE RETURN) (RELCON A G R)))))))
  (FUNCTION (RELCON SYMBOL)
   ((A SYMBOL) (B SYMBOL) (C SYMBOL))
   (LIST (QUOTE AND) (LIST (CAR C) A B) (REL. B (CDDR C))))
  (FUNCTION (OKREL. SYMBOL)
   ((X SYMBOL)) (SIM (QUOTE (OR. A. V. (QUOTE S.))) X)))
****END OF FILE DETECTED
```

```
(SECTION (SECTION (COMPIL SUPV SYS LISP) SYMBOL))
(VHELP (DECLARE (INTLST SYMBOL FLUID)
   (REALST SYMBOL FLUID)
   (SYMLST SYMBOL FLUID) (LOCLST SYMBOL FLUID))
  (FUNCTION (GVCLASS SYMBOL) ((X SYMBOL)) (GVCLAS X))
  (FUNCTION (GVCLAS SYMBOL) ((X SYMBOL)) (CAR X))
  (FUNCTION (GVTYPE SYMBOL) ((X SYMBOL)) (CADR X))
  (FUNCTION (GVREG SYMBOL) ((X SYMBOL)) (CADDR X))
  (FUNCTION (GVADDR SYMBOL) ((X SYMBOL)) (CADDDR X))
  (FUNCTION (GVIND SYMBOL) ((X SYMBOL)) (CAR (CDDDDR X)))
  (FUNCTION (GVBYTE SYMBOL) ((X SYMBOL)) (CADR (CDDDDR X)))
  (FUNCTION (GVBLOT SYMBOL) ((X SYMBOL)) (CADDR (CDDDDR X)))
  (FUNCTION (GVINV SYMBOL) ((X SYMBOL)) (CADDDR (CDDDDR X)))
  (FUNCTION (VLIST SYMBOL)
   NIL (LIST VCLASS VTYPE VREG VADDR VIND VBYTE VBLOT VINV))
  (FUNCTION (VSET SYMBOL)
   ((X SYMBOL))
   (BLOCK NIL (SET VCLASS (GVCLAS X))
    (SET VTYPE (GVTYPE X))
    (SET VREG (GVREG X))
    (SET VADDR (GVADDR X))
    (SET VIND (GVIND X))
    (SET VBYTE (GVBYTE X))
    (SET VBLOT (GVBLOT X)) (SET VINV (GVINV X))))
  (FUNCTION (INHERIT SYMBOL)
   ((B SYMBOL))
   (BLOCK ((A SYMBOL))
    (SET A (UNION VBLOT (GVBLOT B))) (VSET B) (SET VBLOT A)))
  (FUNCTION (LSTLST SYMBOL)
   ((LST SYMBOL)) (IF LST (SET LISTING (NCONC LST LISTING)) NIL))
  (FUNCTION (CLUNK SYMBOL) NIL (NCONC (VLIST) (LIST LISTING)))
  (FUNCTION (RESTORE SYMBOL)
   ((X SYMBOL)) (BLOCK NIL (INHERIT X) (RETURN (LSTLST (LAST X))))))
(COMPILER (FUNCTION (COMTOP SYMBOL)
   ((X SYMBOL) (E SYMBOL))
   (BLOCK ((VCLASS SYMBOL)
     (VTYPE SYMBOL)
     (VREG SYMBOL)
     (VADDR SYMBOL)
     (VIND SYMBOL)
     (VBYTE SYMBOL) (VBLOT SYMBOL) (VINV SYMBOL) (LISTING SYMBOL))
    (COMTYP X E) (RETURN (CLUNK))))
  (FUNCTION (CALCOMP SYMBOL)
   ((DV SYMBOL))
   (BLOCK NIL (BLOTTO)
    (ATTACH (LIST (QUOTE CALL)
      (CONS DV (QUOTE SYS)) (GETDEC DV (QUOTE SYS))))))
  (FUNCTION (COMARGS SYMBOL)
   NIL (BLOCK ((L SYMBOL) (M SYMBOL))
    (SET L EXP)
    (SET M (LIST NIL))
    TAG (IF (NULL (SET L (CDR L))) (RETURN M))
    (BLOCK ((VCLASS SYMBOL)
      (VTYPE SYMBOL)
      (VREG SYMBOL)
      (VADDR SYMBOL)
      (VIND SYMBOL)
      (VBYTE SYMBOL) (VBLOT SYMBOL) (VINV SYMBOL) (LISTING SYMBOL))
     (COMEXP1 (CAR L)) (SET M (CONS (CLUNK) M)) G01595) (GO TAG)))
  (FUNCTION (COMTYP SYMBOL)
   ((XTYPE SYMBOL) (EXP SYMBOL))
   (BLOCK ((SCLASS SYMBOL) (PCLASS SYMBOL) (TERGO SYMBOL))
    (COMEXP EXP)
  (IF (EQN VCLASS (QUOTE DATUM))
   (GO DAT)
   (EQN XTYPE VTYPE)
   (RETURN NIL)
   (AND (EQN VTYPE (QUOTE OCTAL))
    (EQN XTYPE (QUOTE INTEGER)) (NOT VBYTE)) (GO OCTINT))
  XYZ (MOVACTIVE XTYPE (QUOTE AC) NIL)
  (RETURN NIL)
  OCTINT (SET VTYPE (QUOTE INTEGER))
  (RETURN NIL)
  DAT (INHERIT (COMDAT (NCONC (VLIST) (QUOTE (NIL)))))
  (SET VADDR (CNVDATM VTYPE VADDR XTYPE)) (SET VTYPE XTYPE)))
(FUNCTION (COMDAT SYMBOL)
 ((FORM SYMBOL))
 (BLOCK ((VCLASS SYMBOL)
   (VTYPE SYMBOL)
   (VBYTE SYMBOL)
   (VIND SYMBOL)
   (VBLOT SYMBOL)
   (VREG SYMBOL) (VINV SYMBOL) (VADDR SYMBOL) (LISTING SYMBOL))
  (RESTORE FORM)
  (IF (MEMBER (QUOTE MINUS) VINV) (SET VADDR (MINUS VADDR)))
  (IF (MEMBER (QUOTE RECIP) VINV)
   (BLOCK NIL (SET VADDR (IQUOTIENT 1.0 VADDR))
    (SET VTYPE (QUOTE REAL)) G01596))
  (SET VINV NIL) (RETURN (CLUNK))))
(FUNCTION (COMEXP1 SYMBOL)
 ((EXP SYMBOL))
 (BLOCK ((PCLASS SYMBOL)
   (SCLASS SYMBOL) (TERGO SYMBOL) (XTYPE SYMBOL))
  (RETURN (COMEXP EXP))))
(FUNCTION (COMLCK SYMBOL)
 ((NUM SYMBOL))
 (BLOCK NIL (IF (EQ (LENGTH EXP) NUM) (RETURN NIL))
  (COMER2 (CAR EXP) (QUOTE (WRONG NUM OF ARGS)))
  (INHERIT (QUOTE (DATUM INTEGER NIL 0 NIL NIL NIL NIL NIL)))
  (RETURN TRUE)))
(FUNCTION (VINDX SYMBOL)
 NIL (AND (EQN VCLASS (QUOTE LOC))
  (NOT VIND)
  (NOT VINV)
  (OR (NULL VBYTE)
   (EQN VBYTE (QUOTE RH))
   (AND (NOT (ATOM VBYTE))
    (EQ (CAR VBYTE) 0) (GR (CADR VBYTE) 23))))))
(FUNCTION (COMCAR SYMBOL)
 ((B SYMBOL))
 (BLOCK NIL (IF (COMLCK 2) (RETURN NIL))
  (COMEXP1 (CADR EXP))
  (IF (EQN VTYPE (QUOTE SYMBOL)) (GO A))
  (COMER2 (CADR EXP) (QUOTE (NOT SYMBOL)))
  A (IF (VINDX) (GO LOC))
  (MOVACTIVE (QUOTE SYMBOL) (QUOTE AC) NIL)
  (SET VCLASS (QUOTE LOC))
  (SET VADDR 0)
  (SET VBYTE B) (RETURN NIL) LOC (SET VIND TRUE) (SET VBYTE B))))
PTIMUM (FUNCTION (COMOPT SYMBOL)
 ((FRMLST SYMBOL)
  (ALGFCN (FUNCTIONAL SYMBOL))
  (MOVE (FUNCTIONAL SYMBOL SYMBOL))
  (MOVEP (FUNCTIONAL SYMBOL SYMBOL)) (MOVPOL (FUNCTIONAL SYMBOL)))
 (BLOCK ((LISTING SYMBOL)
   (VCLASS SYMBOL)
   (VTYPE SYMBOL)
   (VREG SYMBOL)
   (VADDR SYMBOL)
   (VIND SYMBOL)
   (VBYTE SYMBOL)
   (VBLOT SYMBOL)
   (VINV SYMBOL)
   (FORM SYMBOL) (LOCLST SYMBOL) (ACTLST SYMBOL) (TEM SYMBOL))
  (SET ACTLST (LIST NIL))
  (IF (NOT (SET FORM (CAR FRMLST))) (RETURN NIL))
  A1 (IF (EQN (SET TEM (GVCLAS FORM)) (QUOTE ACTIVE))
   (GO B1)
   (EQN TEM (QUOTE DATUM))
   (GO C1) (AND (NOT (GVREG FORM)) (MOVEP FORM)) (GO C2))
  (SET FORM (MOVE FORM))
  B1 (SET ACTLST (CONS FORM ACTLST))
  A2 (IF (SET FORM (CAR (SET FRMLST (CDR FRMLST))))
   (GO A1)
   (SET FORM (CAR ACTLST)) (GO D1) (CDR LOCLST) (GO E1))
  RET (RETURN (LIST (VLIST) LISTING))
  C1 (SET FORM (COMDAT FORM))
  C2 (RESTORE FORM)
  (SET (CAR (CDDDDR (CDDDDR FORM))) NIL)
  (SET LOCLST (CONS FORM LOCLST))
  (GO A2)
  E1 (INHERIT (QUOTE (NIL NIL NIL NIL NIL NIL NIL NIL)))
  (ALGFCN)
  (GO RET)
  D1 (RESTORE FORM)
  (IF LOCLST (ALGFCN))
  (IF (NOT (SET FORM (CAR (SET ACTLST (CDR ACTLST))))) (GO RET))
  (SET LOCLST (LIST (MOVPDL))) (GO D1)))
(FUNCTION (COMARI SYMBOL)
 ((DATUM SYMBOL)
  (LST SYMBOL)
  (FCN (FUNCTIONAL SYMBOL SYMBOL SYMBOL))
  (IALGFCN SYMBOL)
  (IMOVE SYMBOL)
  (IMOVEP SYMBOL)
  (IMOVPDL SYMBOL)
  (RALGFCN SYMBOL)
  (RMOVE SYMBOL) (RMOVEP SYMBOL) (RMOVPDL SYMBOL))
 (BLOCK ((INTLST SYMBOL)
   (REALST SYMBOL)
   (SYMLST SYMBOL)
   (DATA SYMBOL) (TEM SYMBOL) (TYPE SYMBOL) (FORM SYMBOL))
  (IF (GR (SET TEM (LENGTH LST)) 2)
   (GO MANY) (NOT (EQUALN TEM 1)) (GO ARG1))
  (COMERR (QUOTE (0 ARG TO ARITH)))
  NRET (RETURN (LIST NIL))
  ARG1 (IF (EQN (GVCLAS (CAR LST)) (QUOTE DATUM))
   (SET (CAR LST) (COMDAT (CAR LST))))
  (RESTORE (CAR LST))
  (GO NRET)
  MANY (SET DATA DATUM)
  (SET INTLST (SET REALST (SET SYMLST (LIST NIL))))
  PARC (SET TYPE (GVTYPE (SET FORM (CAR LST))))
  (IF (EQN (GVCLAS FORM) (QUOTE DATUM))
   (BLOCK NIL (IF (NUMBP (GVADDR FORM)) (GO XYZ))
    (COMER2 (GVADDR FORM) (QUOTE (IS NON NUM DATA IN ARITH)))
    (GO G01597)
    XYZ (RESTORE (SET FORM (COMDAT FORM)))
    (SET DATA (FCN DATA (GVADDR FORM))) G01597)
```

```
     (EQN TYPE (QUOTE SYMBOL))
     (SET SYMLST (CONS FORM SYMLST))
     (OR (EQN TYPE (QUOTE REAL)) (MEMBER (QUOTE RECIP) VINV))
     (SET REALST (CONS FORM REALST))
     (MEMBER TYPE (QUOTE (INTEGER OCTAL)))
     (SET INTLST (CONS FORM INTLST))
     (COMER2 TYPE (QUOTE (TYPE ARG TO ARITH))))
    (IF (CAR (SET LST (CDR LST)))
     (GO PARC)
     (EQ DATA DATUM)
     (GO ND)
     (FIXP DATA)
     (SET INTLST (CONS (LIST (QUOTE DATUM)
        (QUOTE INTEGER) NIL DATA NIL NIL VBLOT NIL NIL) INTLST))
     (SET REALST (CONS (LIST (QUOTE DATUM)
        (QUOTE REAL) NIL DATA NIL NIL VBLOT NIL NIL) REALST)))
    ND (SET INTLST (COMOPT INTLST IALGFCN IMOVE IMOVEP IMOVPDL))
    (IF (NOT (CAR REALST))
     (IF INTLST (GO INTUP) (GO NOUP))
     INTLST (BLOCK ((VCLASS SYMBOL)
       (VTYPE SYMBOL)
       (VADDR SYMBOL)
       (VREG SYMBOL)
       (VIND SYMBOL)
       (VBYTE SYMBOL) (VBLOT SYMBOL) (VINV SYMBOL) (LISTING SYMBOL))
      (VSET (CAR INTLST))
      (SET LISTING (CADR INTLST))
      (MOVACTIVE (QUOTE REAL) (QUOTE AC) NIL)
      (SET REALST (CONS (CLUNK) REALST)) G01598))
    (SET REALST (COMOPT REALST RALGFCN RMOVE RMOVEP RMOVPDL))
    REALUP (LSTLST (CADR REALST))
    (INHERIT (CAR REALST))
    (RETURN SYMLST)
    INTUP (SET REALST INTLST)
    (GO REALUP)
    NOUP (INHERIT (QUOTE (NIL NIL NIL NIL NIL NIL NIL NIL)))
    (RETURN SYMLST))))
(COMFLP (INSTRUCTIONS ((FLOAT . LISP) NOVALUE)
   NIL (BLOCK NIL (IF (COMLCK 2) (RETURN NIL))
    (COMTYP (QUOTE INTEGER) (CADR EXP))
    (MOVACTIVE (QUOTE REAL) (QUOTE AC) NIL)))
  (INSTRUCTIONS ((PROP . LISP) NOVALUE) NIL (COMCAR (QUOTE (  18))))
  (INSTRUCTIONS ((CAR . LISP) NOVALUE)
   NIL (COMCADR (QUOTE (24 18))))
  (INSTRUCTIONS ((CDR . LISP) NOVALUE) NIL (COMCADR (QUOTE (  24))))
  (FUNCTION ((COMCADR . COMPIL) SYMBOL)
   ((X SYMBOL))
   (BLOCK ((EXP FLUID (IF (DEBUGGING)
       (LIST (CAR EXP)
        (CONS (QUOTE (ATMCHK . DEBUG)) (CDR EXP))) EXP)))
    (RETURN (COMCAR X))))
  (INSTRUCTIONS ((BIT . LISP) NOVALUE)
   NIL (BLOCK ((NB SYMBOL) (EL SYMBOL))
    (IF (COMLCK 4)
     (RETURN NIL)
     (AND (NUMBP (CADR EXP)) (NUMBP (CADDR EXP))) (GO CK))
    (RETURN (COMEXP (CONS (QUOTE (BITS . SYS)) (CDR EXP))))
    CK (COMTYP (QUOTE OCTAL) (CADDDR EXP))
    (IF VINV (MOVACTIVE VTYPE (QUOTE AC) NIL))
    (SET VBYTE (WHATBITS VBYTE))
    (IF (AND (LS (SET NB (PLUS (CADR EXP) (CAR VBYTE))) 48)
      (LS (PLUS NB (SET EL (CADDR EXP))) 49)
      (LS EL (PLUS 1 (CADR VBYTE)))) (GO FIN))
     (MOVACTIVE VTYPE (QUOTE AC) NIL)
     (SET VBYTE (LIST (CADR EXP) (CADDR EXP)))
     (RETURN NIL) FIN (SET VBYTE (LIST NB EL)))))
 (TIMER (MACRO ((RECIP . LISP) SYMBOL)
    ((EXP SYMBOL))
    (IF (COMLCK 2) NIL (LIST (QUOTE QUOTIENT) 1.0 (CADR EXP))))
   (INSTRUCTIONS ((IQUOTIENT . LISP) NOVALUE)
    NIL (BLOCK NIL (DIVIDE. (QUOTE INTEGER) (QUOTE LDA))
     (ATTACH (QUOTE (MUL 1 (L567.7 R S))))
     (ATTACH1 (CONS (QUOTE DVD)
      (MOVARG (QUOTE INTEGER) (QUOTE AC) NIL (QUOTE LDA))))
     (INHERIT (QUOTE (ACTIVE INTEGER AC NIL NIL NIL (AC B) NIL)))))
   (INSTRUCTIONS ((QUOTIENT . LISP) NOVALUE)
    NIL (BLOCK NIL (DIVIDE. (QUOTE REAL) (QUOTE FAD))
     (ATTACH1 (CONS (QUOTE FDV)
      (MOVARG (QUOTE REAL) (QUOTE AC) NIL (QUOTE FAC))))
     (INHERIT (QUOTE (ACTIVE REAL AC NIL NIL NIL (AC B) NIL)))))
   (INSTRUCTIONS ((TIMES . LISP) NOVALUE)
    NIL (BLOCK ((TYPE SYMBOL)
      (PARITY SYMBOL) (FORM SYMBOL) (SYM SYMBOL))
     (SET SYM (COMARI 1 (COMARGS)
      (STIMS . SYS)
      MPIALG PLSMOV PLIMVP PLSPDL MPRALG PLSMOV PLRMVP PLSPDL))
     (IF VCLASS (IF (SET FORM (CAR SYM)) (GO TIM) (RETURN NIL))
      (CAR SYM) (GO INT))
     (INHERIT (QUOTE (DATUM INTEGER NIL 1 NIL NIL NIL NIL)))
     (RETURN NIL)
     INT (RESTORE (CAR SYM))
     LOOP (IF (NOT (SET FORM (CAR (SET SYM (CDR SYM)))))
      (RETURN NIL))
     TIM (ATTACH (QUOTE (ARGS)))
     (SET TYPE VTYPE)
     (IF (EQN VTYPE (QUOTE SYMBOL)) (GO ST))
     (MOVPDS VTYPE NIL)
     (INHERIT FORM)
     (SET PARITY VINV)
     AT (SET VINV NIL)
     (LSTLST (LAST FORM))
     (MOVACTIVE (QUOTE SYMBOL) (QUOTE AC) NIL)
     (CALCOMP (IF (EQN TYPE (QUOTE SYMBOL))
      (QUOTE STIMS)
      (EQN TYPE (QUOTE REAL)) (QUOTE STIMR) (QUOTE STIMI)))
     (INHERIT (QUOTE (ACTIVE NIL AC NIL NIL NIL NIL NIL)))
     (BLOTTC)
     (SET VINV PARITY)
     (SET VTYPE (IF (EQN TYPE (QUOTE REAL)) TYPE (QUOTE SYMBOL)))
     (GO LOOP)
     ST (SET PARITY VINV)
     (SET VINV NIL)
     (MOVPDS VTYPE NIL)
     (INHERIT FORM)
     (SET PARITY (IF (NOT (EQ (MEMBER (QUOTE MINUS) VINV)
         (MEMBER (QUOTE MINUS) PARITY))) (QUOTE (MINUS)) NIL))
     (GO AT)))
   (FUNCTION (DIVIDE. SYMBOL)
    ((XTYPE SYMBOL) (INST SYMBOL))
    (BLOCK ((X SYMBOL))
     (IF (COMLCK 3) (RETURN (CLUNK)))
     (COMTYP XTYPE (CADDR EXP))
     (IF (OR VREG VINV (NOT (EXHOCKY INST NIL))) (MOVPDS XTYPE NIL))
     (SET X (VLIST))
     (COMVAL (CADR EXP) XTYPE NIL (QUOTE AC)) (INHERIT X)))
   (FUNCTION (MPYALG SYMBOL)
   ((TYPE SYMBOL) (XREG SYMBOL) (INST SYMBOL))                           0031600
   (BLOCK ((PARITY SYMBOL))                                              0031700
    (IF (NOT VCLASS)                                                     0031800
     (BLOCK NIL (INHERIT (CAR LOCLST))                                   0031900
      (SET LOCLST (CDR LOCLST)) G01599))                                 0032000
    LOOP (IF (NOT (AND (FULLW VBYTE)                                     0032100
       (EQN VCLASS (QUOTE ACTIVE)) (EQN (QUOTE AC) VREG)))               0032200
     (MOVACTIVE VTYPE (QUOTE AC) NIL))                                   0032300
    (SET PARITY (IF (EQ (MEMBER (QUOTE MINUS) VINV)                      0032400
       (MEMBER (QUOTE MINUS) (GVINV (CAR LOCLST))))                      0032500
      NIL (QUOTE (MINUS))))                                              0032600
    (INHERIT (CAR LOCLST))                                               0032700
    (ATTACH1 (CONS INST (MOVARG (IF (EQN TYPE (QUOTE INTEGER))           0032800
        VTYPE TYPE) (QUOTE AC) NIL (QUOTE LDA))))                        0032900
    (INHERIT (QUOTE (ACTIVE NIL NIL NIL NIL NIL (AC B) NIL)))            0033000
    (SET VINV PARITY)                                                    0033100
    (SET VREG XREG)                                                      0033200
    (SET VTYPE TYPE) (IF (SET LOCLST (CDR LOCLST)) (GO LOOP))))          0033300
  (FUNCTION (MPIALG SYMBOL)                                              0033400
   NIL (MPYALG (QUOTE INTEGER) (QUOTE B) (QUOTE MUL)))                   0033500
  (FUNCTION (MPRALG SYMBOL)                                              0033600
   NIL (MPYALG (QUOTE REAL) (QUOTE AC) (QUOTE FMP))))                    0033700
(ADDER (INSTRUCTIONS ((PLUS . LISP) NOVALUE)                             0033800
   NIL (BLOCK ((SYM SYMBOL)                                              0033900
     (FORM SYMBOL)                                                       0034000
     (VPS SYMBOL) (VPT SYMBOL) (VAS SYMBOL) (PTYPE SYMBOL))              0034100
    (SET SYM (COMARI 0 (COMARGS)                                         0034200
      (SPLUS . SYS)                                                      0034300
      PLIALG PLSMOV PLIMVP PLSPDL PLRALG PLSMOV PLRMVP PLSPDL))          0034400
    (IF VCLASS (IF (SET FORM (CAR SYM)) (GO PLU) (RETURN NIL))           0034500
     (CAR SYM) (GO COP))                                                 0034600
    (INHERIT (QUOTE (DATUM INTEGER NIL 0 NIL NIL NIL NIL)))              0034700
    (RETURN NIL)                                                         0034800
    COP (RESTORE (CAR SYM))                                              0034900
    LOOP (IF (NOT (SET FORM (CAR (SET SYM (CDR SYM)))))                  0035000
     (RETURN NIL))                                                       0035100
    PLU (ATTACH1 (LIST (QUOTE ARGS)))                                    0035200
    (MOVPDS VTYPE NIL)                                                   0035300
    (SET PTYPE (SET VPT VTYPE))                                          0035400
    (SET VPS (IF (MEMBER (QUOTE MINUS) VINV) FALSE 1))                   0035500
    (RESTORE FORM)                                                       0035600
    (SET VAS (IF (MEMBER (QUOTE MINUS) VINV) FALSE 1))                   0035700
    (SET VPT (IF (EQN VPT (QUOTE SYMBOL))                                0035800
      (QUOTE (SPLUS SMINS))                                              0035900
      (EQN VPT (QUOTE REAL))                                             0036000
      (QUOTE (SPLUR SMINR)) (QUOTE (SPLUI SMINI))))                      0036100
    (SET VPT (IF (EQUALN VPS VAS) (CAR VPT) (CADR VPT)))                 0036200
    (SET VINV (IF (MEMBER (QUOTE RECIP) VINV) (QUOTE (RECIP)) NIL))      0036300
    (MOVACTIVE (QUOTE SYMBOL) (QUOTE AC) NIL)                            0036400
    (IF (EQN PTYPE (QUOTE REAL)) (SET VTYPE (QUOTE REAL)))               0036500
    (CALCOMP VPT) (GO LOOP)))                                            0036600
  (FUNCTION (MOVPRD SYMBOL)                                              0036700
   ((FORM SYMBOL) (I SYMBOL))                                            0036800
   (BLOCK ((VCLASS SYMBOL)                                               0036900
     (VTYPE SYMBOL)                                                      0037000
     (VREG SYMBOL)                                                       0037100
     (VADDR SYMBOL)                                                      0037200
     (VBYTE SYMBOL)                                                      0037300
     (VBLOT SYMBOL) (VINV SYMBOL) (VIND SYMBOL) (LISTING SYMBOL))        0037400
    (VSET FORM) (RETURN (EXHOCKY I NIL))))                               0037500
  (FUNCTION (PLIMVP SYMBOL)                                              0037600
   ((FORM SYMBOL)) (MOVPRD FORM (QUOTE LDA)))                            0037700
  (FUNCTION (PLRMVP SYMBOL)                                              0037800
 ((FORM SYMBOL)) (MOVPRD FORM (QUOTE FAD)))
(FUNCTION (PLSMOV SYMBOL)
 ((FORM SYMBOL))
 (BLOCK ((LISTING SYMBOL))
  (RESTORE FORM)
  (MOVACTIVE VTYPE (QUOTE AC) NIL) (RETURN (CLUNK))))
(FUNCTION (PLSPOL SYMBOL)
 NIL (BLOCK ((TEM SYMBOL) (BYTE SYMBOL))
  (SET TEM (VLIST))
  (SET BYTE (IF (BLOCK ((LISTING SYMBOL))
     (MOVPOS VTYPE VBYTE)
     (RETURN (EXFOCKY (IF (EQN (QUOTE REAL) VTYPE)
        (QUOTE FAD) (QUOTE LDA)) NIL))) VBYTE NIL))
  (VSET TEM)
  (MOVPOS VTYPE BYTE) (RETURN (NCONC (VLIST) (LIST NIL)))))
(FUNCTION (PLSALG SYMBOL)
 ((TYPE SYMBOL) (BLOT SYMBOL) (ADD SYMBOL) (SUB SYMBOL))
 (BLOCK ((ACS SYMBOL) (LCS SYMBOL))
  (IF VCLASS (GO TAG))
  (INHERIT (CAR LOCLST))
  (SET LOCLST (CDR LOCLST))
  TAG (MOVACTIVE TYPE (QUOTE AC) NIL)
  (SET ACS (IF (MEMBER (QUOTE MINUS) VINV)
    (QUOTE (MINUS)) FALSE))
  LOOP (INHERIT (CAR LOCLST))
  (SET LCS (IF (MEMBER (QUOTE MINUS) VINV) (QUOTE (MINUS)) FALSE))
  (ATTACH1 (CONS (IF (EQN ACS LCS) ADD SUB)
    (MOVARG (IF (EQN TYPE (QUOTE INTEGER)) VTYPE TYPE)
     (QUOTE AC) NIL (QUOTE LDA))))
  (IF (SET LOCLST (CDR LOCLST)) (GO LOOP))
  (INHERIT (LIST (QUOTE ACTIVE)
    TYPE (QUOTE AC) NIL NIL NIL BLOT ACS)) (RETURN NIL)))
(FUNCTION (PLIALG SYMBOL)
 NIL (PLSALG (QUOTE INTEGER)
  (QUOTE (AC)) (QUOTE ADD) (QUOTE SUB)))
(FUNCTION (PLRALG SYMBOL)
 NIL (PLSALG (QUOTE REAL) (QUOTE (AC B)) (QUOTE FAD) (QUOTE FSB)))
(FUNCTION (MINUS1 SYMBOL) ((J SYMBOL)) (MINUS J))
(FUNCTION (COMINV SYMBOL)
 ((INV SYMBOL) (OTH SYMBOL) (FCN (FUNCTIONAL SYMBOL SYMBOL)))
 (BLOCK NIL (IF (COMLCK 2) (RETURN NIL))
  (COMEXP1 (CADR EXP))
  (IF (EQN VCLASS (QUOTE DATUM))
   (GO DAT)
   (MEMBER VTYPE (QUOTE (OCTAL INTEGER REAL SYMBOL))) (GO NUM))
  (COMERR (QUOTE (NON NUM ARG TO INV)))
  (RETURN NIL)
  NUM (IF (NOT (FULLW VBYTE)) (MOVACTIVE VTYPE (QUOTE AC) NIL))
  (IF (EQN VTYPE (QUOTE OCTAL)) (SET VTYPE (QUOTE INTEGER)))
  (SET VINV (IF (MEMBER INV VINV) NIL (LIST INV)))
  (RETURN NIL)
  DAT (IF (NUMBP VADDR) (GO DAT1))
  (COMER2 (CADR EXP) (QUOTE (NON NUM DATA TO INV)))
  (RETURN NIL)
  DAT1 (IF (EQ (SET VADDR (FCN VADDR)) 0)
   (SET VADDR (ITYPE VTYPE))) (INHERIT (COMDAT (CLUNK)))))
(MACRO ((DIFFERENCE . LISP) SYMBOL)
 ((EXP SYMBOL))
 (IF (COMLCK 3)
  0 (LIST (QUOTE PLUS)
   (CADR EXP) (LIST (QUOTE MINUS) (CADDR EXP)))))
(INSTRUCTIONS ((MINUS . LISP) NOVALUE)
 NIL (COMINV (QUOTE MINUS) (QUOTE RECIP) MINUS1))
 (INSTRUCTICNS ((ABS . LISP) NOVALLE)
  NIL (BLCCK ((Y SYMBCL))
   (IF (CCMLCK 2) (RETURN NIL))
   START (CCMEXP1 (CADR EXP))
   (IF (ECN (QLCTE DATUM) VCLASS)
    (GO DAT)
    (EQN VTYPE (QLOTE SYMBOL))
    (GC SYM)
    (MEMBER VTYPE (QUOTE (OCTAL INTEGER REAL))) (GC NUM))
   ERR (CCMERR (QLOTE (NON NUM ARG TO ABS)))
   (RETURN NIL)
   DAT (IF (NCT (NUMBP VADDR)) (GC ERR))
   (INHERIT (COMDAT (CLUNK)))
   (SET VADDR (ABS VADDR))
   (RETURN NIL)
   SYM (ATTACH (QLCTE (ARGS)))
   (MOVACTIVE (QUCTE SYMBCL) (QUOTE AC) NIL)
   (CALCOMP (QLCTE SYMABS))
   (GO HOME)
   NUM (IF (NCT (SET Y (MCVARG VTYPE (QUOTE AC) NIL (QUOTE LDA))))
    (SET Y (QLCTE (A.))))
   (ATTACH (CCNS (QUCTE LDM) Y))
   (INHERIT (LIST (QUCTE ACTIVE)
     VTYPE (QUCTE AC) NIL NIL NIL (QUCTE AC) VINV))
   HOME (SET VINV (IF (MEMBER (QUCTE RECIP) VINV)
     (QUCTE RECIP) NIL))))
 (INSTRUCTICNS ((SIGN . LISP) NOVALUE)
  NIL (BLCCK NIL (IF (CCMLCK 2) (RETURN NIL))
   (COMEXP1 (CADR EXP))
   (MOVACTIVE VTYPE (QUCTE AC) NIL)
   (IF (EQN VTYPE (QUOTE SYMBCL))
    (GO S)
    (MEMBER VTYPE (QUCTE (FUNCTIONAL BOOLEAN)))
    (COMERR (QUCTE (ILEG TYP TO SIGN))))
   (ATTACH (QLCTE (BOZ (D. 4))))
   (ATTACH (QLCTE (PER 3 0 43Q)))
   (ATTACH (QLCTE (LDA B.)))
   (ATTACH (QLCTE (CCN 1 (R L7.7 3Q5))))
   B (INHERIT (QUCTE (ACTIVE INTEGER AC NIL NIL NIL NIL NIL)))
   (RETURN NIL)
   S (ATTACH (QUCTE (ARGS))) (CALCCMP (QUOTE SYMSGN)) (GO B))))
(CHEAT (FUNCTICN (CHIZLE SYMBOL)
  ((FROM SYMBCL) (TO SYMBCL))
  (BLOCK NIL (IF (COMLCK 2) (RETURN NIL))
   (COMTYP FRCM (CADR EXP))
   (IF (EQN VCLASS (QUCTE DATUM)) (MOVACTIVE VTYPE (QUOTE AC) NIL))
   (IF (AND (CR (EQN FRCM (QUOTE INTEGER))
      (EQN TO (QUCTE INTEGER)) (EQN TC (QUOTE REAL))) VBYTE)
    (MOVACTIVE VTYPE (QUOTE AC) NIL))
   CAT (SET VTYPE TC) (RETURN NIL)))
 (FUNCTION (FTYPEP SYMBCL)
  ((X SYMBOL))
  (IF (FTYPP X)
   NIL (BLOCK NIL (COMER2 X (QUOTE (NCT A TYPE)))
    (INHERIT (QLOTE (DATUM OCTAL NIL 0 NIL NIL NIL NIL)))
    (RETURN TRUE))))
 (INSTRUCTICNS ((CCRE . LISP) NOVALUE)
  NIL (BLCCK NIL (IF (CCMLCK 2) (RETURN NIL))
   (COMEXP1 (CADR EXP))
   (IF (ECN (QUCTE DATUM) VCLASS)
    (GO DAT)
    (AND (OR (ECN VTYPE (QUOTE INTEGER)) (EQN (QUOTE OCTAL) VTYPE))
     (VINDX)) (GO LCC))
    XYZ (MOVACTIVE (QUOTE INTEGER) (QUOTE AC) NIL)
    (INHERIT (QUOTE (LOC OCTAL AC 0 NIL NIL NIL NIL)))
    (RETURN NIL)
    LOC (SET VTYPE (QUOTE OCTAL))
    (SET VIND TRUE)
    (SET VBYTE NIL)
    (RETURN NIL)
    CAT (INHERIT (COMDAT (VLIST)))
    (SET VADDR (CNVDATM VTYPE VADDR (QUOTE INTEGER)))
    (SET VCLASS (QUOTE LOC)) (SET VTYPE (QUOTE OCTAL))))
  (INSTRUCTIONS ((CHEAT . LISP) NOVALUE)
   NIL (IF (OR (COMLCK 4)
     (FTYPEP (CADR EXP)) (FTYPEP (CADDR EXP)))
    NIL (BLOCK ((X SYMBOL EXP) (EXP SYMBOL (CDDR EXP)))
     (RETURN (CHIZLE (CADR X) (CADDR X))))))
  (INSTRUCTIONS ((DRIVE . LISP) NOVALUE)
   NIL (IF (OR (COMLCK 3) (FTYPEP (CADR EXP)))
    NIL (COMTYP (CADR EXP) (CADDR EXP))))
  (INSTRUCTIONS ((CORENTRY . LISP) NOVALUE)
   NIL (IF (COMLCK 2)
    NIL (BLOCK NIL (INHERIT (QUOTE (LOC OCTAL NIL NIL NIL NIL NIL NIL)
      )) (SET VADDR (CONS (QUOTE ENTRY) (CDR EXP))))))
  (INSTRUCTIONS ((ENTRY . LISP) NOVALUE)
   NIL (IF (COMLCK 2)
    NIL (BLOCK NIL (ATTACH (CONS (QUOTE LDA)
       (LIST EXP (QUOTE (L567.7 R)))))
     (INHERIT (QUOTE (ACTIVE OCTAL AC NIL NIL NIL NIL NIL))))))
  (INSTRUCTIONS ((S2C. . LISP) NOVALUE)
   NIL (CHIZLE (QUOTE SYMBOL) (QUOTE OCTAL)))
  (INSTRUCTIONS ((R2C. . LISP) NOVALUE)
   NIL (CHIZLE (QUOTE REAL) (QUOTE OCTAL)))
  (INSTRUCTIONS ((B2C. . LISP) NOVALUE)
   NIL (CHIZLE (QUOTE BOOLEAN) (QUOTE OCTAL)))
  (INSTRUCTIONS ((I2C. . LISP) NOVALUE)
   NIL (CHIZLE (QUOTE INTEGER) (QUOTE OCTAL)))
  (INSTRUCTIONS ((F2C. . LISP) NOVALUE)
   NIL (CHIZLE (QUOTE FUNCTIONAL) (QUOTE OCTAL)))
  (INSTRUCTIONS ((C2S. . LISP) NOVALUE)
   NIL (CHIZLE (QUOTE OCTAL) (QUOTE SYMBOL)))
  (INSTRUCTIONS ((C2R. . LISP) NOVALUE)
   NIL (CHIZLE (QUOTE OCTAL) (QUOTE REAL)))
  (INSTRUCTIONS ((C2B. . LISP) NOVALUE)
   NIL (CHIZLE (QUOTE OCTAL) (QUOTE BOOLEAN)))
  (INSTRUCTIONS ((C2I. . LISP) NOVALUE)
   NIL (CHIZLE (QUOTE OCTAL) (QUOTE INTEGER)))
  (INSTRUCTIONS ((C2F. . LISP) NOVALUE)
   NIL (CHIZLE (QUOTE OCTAL) (QUOTE FUNCTIONAL))))
(WORDS (MACRO ((INVERT . LISP) SYMBOL)
   ((EXP SYMBOL))
   (IF (COMLCK 2)
    NIL (LIST (QUOTE WORDXOR) 7777777777777777Q (CADR EXP))))
  (INSTRUCTIONS ((WORDOR . LISP) NOVALUE)
   NIL (COMWRD 0Q (QUOTE ORA)
    (FUNARG SYMBOL ((A SYMBOL) (B SYMBOL)) (WORDOR A B))))
  (INSTRUCTIONS ((WORDAND . LISP) NOVALUE)
   NIL (COMWRD 7777777777777777Q (QUOTE ANA)
    (FUNARG SYMBOL ((A SYMBOL) (B SYMBOL)) (WORDAND A B))))
  (INSTRUCTIONS ((WORDXOR . LISP) NOVALUE)
   NIL (COMWRD 0Q (QUOTE XOR)
    (FUNARG SYMBOL ((A SYMBOL) (B SYMBOL)) (WORDXOR A B))))
  (FUNCTION (WRDHLP SYMBOL)
   ((LST SYMBOL))
   (COMOPT LST (FUNCTION (G01600 SYMBOL)
     NIL (BLOCK NIL (IF VCLASS (GO TAG))
      (INHERIT (CAR LOCLST))
      (SET LOCLST (CDR LOCLST))
      TAG (MOVACTIVE (QUOTE OCTAL) (QUOTE AC) NIL)
      LOP (INHERIT (CAR LOCLST))
      (ATTACH1 (CONS INSTRUCTION (MOVARG (QUOTE OCTAL)
         (QUOTE AC) NIL (QUOTE LDA))))
      (IF (SET LOCLST (CDR LOCLST)) (GO LOP))
      (INHERIT (QUOTE (ACTIVE OCTAL AC NIL NIL NIL (AC) NIL)))))
    PLSMOV (FUNCTION (G01601 SYMBOL)
     ((FORM SYMBOL))
     (AND (NOT (GVBYTE FORM)) (MOVPRD FORM (QUOTE LDA))))
    (FUNCTION (G01602 SYMBOL)
     NIL (BLOCK NIL (MOVPDS (QUOTE OCTAL) NIL)
      (RETURN (NCONC (VLIST) (LIST NIL)))))))
  (FUNCTION (COMWRD SYMBOL)
   ((IDATA SYMBOL)
    (INSTRUCTION SYMBOL) (LOGFCN (FUNCTIONAL SYMBOL SYMBOL SYMBOL)))
   (BLOCK ((TEM SYMBOL)
     (LST SYMBOL) (DATA SYMBOL) (EX SYMBOL) (DATLST SYMBOL))
    (SET DATA IDATA)
    (SET LST (QUOTE (NIL)))
    (IF (EQ (LENGTH EXP) 1) (GO NOARG))
    (SET EX EXP)
    LOOP (IF (NULL (SET EX (CDR EX))) (GO BEG))
    (SET TEM (CAR EX))
    (BLOCK ((VCLASS SYMBOL)
      (VTYPE SYMBOL)
      (VREG SYMBOL)
      (VADDR SYMBOL)
      (VIND SYMBOL) (VBYTE SYMBOL) (VINV SYMBOL) (LISTING SYMBOL))
     (COMTYP (QUOTE OCTAL) TEM)
     (IF (EQN (QUOTE DATUM) VCLASS) (GO DAT))
     (SET LST (CONS (CLUNK) LST))
     (GO G01603)
     DAT (SET DATA (LOGFCN DATA VADDR))
     (SET DATLST (NCONC DATLST LISTING)) G01603)
    (GO LOOP)
    BEG (LSTLST DATLST)
    (IF (NOT (EQUALN DATA IDATA))
     (SET LST (CONS (LIST (QUOTE DATUM)
        (QUOTE OCTAL) NIL DATA NIL NIL NIL NIL NIL) LST)))
    (SET LST (WRDHLP LST))
    (IF LST (GO FIN))
    NOARG (INHERIT (LIST (QUOTE DATUM)
      (QUOTE OCTAL) NIL IDATA NIL NIL NIL NIL))
    (RETURN NIL) FIN (INHERIT (CAR LST)) (LSTLST (CADR LST)))))
(ASSIGN (FUNCTION (SET. SYMBOL)
   ((X SYMBOL))
   (MOVLOC (GVTYPE X) (GVADDR X) (GVREG X) (GVIND X) (GVBYTE X)))
  (FUNCTION (EQTYPE SYMBOL)
   ((A SYMBOL) (B SYMBOL))
   (IF (EQN A B)
    A (OR (AND (EQN (QUOTE OCTAL) A) (EQN (QUOTE INTEGER) B))
     (AND (EQN (QUOTE INTEGER) A) (EQN (QUOTE OCTAL) B)))
    (QUOTE OCTAL) NIL))
  (FUNCTION (FULLP SYMBOL)
   ((TYP SYMBOL) (BYTE SYMBOL))
   (BLOCK ((X SYMBOL))
    (RETURN (OR (FULLW BYTE)
      (AND (EQN (QUOTE SYMBOL) TYP)
       (GR (SET X (CADR (WHATBITS BYTE))) 17)
       (EQ (REMAINDER X 6) 0))))))
```

```
(INSTRUCTIONS ((SET . LISP) NOVALUE)
 NIL (BLOCK ((X SYMBOL) (XBLOT SYMBOL) (TYP SYMBOL) (FREG SYMBOL))
  (IF (COMLOK 3) (RETURN NIL))
  (BLOCK ((VCLASS SYMBOL)
    (VTYPE SYMBOL)
    (VREG SYMBOL)
    (VADDR SYMBOL)
    (VIND SYMBOL)
    (VINV SYMBOL) (VBLOT SYMBOL) (VBYTE SYMBOL) (LISTING SYMBOL))
   (COMEXP1 (CADR EXP)) (SET X (CLUNK)) G01604)
  (IF (AND (EQN (GVCLAS X) (QUOTE LOC)) (NOT VINV)) (GO BEG))
  (COMER2 (CADR EXP) (QUOTE (NON LOC 1 ARG TO SET)))
  (INHERIT (QUOTE (DATUM OCTAL NIL 0Q NIL NIL NIL NIL)))
  (RETURN NIL)
  BEG (SET FREG (IF (NOT (MEMBER (QUOTE AC)
      (SET XBLOT (GVBLOT X))))
    (QUOTE AC)
    (NOT (MEMBER (QUOTE B) XBLOT))
    (QUOTE B) (NOT (MEMBER (QUOTE L) XBLOT)) (QUOTE L) NIL))
  (IF SCLASS (GO STAT)
   (SET TYP (EQTYPE XTYPE (GVTYPE X))) (GO ETYP))
  (COMEXP1 (CADDR EXP))
  (IF (MEMBER (QUOTE X2) XBLOT)
   (GO VREGP)
   (AND (SET TYP (EQTYPE (GVTYPE X) VTYPE))
    (NOT (EQN VCLASS (QUOTE DATUM)))) (GO SAVE))
  VREGP (IF (NOT VREG) (GO OK))
  SAFE (MOVPOS VTYPE NIL)
  OK (SET TYP (VLIST))
  (IF (EQN VADDR (QUOTE POP.)) (SET VADDR (QUOTE TOP.)))
  (LSTLST (LAST X))
  (SET. X)
  (INHERIT TYP)
  (IF (AND (EQN VCLASS (QUOTE LOC)) (EQN VADDR (QUOTE POP.)))
   (MOVACTIVE VTYPE (QUOTE AC) NIL))
  (RETURN NIL)
  ETYP (COMTYP TYP (CADDR EXP))
  (IF (MEMBER (QUOTE X2) XBLOT) (GO VREGP))
  SAVE (IF (NOT (FULLP TYP (GVBYTE X)))
   (GO VREGP)
   (MEMBER VREG XBLOT)
   (IF FREG (GO MACT) (GO SAFE))
   (EQN VCLASS (QUOTE ACTIVE))
   (IF (AND (NOT VINV) (FULLP VTYPE VBYTE)) (GO OK) (GO MACT))
   (NOT FREG) (GO OK))
  MACT (MOVACTIVE TYP FREG NIL)
  (GO OK)
  STAT (COMTYP (GVTYPE X) (CADDR EXP))
  (IF (OR (AND VREG (MEMBER (QUOTE X2) XBLOT))
    (AND (SET TYP (MEMBER VREG XBLOT)) (NOT FREG)))
   (MOVPOS VTYPE NIL) (AND TYP FREG) (MOVACTIVE VTYPE FREG NIL))
  (LSTLST (LAST X)) (SET. X)))
(INSTRUCTIONS ((LOCSET . LISP) NOVALUE)
 NIL (BLOCK ((TYP SYMBOL) (BLOT SYMBOL) (LST SYMBOL))
  (IF (COMLOK 3) (RETURN NIL))
  (COMEXP1 (CADDR EXP))
  (IF (AND (EQN (QUOTE LOC) VCLASS) (FULLW VBYTE) (NOT VINV))
   (GO A))
  (COMER2 (CADDR EXP) (QUOTE (NON FULL LOC 2 ARG IN LOCSET)))
  A (SET TYP VTYPE)
  (MAKELOC)
  (SET BLOT VBLOT)
  (SET LST LISTING)
```

```
    (VSET (QUOTE (NIL NIL NIL NIL NIL NIL NIL NIL)))
    (SET LISTING NIL)
    (COMEXP1 (CADR EXP))
    (IF (EQN VTYPE TYP) (GO XGQ))
    (COMERR (QUOTE (UNEQ TYPES IN LOCSET)))
    XGQ (IF (AND (EQN VCLASS (QUOTE LOC))
      (NOT VREG)
      VIND (NOT LISTING) (FULLW VBYTE) (NOT VBLOT) (NOT VINV))
     (GO B))
    (COMER2 (CADR EXP) (QUOTE (NON LOC VAR 1 ARG TO LOCSET)))
    B (SET LISTING (CONS (LIST (QUOTE STF) VADDR) LST))
    (INHERIT (QUOTE (LOC NIL AC 0 NIL NIL NIL NIL)))
    (SET VTYPE TYP) (SET VBLOT BLOT))))
(FOR (FUNCTION (FORSET SYMBOL)
   ((X SYMBOL))
   (IF (AND (ATOM X) (EQN X VADDR))
    NIL (FORX (LIST (QUOTE SET) VADDR X))))
  (FUNCTION (FORX SYMBOL)
   ((X SYMBOL)) (SET LISTING (NCONC LISTING (LIST X))))
  (FUNCTION (FORTRM SYMBOL)
   NIL (BLOCK ((X SYMBOL))
    (SET X (CDDDR EXP))
    (IF (SIM (QUOTE (WHILE S.)) (CAR X))
     (BLOCK NIL (FORX (LIST (QUOTE IF)
        (LIST (QUOTE NOT) (CADAR X)) (LIST (QUOTE GO) FGO)))
      (SET X (CDR X)) G01605))
    (IF (SIM (QUOTE (UNLESS S.)) (CAR X))
     (BLOCK NIL (FORX (LIST (QUOTE IF)
        (CADAR X) (LIST (QUOTE GO) TGO))) (SET X (CDR X)) G01606))
    (FORX (CAR X))))
  (MACRO ((FOR . LISP) SYMBOL)
   ((EXP SYMBOL))
   (BLOCK ((TGO SYMBOL)
     (FGO SYMBOL)
     (L SYMBOL)
     (LISTING SYMBOL)
     (G SYMBOL)
     (G1 SYMBOL)
     (G2 SYMBOL) (FL SYMBOL) (CFL SYMBOL) (VADDR SYMBOL))
    (IF (NOT (SIM (QUOTE (FOR S. (OR. ((OR. LOOP IN ON) S.)
          ((OR. RESET STEP) S. S.)
          (STEP S. S. (OR. EQ NQ LQ GQ LS GR) S.) NIL)
         (ANY. (WHILE S.)) (ANY. (UNLESS S.)) S.)) EXP)) (GO E))
    (SET TGO (GENID))
    (SET L (GENID))
    (SET FGO (GENID))
    (SET G1 (GENID))
    (SET G2 (GENID))
    (SET VADDR (CADR EXP))
    (IF (NULL (SET FL (CADDR EXP)))
     (BLOCK NIL (FORX TGO) (FORX L) (FORTRM) G01607)
     (EQN (SET CFL (CAR FL)) (QUOTE LOCP))
     (BLOCK NIL (FORX TGO)
      (FORX L) (FORSET (CADR FL)) (FORTRM) G01608)
     (EQN CFL (QUOTE RESET))
     (BLOCK NIL (FORSET (CADR FL))
      (FORX L) (FORTRM) (FORX TGO) (FORSET (CADDR FL)) G01609)
     (OR (EQN CFL (QUOTE IN)) (EQN CFL (QUOTE ON)))
     (BLOCK NIL (SET G (LIST (LIST G1 (QUOTE SYMBOL) (CADR FL))))
      (FORX L)
      (FORX (LIST (QUOTE IF)
        (LIST (QUOTE NULL) G1) (LIST (QUOTE GO) FGO)))
      (FORSET (IF (EQN (CAR FL) (QUOTE ON))
      G1 (LIST (QUOTE CAR) G1)))
     (FORTRM)
     (FORX TGC)
     (FORX (LIST (QUOTE SET) G1 (LIST (QUOTE CDR) G1))) G01610)
    (EQN CFL (QUOTE STEP))
    (BLOCK NIL (IF (NUMBP (SET CFL (CADDR FL)))
      (SET G1 CFL) (SET G (LIST (LIST G1 (QUOTE ASSIGNED) CFL))))
     (FORSET (CADR FL))
     (FORX L)
     (IF (SET CFL (CDDDR FL))
      (BLOCK NIL (IF (NUMBP (CADR CFL))
        (SET G2 (CADR CFL))
        (SET G (NCONC G (LIST (LIST G2 (QUOTE ASSIGNED)
            (CADR CFL))))))
       (FORX (LIST (QUOTE IF)
         (LIST (CAR CFL) VADDR G2) (LIST (QUOTE GO) FGO))) G01612))
     (FORTRM)
     (FORX TGC) (FORSET (LIST (QUOTE PLUS) VADDR G1)) G01611))
    (RETURN (NCONC (NCONC (LIST (QUOTE BLOCK) G) LISTING)
     (LIST (LIST (QUOTE GO) L) FGO)))
   E (COMERR (QUOTE (ILLEGAL FOR STATEMENT))))))
****END OF FILE DETECTED
```

```
(MOVEPO (FUNCTION (REVA2L SYMBOL)
  ((XREG SYMBOL))
   (BLOCK ((VREG SYMBOL)) (SET VREG XREG) (RETURN (TRANSA2L))))
  (FUNCTION (LXN SYMBOL)
   ((N SYMBOL)) (IF (GR (ABS N) 377777Q) NIL (LIST N (QUOTE R))))
  (FUNCTION SYMOD (S)
   (BLOCK NIL (IF (IDP S)
     (RETURN (LIST (QUOTE ID) S))
     (BOOLP S)
     (RETURN (IF S 1 0))
     (INTP S)
     (IF (LS (ABS S) 2Q5) (RETURN (WORDOR 0 (PLUS 6Q5 S))))
     (OCTALP S)
     (IF (EQ 0 (WORDAND 7777777776Q5 S)) (RETURN (WORDOR 2Q5 S))))
    (RETURN (LIST (QUOTE QUOTE) S))))
  (FUNCTION (SPARAM SYMBOL)
   ((XBYTE SYMBOL))
   (BLOCK ((I SYMBOL) (TO SYMBOL))
    (RETURN (LIST (IF (AND (EQ (CADR (SET I (WHATBITS VBYTE)))
         (CADR (SET TO (WHATBITS XBYTE))))
        (OR (EQN VTYPE (QUOTE INTEGER))
         (LS (PLUS (CAR I) (CADR I)) 48)))
       (QUOTE SHIFT) (QUOTE MASK))
      (DIFFERENCE (CAR I) (CAR TO)) (QUOTE R)))))
  (FUNCTION (WHATBITS SYMBOL)
   ((X SYMBOL))
   (IF (NULL X)
    (QUOTE (0 48))
    (ATOM X)
    (BLOCK NIL (COMER2 X (QUOTE (ILLEGAL BYTE VALUE)))
     (RETURN (QUOTE (0 48)))) X))
  (FUNCTION (SVACT SYMBOL)
   ((R SYMBOL) (B SYMBOL))
   (BLOCK NIL (SET VCLASS (QUOTE ACTIVE))
    (SET VREG R) (SET VBYTE B) (SET VADDR (SET VIND NIL))))
  (FUNCTION (BBND SYMBOL)
   ((B SYMBOL))
   (AND (EQ (REMAINDER (CAR (SET B (WHATBITS B))) 6) 0)
    (EQ (REMAINDER (CADR B) 6) 0)))
  (FUNCTION (ISINV SYMBOL)
   ((X SYMBOL))
   (BLOCK ((Y SYMBOL))
    (SET Y 0)
    AA (IF (CLVINV X) (GO BB))
    (RETURN (EQ 1 (REMAINDER Y 2))) BB (SET Y (PLUS Y 1)) (GO AA)))
  (FUNCTION (CLVINV SYMBOL)
   ((X SYMBOL))
   (IF (ATOM VINV)
    (IF (EQN VINV X) (BLOCK NIL (SET VINV NIL) (RETURN TRUE)) NIL)
    (BLOCK ((A SYMBOL))
     S (IF (NULL VINV)
      (RETURN (BLOCK NIL (SET VINV (REVERSE A)) (RETURN NIL)))
      (EQN (CAR VINV) X)
      (RETURN (BLOCK NIL (SET VINV (APPEND (REVERSE A) (CDR VINV)))
        (RETURN TRUE))))
     (SET A (CONS (CAR VINV) A)) (SET VINV (CDR VINV)) (GO S))))
  (FUNCTION (LOPC SYMBOL)
   ((R SYMBOL))
   (IF (NOT (MEMBER R (QUOTE (AC B L))))
    (BLOCK NIL (COMER2 R (QUOTE (NOT LEGAL REGISTER FOR LDA CLASS)))
     (RETURN (IF (ISINV (QUOTE MINUS))
       (QUOTE LOADCOMP) (QUOTE LOAD))))
    (ISINV (QUOTE MINUS))
  (CADR (SASSOC R (QUOTE ((AC LDC) (B LBC) (L LLC))) CADRNIL))
  (CADR (SASSOC R (QUOTE ((AC LDA) (B LDB) (L LDL))) CADRNIL))))
(FUNCTION (LOCMP SYMBOL)
 ((R SYMBOL))
 (BLOCK ((VINV SYMBOL))
  (SET VINV (QUOTE (MINUS))) (RETURN (LOPC R))))
(FUNCTION (CADRNIL SYMBOL) NIL (QUOTE (NIL NIL)))
(FUNCTION (TRANSA2L SYMBOL)
 NIL (IF (NUMBP VREG)
  (CONS VREG (QUOTE (Z.)))
  (CADR (SASSOC VREG (QUOTE ((AC A.) (B B.) (L L.))) CADRNIL))))
(FUNCTION (CONVP SYMBOL)
 ((NEW SYMBOL))
 (CADR (SASSOC NEW (CDR (SASSOC VTYPE (QUOTE ((OCTAL (INTEGER OI)
        (REAL IR)
        (SYMBOL (OCT2SYM . LISP)) (BOOLEAN TRU) (OCTAL V))
       (INTEGER (REAL IR)
        (SYMBOL (INT2SYM . LISP))
        (OCTAL MZ) (BOOLEAN TRU) (INTEGER V))
       (REAL (INTEGER (ROUND . LISP))
        (SYMBOL (REAL2SYM . LISP))
        (OCTAL (OCTROUND . LISP)) (BOOLEAN TRU) (REAL V))
       (SYMBOL (INTEGER (SYM2INT . LISP))
        (REAL (SYM2REAL . LISP))
        (OCTAL (SYM2OCT . LISP))
        (BOOLEAN SP) (FUNCTIONAL (SYM2FORM . LISP)) (SYMBOL V))
       (FUNCTIONAL (SYMBOL (FORM2SYM . LISP))
        (BOOLEAN TRU) (FUNCTIONAL V))
       (BOOLEAN (SYMBOL V) (BOOLEAN V))))
     (FUNCTION (G01613 SYMBOL) NIL (QUOTE (A (A A)))))) CADRNIL)))
(FUNCTION (ADDRMODS SYMBOL)
 ((I SYMBOL) (R SYMBOL))
 (IF (NULL I)
  (IF (NULL R)
   0 (EQN R (QUOTE AC))
   15 (NUMBP R)
   R (COMER2 R (QUOTE (ERROR IN VREG OR XREG NO TAG))))
  (NULL R)
  (QUOTE I)
  (EQN R (QUOTE AC))
  (QUOTE (15 I))
  (NUMBP R)
  (LIST R (QUOTE I))
  (ATOM R)
  (COMER2 R (QUOTE (ILLEG TAG)))
  (NULL (CAR R)) (QUOTE I) (LIST (CAR R) (QUOTE I))))
(FUNCTION (BMODS SYMBOL)
 ((XCLASS SYMBOL) (XBYTE SYMBOL))
 (IF (AND (FULLW VBYTE) (FULLW XBYTE))
  NIL (AND (ATOM VBYTE)
   (EQN VBYTE XBYTE) (EQN XCLASS (QUOTE ACTIVE)))
  VBYTE (BLOCK ((E1 SYMBOL)
    (E2 SYMBOL) (E3 SYMBOL) (L1 SYMBOL) (L2 SYMBOL))
   (IF (EQN XCLASS (QUOTE ACTIVE)) (GO A))
   (SET E1 (QUOTE S))
   (SET E2 (TRANS VBYTE))
   (SET E3 (TRANS XBYTE))
   B (IF (OR (NULL E2) (NULL E3))
    (RETURN 0)
    (OR (EQ (SET L1 (LENGTH E2)) (SET L2 (LENGTH E3)))
     (EQ L1 1) (EQ L2 1))
    (GO C)
    (GR L1 L2)
```

```
     (SET E2 (LIST (LAST E2)))
     (EQN E1 (QUOTE 'L))
     (SET E3 (LIST (LAST E3))) (RETURN 7777777777777777Q))
    C (RETURN (MAKEID (COMPRESS (APPEND (CONS E1 E2)
         (CONS (QUOTE '.) E3))))))
    A (SET E1 (QUOTE L))
     (SET E2 (TRANS XBYTE)) (SET E3 (TRANS VBYTE)) (GO B))))
  (FUNCTION (TRANS SYMBOL)
   ((B SYMBOL))
   (IF (NULL B)
    (QUOTE ('7))
    (ATOM B) (COMER2 B (QUOTE (NOT A BYTE MODIFIER))) (TRANS1 B)))
  (FUNCTION (TRANS1 SYMBOL)
   ((B SYMBOL))
   (BLOCK ((R SYMBOL) (N SYMBOL) (A SYMBOL))
    (SET R (CAR B))
    (SET N (CADR B))
    (IF (OR (GR (SET N (PLUS R N)) 48)
      (NOT (EQ (REMAINDER R 6) 0)) (NOT (EQ (REMAINDER N 6) 0)))
     (RETURN NIL))
    X (IF (NULL (CAR (SET A (CONS (GETN (QUOTE (0 '7 6 '6 12 '5 18 '4
           24 '3 30 '2 36 '1 42 '0)) R) A))))
     (RETURN (COMER2 B (QUOTE (ILLEGAL MODIFIER)))))
    (IF (EQ (SET R (PLUS R 6)) N) (RETURN A)) (GO X)))
  (FUNCTION (TAGF SYMBOL)
   ((B SYMBOL) (A SYMBOL))
   (IF (OR (EQ 0 A) (NULL A))
    (IF (NULL B) 0 B)
    (EQ 0 B)
    A (NULL B)
    A (ATOM A)
    (IF (ATOM B) (LIST B A) (CONS A B))
    (ATOM B) (CONS B A) (APPEND B A)))
  (FUNCTION SASSOC (X L (FN (FUNCTIONAL SYMBOL)))
   (IF (SET X (FIND X L)) X (FN))))
(MOVEP1 (FUNCTION (CNVL2AC SYMBOL)
   ((TYP SYMBOL) (BYT SYMBOL))
   (BLOCK ((A SYMBOL) (B SYMBOL))
    (IF (NULL (SET A (CONVP TYP)))
     (RETURN 0)
     (NOT (ATOM A))
     (ATTACH (QUOTE (ARGS)))
     (EQN A (QUOTE V))
     (RETURN (BLOCK NIL (SET VTYPE TYP) (RETURN 2)))
     (EQN A (QUOTE OI))
     (GO NC)
     (EQN A (QUOTE TRU))
     (RETURN 1)
     (EQN A (QUOTE IR))
     (IF (NOT (FULLW BYT)) (RETURN 0) NIL)
     (NOT (OR (EQN A (QUOTE MZ)) (EQN A (QUOTE SP)))) (RETURN 5))
    (MOVACTIVE VTYPE (QUOTE AC) NIL)
    (IF (ATOM A) (GO OPEN))
    (ATTACH (LIST (QUOTE CALL) A))
    (BLOTTO)
    FIN (SVACT (QUOTE AC) NIL)
    (SET VTYPE TYP)
    (IF (FULLW BYT) (RETURN NIL))
    (RETURN (ACT2ACT (QUOTE AC) BYT))
    OPEN (IF (EQN A (QUOTE IR))
     (BLOCK NIL (ATTACH (QUOTE (FLT (ENTRY B48.))))
      (BLOTCH (QUOTE B)))
     (EQN A (QUOTE MZ))
   (ATTACH (QUOTE (BUC (ENTRY I2CENT) 0 4)))
   (EQN A (QUOTE SP)) (GO S2B))
  (BLOTCH (QUOTE AC))
  (GO FIN)
  S2B (ATTACH (QUOTE (BUC (ENTRY STBENT) 0 4)))
  (INHERIT (QUOTE (ACTIVE BOOLEAN AC NIL NIL NIL (AC) NIL)))
  (RETURN NIL)
  NO (IF (BBND VBYTE) (RETURN 3))
  (RETURN (MOVOI (QUOTE AC) BYT))))
(FUNCTION (DXREG SYMBOL)
 ((R SYMBOL)) (IF (NUMBP R) (QUOTE LDX) (QUOTE LDA)))
(FUNCTION (MLDX SYMBOL)
 ((I SYMBOL) (R SYMBOL))
 (IF (NULL (CDR I))
  (LIST (QUOTE LDX) (CAR I) 0 R)
  (APPEND (CONS (QUOTE LDX) I) (LIST R))))
(FUNCTION (L2AP. SYMBOL)
 NIL (BLOCK ((X SYMBOL) (Y SYMBOL))
  (IF (NOT (EQN VCLASS (QUOTE LOC)))
   (RETURN NIL)
   VIND (GO IND)
   (NOT (ATOM (SET X VADDR)))
   (GO XRS)
   (NULL (SET X (CADR (SASSOC VADDR (QUOTE ((A. AC)
         (B. B) (L. L))) CADRNIL)))) (RETURN NIL))
  (SVACT X VBYTE)
  (RETURN TRUE)
  IND (IF (NOT (AND (EQN VADDR (QUOTE A.)) (NULL VREG)))
   (RETURN NIL))
  (SET VADDR 0)
  (SET VIND NIL)
  (SET VREG 15)
  (RETURN NIL)
  XRS (IF (NOT (AND (MEMBER (QUOTE Z.) X) (EQ 2 (LENGTH X))))
   (RETURN NIL))
  XRS1 (IF (NUMBP (SET Y (CAR X)))
   (GO XRS2) (NULL (SET X (CDR X))) (RETURN NIL))
  (GO XRS1) XRS2 (SVACT Y VBYTE) (RETURN TRUE)))
(FUNCTION (ITSTRU SYMBOL)
 ((R SYMBOL) (B SYMBOL))
 (BLOCK NIL (SETTRU) (RETURN (MOVACTIVE (QUOTE BOOLEAN) R B))))
(FUNCTION (SETTRU SYMBOL)
 NIL (BLOCK NIL (SET VCLASS (QUOTE DATUM))
  (SET VTYPE (QUOTE BOOLEAN))
  (SET VADDR (QUOTE TRUE)) (SET VREG (SET VIND (SET VBYTE NIL)))))
(FUNCTION (LXRM SYMBOL)
 ((B SYMBOL))
 (IF (OR (NULL B) (AND (NOT (ATOM B)) (EQ 0 (CAR B))))
  (QUOTE RA) (AND (NOT (ATOM B)) (EQ (CAR B) 24)) (QUOTE LA) NIL))
(FUNCTION (BEQ SYMBOL)
 ((XB SYMBOL)) (OR (EQ VBYTE XB) (AND (FULLW VBYTE) (FULLW XB))))
(FUNCTION (NADDR SYMBOL)
 ((VAL SYMBOL) (TYPE SYMBOL) (B SYMBOL))
 (BLOCK ((C SYMBOL) (V SYMBOL) (VBYTE SYMBOL))
  (IF (NUMBP (SET V (BMODS (QUOTE ACTIVE) B)))
   (RETURN (SHFTRA VAL TYPE B))
   (EQN VTYPE (QUOTE BOOLEAN))
   (BLOCK NIL (SET VAL (IF (NULL VAL) 0 1)) (SET C V))
   (EQN TYPE (QUOTE OCTAL))
   (GO OCF) (NOT (NUMBP VAL)) (SET VAL 0))
  ND1 (IF (EQN TYPE (QUOTE INTEGER))
   (SET C (TAGF V (QUOTE S))) (SET C V))
  (IF (OR (AND (REALP VAL)
        (IF (EQ VAL 0) (BLOCK NIL (SET VAL 0) (RETURN NIL)) TRUE))
       (GR (ABS VAL) 377777Q))
      (RETURN (CONS (LIST (QUOTE NUMBER) VAL) (IF C (LIST C) NIL))))
     (RETURN (LIST VAL (IF V (TAGF (QUOTE R) C)
        (TAGF (QUOTE L567.7) (TAGF (QUOTE R) C)))))
     OOF (IF (NOT (NUMBP VAL))
      (GO OOF2) (EQ 0 (WORDAND VAL 4Q15)) (GO ND1))
     (RETURN (CONS (LIST (QUOTE NUMBER) VAL)
       (IF C (LIST C) NIL))) OOF2 (SET VAL 0) (GO ND1)))
  (FUNCTION (SHFTRA SYMBOL)
   ((VAL SYMBOL) (TYPE SYMBOL) (B SYMBOL))
   (BLOCK ((V SYMBOL))
     (SET V (WHATBITS B))
     (IF (EQN (QUOTE REAL) TYPE)
      (RETURN (IF (EQ 0 VAL) (QUOTE (0 R)) NIL))
      (EQN (QUOTE BOOLEAN) TYPE)
      (IF VAL (SET VAL 1) (RETURN (QUOTE (0 R))))
      (NOT (NUMBP VAL)) (SET VAL 0))
     NUM (IF (EQN TYPE (QUOTE INTEGER))
      (GO A2)
      (NOT (EQN TYPE (QUOTE OCTAL)))
      (RETURN NIL)
      (EQ 0 (WORDAND VAL 4Q15))
      (GO A1)
      (FULLW B)
      (RETURN (CONS (LIST (QUOTE NUMBER) VAL) NIL))
      (SET VAL (WORDAND VAL (MMSK (LIST 0 (CADR V))))))
     A1 (SET B NIL)
     (GO A3)
     A2 (SET B (QUOTE (S)))
     A3 (SET TYPE (IF (LS VAL 0) -1 1))
     (IF (GR (SET VAL (SHIFT (WORDAND (ABS VAL)
          (PLUS (SHIFT 1Q (CADR V)) -1))
        (REMAINDER (CAR V) 6))) 3777777Q)
      (RETURN (CONS (LIST (QUOTE NUMBER) (TIMES TYPE VAL)) NIL)))
     (SET V (GETN (QUOTE (0 (R L4567.7)
        1 (R L3456.7)
        2 (R L2345.7)
        3 (R L1234.7)
        4 (R L0123.7) 5 (R L012.7) 6 (R L01.7) 7 (R L0.7)))
      (IQUOTIENT (CAR V) 6)))
     (RETURN (LIST (WORDOR 0Q (TIMES TYPE VAL))
       (APPEND (IF V V (QUOTE (R))) B)))))
  (FUNCTION (MMSK SYMBOL)
   ((B SYMBOL))
   (BLOCK NIL (SET B (WHATBITS B))
     (RETURN (WORDAND (SHIFT 37777777777777777Q (CAR B))
       (SHIFT 37777777777777777Q (PLUS -47 (CAR B) (CADR B)))))))
  (FUNCTION (MDECR SYMBOL)
   NIL (IF (EQN VCLASS (QUOTE ACTIVE))
     (TRANSA2L)
     (EQN VCLASS (QUOTE LOC))
     (IF (OR VIND VREG) (COMERR (QUOTE (CAN NOT BE A DECR))) VADDR)
     (NOT (EQN VCLASS (QUOTE DATUM)))
     (COMER2 VCLASS (QUOTE (NOT A LEGAL CLASS)))
     (OR (EQN VTYPE (QUOTE REAL))
      (EQN VTYPE (QUOTE OCTAL)) (EQN VTYPE (QUOTE INTEGER)))
     (LIST (QUOTE NUMBER) VADDR)
     (EQN VTYPE (QUOTE SYMBOL))
     (SYMOD VADDR)
     (EQN VTYPE (QUOTE BOOLEAN))
     (IF VADDR 1 0) (COMER2 VTYPE (QUOTE (IS NOT A LEGAL DECR))))))
(MOVEP2 (FUNCTION (CNVD SYMBOL)
 ((NT SYMBOL))
 (BLOCK NIL (IF (NULL (CONVP NT))
   (RETURN (COMERR (QUOTE (TYPES NOT CONVERTIBLE)))))
  (SET VADDR (CNVDATM VTYPE VADDR NT)) (SET VTYPE NT)))
(FUNCTION (CNVDATM SYMBOL)
 ((OTYP SYMBOL) (VAL SYMBOL) (NTYP SYMBOL))
 (BLOCK ((W SYMBOL) (VTYPE SYMBOL))
  (IF (OR (EQN NTYP (QUOTE SYMBOL)) (EQN OTYP NTYP)) (RETURN VAL))
  (SET VTYPE OTYP)
  (IF (NOT (ATOM (SET W (CONVP NTYP))))
   (GO CC)
   (NULL W)
   (RETURN NIL)
   (EQN W (QUOTE V))
   (RETURN VAL)
   (EQN W (QUOTE OI))
   (RETURN (IF (NUMBP VAL) (PLUS VAL 0) 0))
   (EQN W (QUOTE IR))
   (GO CCC)
   (EQN W (QUOTE TRU))
   (RETURN (QUOTE TRUE))
   (EQN W (QUOTE MZ))
   (RETURN (IF (EQ 0 VAL) 0 (WORDOR OQ VAL)))
   (EQN W (QUOTE SP))
   (RETURN (IF (OR (NULL VAL) (EQN VAL (QUOTE FALSE)))
     NIL (QUOTE TRUE))))
  CC (IF (EQN OTYP (QUOTE REAL))
   (IF (NOT (NUMBP VAL))
    (GO CCC)
    (EQN NTYP (QUOTE INTEGER))
    (RETURN (ENTIER VAL))
    (EQN NTYP (QUOTE OCTAL))
    (RETURN (WORDOR OQ (ENTIER VAL))) NIL)
   (EQN OTYP (QUOTE SYMBOL))
   (IF (EQN NTYP (QUOTE FUNCTIONAL))
    (RETURN (COMERR (QUOTE (CNV OF A DATUM TO TYPE FUNCTIONAL))))
    (MEMBER NTYP (QUOTE (INTEGER OCTAL REAL))) (GO CCC)))
  (RETURN (COMERR (LIST OTYP (QUOTE TO)
     NTYP (QUOTE (DOESNT MAKE SENSE FOR DATUM)))))
  CCC (IF (NOT (NUMBP VAL))
   (BLOCK NIL (COMER2 VAL (QUOTE (IS NOT A NUMBER YOU KNOW)))
    (SET VAL 0)))
  (RETURN (IF (EQN NTYP (QUOTE INTEGER))
    (ENTIER VAL)
    (EQN NTYP (QUOTE REAL))
    (FLOAT VAL)
    (EQN NTYP (QUOTE OCTAL)) (WORDOR OQ (ENTIER VAL)) 0))))
(FUNCTION (MOVARG SYMBOL)
 ((XTYPE SYMBOL) (XREG SYMBOL) (XBYTE SYMBOL) (ICLASS SYMBOL))
 (IF (EQN ICLASS (QUOTE BXE))
  (MDECR)
  (NOT (EQN VTYPE XTYPE))
  (MOVARG2 XTYPE XREG XBYTE ICLASS)
  (NULL (SET XTYPE (EXHOCKY ICLASS XBYTE)))
  (MOVACTIVE VTYPE XREG XBYTE)
  (EQN ICLASS (QUOTE LDX)) (CDR (MLDX XTYPE XREG)) XTYPE))
(FUNCTION (MOVARG2 SYMBOL)
 ((XTYPE SYMBOL) (XREG SYMBOL) (XBYTE SYMBOL) (ICLASS SYMBOL))
 (BLOCK ((X SYMBOL))
  (IF (NULL (SET X (CONVP XTYPE)))
   (RETURN (COMERR (QUOTE (TYPE CONV NOT LEGAL))))
   (NOT (ATOM X))
   (RETURN (MOVACTIVE XTYPE XREG XBYTE))
   (EQN X (QUOTE OI))
   (GO AFTER)
   (EQN X (QUOTE V))
   (GO BEFORE)
   (EQN X (QUOTE TRU))
   (GO DAT) (RETURN (MOVACTIVE XTYPE XREG XBYTE)))
  AFTER (SET X (MOVARG VTYPE XREG XBYTE ICLASS))
  (GO END)
  BEFORE (SET X (BLOCK ((VTYPE SYMBOL))
    (SET VTYPE XTYPE) (RETURN (MOVARG XTYPE XREG XBYTE ICLASS))))
  (GO END)
  DAT (SET X (BLOCK ((VCLASS SYMBOL)
     (VTYPE SYMBOL)
     (VREG SYMBOL)
     (VIND SYMBOL) (VBYTE SYMBOL) (VINV SYMBOL) (VADDR SYMBOL))
    (SETTRU) (RETURN (MOVARG XTYPE XREG XBYTE ICLASS))))
  END (IF (NULL X) (SET VTYPE XTYPE)) (RETURN X)))
(FUNCTION (EXHOCKY SYMBOL)
 ((C SYMBOL) (B SYMBOL))
 (BLOCK ((V SYMBOL) (W SYMBOL) (A SYMBOL))
  (IF (EQN C (QUOTE LDA))
   (GO A)
   (EQN C (QUOTE LDX))
   (GO X)
   (AND (EQN C (QUOTE FAD)) (FULLW B) (FULLW VBYTE))
   (GO F) (RETURN NIL))
  A (IF (EQN VCLASS (QUOTE DATUM))
   (GO DA)
   (NUMBP (SET C (BMODS (QUOTE ACTIVE) B)))
   (RETURN NIL) (EQN VCLASS (QUOTE ACTIVE)) (GO AA))
  (SET V VIND)
  (SET W VREG)
  (SET A VADDR)
  AO (IF (EQN VTYPE (QUOTE INTEGER))
   (SET C (TAGF (IF (FULLW VBYTE) 0 (QUOTE S))
     (TAGF C (ADDRMODS V W)))) (SET C (TAGF C (ADDRMODS V W))))
  (IF (NULL C) (RETURN (LIST A)) (RETURN (LIST A C)))
  AA (SET V (SET W NIL))
  (SET A (TRANSA2L))
  (GO AO)
  DA (IF (AND (EQN VTYPE (QUOTE REAL)) (FULLW B))
   (RETURN (LIST (LIST (QUOTE NUMBER) VADDR)))
   (OR (EQN VTYPE (QUOTE INTEGER)) (EQN VTYPE (QUOTE OCTAL)))
   (RETURN (NADDR VADDR VTYPE B))
   (EQN VTYPE (QUOTE SYMBOL))
   (GO DSYM)
   (EQN VTYPE (QUOTE BOOLEAN))
   (RETURN (NADDR VADDR (QUOTE BOOLEAN) B))
   (EQN VTYPE (QUOTE FUNCTIONAL))
   (RETURN (LIST VADDR)) (RETURN NIL))
  DSYM (SET W (QUOTE (R)))
  (IF (AND (NOT (ATOM (SET A (SYMCD VADDR))))
    (IF (EQN (CAR A) (QUOTE QUOTE)) (NULL (SET W NIL)) NIL)
    (FULLW B))
   (RETURN (LIST A))
   (FULLW B)
   (RETURN (CONS A (IF (NULL W) NIL (QUOTE ((R L4567.7))))))
   (NUMBP (SET V (BMODS (QUOTE ACTIVE) B)))
   (RETURN (IF (ATOM A) (SHFTRA A (QUOTE OCTAL) B) NIL))
   (NULL W) (RETURN (LIST A V)) (RETURN (LIST A (CONS V W))))
  X (IF (NOT (EQN (LXRM B) (QUOTE RA)))
   (RETURN NIL)
   (EQN VCLASS (QUOTE DATUM))
```

```
     (GO LDXR)
     (NULL (SET A (LXRM VBYTE)))
     (RETURN NIL)
     (EQN VCLASS (QUOTE ACTIVE))
     (RETURN (IF (EQN A (QUOTE RA))
       (IF (NUMBP VREG)
        (LIST 0 (LIST (QUOTE R) VREG))
        (EQN VREG (QUOTE AC))
        (LIST 0 (QUOTE (R 15))) (LIST (TRANSA2L)))
       (LIST (TRANSA2L) A)))
     (RETURN (LIST VADDR (TAGF A (ADDRMODS VIND VREG)))))
    LDXR (IF (OR (EQN VTYPE (QUOTE INTEGER))
      (EQN VTYPE (QUOTE OCTAL)))
     (RETURN (LXN VADDR))
     (EQN VTYPE (QUOTE REAL))
     (RETURN (LXN (ENTIER VADDR)))
     (EQN VTYPE (QUOTE SYMBOL))
     (RETURN (LIST (SYMCD VADDR) (QUOTE R)))
     (EQN VTYPE (QUOTE BOOLEAN))
     (RETURN (LIST (IF VADDR 1 0) (QUOTE R)))
     (EQN VTYPE (QUOTE FUNCTIONAL))
     (RETURN (LIST (LIST (QUOTE FUNCTION) VADDR) (QUOTE R)))
     (RETURN NIL))
    F (IF (EQN VCLASS (QUOTE DATUM))
     (GO FD) (EQN VTYPE (QUOTE REAL)) (GO FR) (RETURN NIL))
    FD (IF (EQN VTYPE (QUOTE REAL))
     (RETURN (LIST (LIST (QUOTE NUMBER) VADDR)))
     (OR (EQN VTYPE (QUOTE INTEGER)) (EQN VTYPE (QUOTE OCTAL)))
     (RETURN (LIST (LIST (QUOTE NUMBER) (FLOAT VADDR))))
     (RETURN NIL))
    FR (IF (EQN VCLASS (QUOTE ACTIVE))
     (RETURN (LIST (TRANSA2L)))
     (EQN VCLASS (QUOTE LOC))
     (RETURN (LIST VADDR (ADDRMODS VIND VREG))))))
  (FUNCTION (MOVOI SYMBOL)
   ((XREG SYMBOL) (XBYTE SYMBOL))
   (BLOCK NIL (MOVACTIVE VTYPE XREG XBYTE)
    (SET VTYPE (QUOTE INTEGER)) (RETURN NIL))))
(MOVEP3 (FUNCTION (MAKELOC SYMBOL)
   NIL (BLOCK NIL (IF (NOT (AND (EQN VCLASS (QUOTE LOC))
       (FULLW VBYTE) (NULL VINV)))
     (RETURN (COMER2 (VLIST) (QUOTE (NOT LEGAL FOR MAKELOC))))
     (ATOM VREG)
     (IF (NULL VREG) (SET VREG 0) NIL) (SET VREG (CAR VREG)))
    (IF VIND (GO INDIRECT)
     (OR (NULL VADDR) (EQ VADDR 0))
     (GO TONLY)
     (MEMBER VREG (QUOTE (AC 15)))
     (GO AVADDR)
     (EQ 0 VREG)
     (BLOCK NIL (ATTACH (LIST (QUOTE LDX) VADDR (QUOTE R) 4))
      (SET VREG 4))
     (ATTACH (LIST (QUOTE BAX) (QUOTE (D. 1)) VREG VADDR)))
    (BLOTCH VREG)
    (GO LONLY)
    AVADDR (ATTACH (LIST (QUOTE ADD) VADDR (QUOTE (R L567.7))))
    (GO DONE)
    TONLY (IF (MEMBER VREG (QUOTE (AC 15))) (GO DONE))
    LONLY (ATTACH (LIST (QUOTE LDA) (TRANSA2L) (QUOTE L567.7)))
    (GO DONE)
    INDIRECT (ATTACH1 (LIST (QUOTE LDA) VADDR VREG))
    DONE (SET VTYPE (QUOTE SYMBOL))
    (BLOTCH (QUOTE AC)) (RETURN (SVACT (QUOTE AC) NIL))))
```

```
(FUNCTION (MOVACTIVE SYMBOL)
 ((XTYPE SYMBOL) (XREG SYMBOL) (XBYTE SYMBOL))
 (BLOCK ((S SYMBOL) (R SYMBOL))
  (IF VINV (GO SPECH))
  MOVA (IF (EQN VTYPE XTYPE)
   (GO NOCNV)
   (EQN VCLASS (QUOTE LOC))
   (GO L2A)
   (EQN VCLASS (QUOTE ACTIVE))
   (GO A2L)
   (EQN VCLASS (QUOTE DATUM))
   (GO CNVDAT)
   (RETURN (COMER2 VCLASS (QUOTE (NOT A CLASS)))))
  CNVDAT (CNVD XTYPE)
  DAT2ACT (IF (CANSTZ XBYTE (LIST (REVA2L XREG)))
   (GO DN)
   (NULL (SET R (EXHOOKY (DXREG XREG) XBYTE)))
   (RETURN (BLOCK NIL (SET VBYTE NIL)
     (MOVACTIVE XTYPE (QUOTE AC) NIL) (ACT2ACT XREG XBYTE)))
   (NUMBP XREG)
   (ATTACH1 (MLDX R XREG)) (ATTACH1 (CONS (LOPC XREG) R)))
  DN (BLOTCH XREG)
  (RETURN (SVACT XREG XBYTE))
  L2A (IF (OR (EQN VADDR (QUOTE PUSHA.))
    (EQN VADDR (QUOTE PUSHP.))) (SET VADDR (QUOTE POP.)))
  A2L (IF (NULL (SET S (CNVL2AC XTYPE (IF (EQN XREG (QUOTE AC))
       XBYTE NIL))))
   (RETURN (ACT2ACT XREG XBYTE))
   (EQ 0 S)
   (GO CNVERR)
   (EQ S 1)
   (RETURN (ITSTRU XREG XBYTE))
   (EQ S 2)
   (GO NOCNV) (EQ 3 S) (RETURN (MOVOI XREG XBYTE)))
  CNVERR (COMERR (LIST (VLIST)
    (QUOTE MOVACTIVE)
    (QUOTE TO) XTYPE XREG XBYTE (QUOTE ILLEGAL) S))
  (SET VTYPE XTYPE)
  (GO MOVA)
  NOCNV (IF (EQN VCLASS (QUOTE LOC))
   (RETURN (LOC2ACT XREG XBYTE))
   (EQN VCLASS (QUOTE DATUM))
   (GO DAT2ACT)
   (NOT (EQN VCLASS (QUOTE ACTIVE)))
   (RETURN (BLOCK NIL (COMER2 (VLIST)
      (QUOTE (NOT LEGAL FOR MOVACTIVE)))
     (SVACT XREG XBYTE) (SET VTYPE XTYPE) (RETURN NIL))))
  A2A (RETURN (ACT2ACT XREG XBYTE))
  SPECH (IF (AND (EQN VTYPE (QUOTE SYMBOL)) (ISINV (QUOTE MINUS)))
   (LSYMNS)) (GO MOVA)))
(FUNCTION (LSYMNS SYMBOL)
 NIL (BLOCK NIL (ATTACH (QUOTE (ARGS)))
  (MOVACTIVE VTYPE (QUOTE AC) NIL) (CALCOMP (QUOTE MINSYM))))
(FUNCTION (ACT2ACT SYMBOL)
 ((XREG SYMBOL) (XBYTE SYMBOL))
 (BLOCK ((A SYMBOL))
  (IF (NOT (EQN VCLASS (QUOTE ACTIVE)))
   (RETURN (BLOCK NIL (COMERR (QUOTE (ACT2ACT ENTERED WITH VCLASS
       NOT ACTIVE))) (RETURN (MOVACTIVE VTYPE XREG XBYTE))))
   (EQN VREG XREG) (GO BONLY) (BEQ XBYTE) (GO RONLY))
  BOTH (IF (AND (OR (NUMBP VREG) (NUMBP XREG))
    (LXRM VBYTE) (LXRM XBYTE))
   (GO HLFS)
  (OR (EQN VREG (QUOTE AC)) (EQN VREG (QUOTE B)))
  (GO B1)
  (OR (EQN XREG (QUOTE AC)) (EQN XREG (QUOTE B)))
  (GO R1)
  (NOT (NUMBP VREG))
  (GO ISL) (NULL (SET A (EXHOCKY (QUOTE LDX) XBYTE))) (GO USEB))
 (ATTACH1 (MLDX A XREG))
 (GO DONE1)
 B1 (IF (AND (OR (EQN XREG (QUOTE AC))
     (EQN XREG (QUOTE B)) (EQN XREG (QUOTE L)))
   (NOT (NUMBP (SET A (BMODS (QUOTE ACTIVE) XBYTE)))))
  (GO B2) (NUMBP XREG) (GO B3))
 (ACT2ACT VREG XBYTE)
 (IF (NOT (EQN XREG (QUOTE AC)))
  (GO RONLY)
  (AND (EQN (QUOTE ANS) (CAR (SET A (CAR LISTING))))
   (EQ (CADR A) (TRANSA2L)))
  (SET LISTING (CONS (CONS (QUOTE ANA) (CDR A)) (CDR LISTING)))
  (GO RONLY))
 (GO DONE)
 R1 (ACT2ACT XREG (IF XBYTE NIL VBYTE))
 (GO BONLY)
 USEB (ACT2ACT (QUOTE B) NIL)
 (ATTACH (MLDX (QUOTE (B. RA)) XREG))
 (GO DONE1)
 ISL (IF (NUMBP (SET A (BMODS (QUOTE ACTIVE) XBYTE)))
  (GO SWAP))
 B2 (ATTACH1 (LIST (LOPC XREG) (TRANSA2L) (IF A A 0)))
 (GO DONE1)
 B3 (ACT2ACT VREG (QUOTE (0 18)))
 (GO XR)
 BONLY (IF (NOT (ISINV (QUOTE MINUS)))
  (GO NOINV)
  (NUMBP VREG)
  (RETURN (BLOCK ((VINV SYMBOL))
    (SET VINV (QUOTE (MINUS)))
    (ACT2ACT (QUOTE AC) XBYTE) (ACT2ACT XREG XBYTE)))
  (ATTACH (LIST (LDCMP VREG) (TRANSA2L))))
 NOINV (IF (BEQ XBYTE) (RETURN (SVACT XREG XBYTE)))
 ADJBYT (IF (NUMBP XREG)
  (GO SWAP)
  (SET A (EXHOCKY (QUOTE LDA) XBYTE))
  (ATTACH1 (CONS (LOPC XREG) A))
  (EQN VREG (QUOTE L)) (GO TSTL) (GO ABSHFT))
 (GO DONE)
 ABSHFT (IF (NOT (EQ 0 (CADR (SET A (SPARAM XBYTE)))))
  (ATTACH (CONS (IF (EQN VREG (QUOTE AC))
     (QUOTE SFA) (EQN VREG (QUOTE B)) (QUOTE SFB) (QUOTE SHIFT))
    (CDR A))))
 (IF (EQN (CAR A) (QUOTE SHIFT)) (GO DONE))
 MSK (SET VBYTE (CONS (CAR (SET A (WHATBITS XBYTE)))
   (IF (GR (CADR (SET VBYTE (WHATBITS VBYTE)))
     (CADR (SET A (WHATBITS XBYTE)))) (CDR A) (CDR VBYTE))))
 (IF (OR (EQN VREG (QUOTE B)) (EQN VREG (QUOTE L))) (GO INB))
 (ATTACH (CONS (QUOTE ANA)
   (NADDR (MMSK VBYTE) (QUOTE OCTAL) NIL)))
 (GO DONE)
 INB (ATTACH (CONS (QUOTE LDA)
   (NADDR (MMSK VBYTE) (QUOTE OCTAL) NIL)))
 (BLOTCH (QUOTE AC))
 (ATTACH (LIST (QUOTE ANS) (TRANSA2L)))
 (GO DONE)
 TSTL (IF (EQ (CADR (SPARAM XBYTE)) 0) (GO MSK))
```

```
    SWAP (ACT2ACT (QUOTE B) XBYTE)
    (RETURN (ACT2ACT XREG XBYTE))
    RONLY (IF (NUMBP XREG)
     (GO STIXR)
     (NOT (BEQ XBYTE))
     (GO ER1)
     (EQN VREG (QUOTE AC))
     (ATTACH (LIST (QUOTE STF) (REVA2L XREG)))
     (ATTACH (LIST (LOPC XREG) (TRANSA2L))))
    (GO DONE1)
    ER1 (ATTACH (LIST (LOPC XREG)
      (TRANSA2L)
      (IF (NUMBP (SET A (BMODS (QUOTE ACTIVE) XBYTE))) 0 A)))
    (COMERR (QUOTE (BYTE TROUBLE AT RONLY IN ACT2ACT)))
    (GO DONE1)
    HLFS (IF (NOT (NUMBP VREG))
     (GO HLFS1)
     (NUMBP XREG)
     (ATTACH (LIST (QUOTE LDX) 0 VREG XREG))
     (SET A (EXHOCKY (QUOTE LDA) XBYTE))
     (ATTACH1 (CONS (LOPC XREG) A)) (GO HLFO))
    (GO DONE1)
    HLFO (ATTACH (LIST (QUOTE STX) (REVA2L XREG) 0 VREG))
    (SVACT XREG NIL)
    (BLOTCH XREG)
    (GO BONLY)
    HLFS1 (IF (EQ 24 (SET A (CAR (WHATBITS VBYTE))))
     (SET A (QUOTE LA)) (EQ 0 A) NIL (GO ON))
    (ATTACH (LIST (QUOTE LDX) (TRANSA2L) A XREG))
    (GO DONE1)
    ON (IF (EQN (QUOTE L) VREG) (SET A (QUOTE B)) (SET A VREG))
    (ACT2ACT A NIL)
    (RETURN (ACT2ACT XREG XBYTE))
    STIXR (IF (NUMBP VREG) (GO XR))
    (ATTACH (LIST (IF (EQN VREG (QUOTE AC))
       (QUOTE STF)
       (EQN VREG (QUOTE B))
       (QUOTE STB) (EQN VREG (QUOTE L)) (QUOTE STL) (QUOTE LOADXR))
      (REVA2L XREG)))
    (GO DONE1)
    XR (IF (SET A (EXHOCKY (QUOTE LDX) XBYTE))
     (ATTACH1 (MLDX A XREG)) (GO SWAP))
    DONE1 (BLOTCH XREG) DONE (RETURN (SVACT XREG XBYTE)))))
(MOVEP4 (FUNCTION (LOC2ACT SYMBOL)
   ((R SYMBOL) (B SYMBOL)) (IF (L2AP.) (ACT2ACT R B) (LC2ACT R B)))
  (FUNCTION (LC2ACT SYMBOL)
   ((R SYMBOL) (B SYMBOL))
   (BLOCK ((X SYMBOL))
    (IF (NULL (SET X (EXHOCKY (DXREG R) B)))
     (GO WORK)
     (NUMBP R) (ATTACH1 (MLDX X R)) (ATTACH1 (CONS (LOPC R) X)))
    END (BLOTCH R)
    (RETURN (SVACT R B))
    WORK (IF (NUMBP R)
     (IF (NUMBP (BMODS (QUOTE ACTIVE) B))
      (IF (EQN VADDR (QUOTE A.))
       (SET X (QUOTE AC)) (SET X (QUOTE B))) (SET X (QUOTE L)))
     (EQN R (QUOTE L)) (SET X (QUOTE B)) (GO SAME))
    EX1 (MOVACTIVE VTYPE X B)
    (GO EXIT)
    SAME (IF (BBND VBYTE)
     (LC2ACT R VBYTE)
     (EQ R (QUOTE AC))
   (BLOCK NIL (ATTACH1 (LIST (LOPC R) VADDR (ADDRMODS VIND VREG)))
    (SVACT R VBYTE)
    (BLOTCH R)
    (ACT2ACT R (CONS (CAR (WHATBITS B)) (CDR (WHATBITS VBYTE)))))
   (BLOCK NIL (SET X (QUOTE AC)) (GO EX1)))
  EXIT (RETURN (ACT2ACT R B))))
(FUNCTION (MOVPOS SYMBOL)
 ((XTYPE SYMBOL) (XBYTE SYMBOL))
 (BLOCK NIL (MOVLOC XTYPE (IF (OR (EQN XTYPE (QUOTE SYMBOL))
     (EQN XTYPE (QUOTE FUNCTIONAL)))
    (QUOTE PUSHP.) (QUOTE PUSHA.)) NIL NIL XBYTE)
  (SET VADDR (QUOTE POP.))))
(FUNCTION (MOVLOC SYMBOL)
 ((XTYPE SYMBOL)
  (XADDR SYMBOL) (XREG SYMBOL) (XIND SYMBOL) (XBYTE SYMBOL))
 (BLOCK ((V SYMBOL) (Y SYMBOL) (L SYMBOL))
  (IF (NULL XREG) (RETURN (MVLOCP XTYPE XADDR XREG XIND XBYTE)))
  (SET V (VLIST))
  (SET Y (CAR (SET L (BLOCK ((LISTING SYMBOL) (VBLOT SYMBOL))
      (MVLOCP XTYPE XADDR XREG XIND XBYTE)
      (RETURN (CONS VBLOT LISTING))))))
  (IF (NOT (DSTRYD XREG Y)) (GO END))
  FNCALL (IF (NULL (SET L (CDR L)))
   (GO NOCALL)
   (EQN (CAAR L) (QUOTE CALL)) (GO MOVSAV) (GO FNCALL))
  NOCALL (IF (NOT (ACEQ (QUOTE AC) XREG)) (GO MOVSAV))
  (VSET V)
  (SET Y (CAR (SET L (BLOCK ((LISTING SYMBOL) (VBLOT SYMBOL))
      (MOVACTIVE XTYPE (QUOTE B) XBYTE)
      (ACT2LOC XADDR XREG XIND XBYTE)
      (RETURN (CONS VBLOT LISTING))))))
  (IF (DSTRYD XREG Y) (GO MOVSAV))
  END (SET LISTING (NCONC (CDR L) LISTING))
  (SET VBLOT (UNION VBLOT Y))
  (RETURN NIL)
  MOVSAV (VSET V)
  (RETURN (MOVSAV XTYPE XADDR XREG XIND XBYTE NIL))))
(FUNCTION (MVLOCP SYMBOL)
 ((XTYPE SYMBOL)
  (XADDR SYMBOL) (XREG SYMBOL) (XIND SYMBOL) (XBYTE SYMBOL))
 (BLOCK ((A SYMBOL))
  (IF (NOT (EQN VTYPE XTYPE)) (GO SE))
  ST (IF (OR (EQN VCLASS (QUOTE ACTIVE)) (L2AP.))
   (RETURN (ACT2LOC XADDR XREG XIND XBYTE))
   (EQN VCLASS (QUOTE LOC))
   (GO EQQ)
   (NOT (EQN VCLASS (QUOTE DATUM)))
   (RETURN (COMER2 VCLASS (QUOTE (NOT A PROPER CLASS))))
   (CANSTZ XBYTE (LIST XADDR (ADDRMODS XIND XREG)))
   (RETURN (SVLOC XADDR XREG XIND XBYTE)) (FULLW XBYTE) (GO STF))
  (SET A (QUOTE L))
  (GO C2)
  EQQ (IF (AND (BEQ XBYTE)
    (NOT VINV) (EQ XADDR VADDR) (ACEQ XREG VREG) (EQ XIND VIND))
   (RETURN NIL) (NOT (FULLW XBYTE)) (GO C1))
  STF (MOVACTIVE XTYPE (QUOTE AC) NIL)
  (RETURN (ACT2LOC XADDR XREG XIND XBYTE))
  C1 (IF (BBND XBYTE)
   (IF (BBND VBYTE) (SET A (QUOTE L)) (SET A (QUOTE B)))
   (SET A (QUOTE AC)))
  C2 (MOVACTIVE XTYPE A XBYTE)
  (RETURN (ACT2LOC XADDR XREG XIND XBYTE))
  SE (IF (EQN VCLASS (QUOTE DATUM))
   (CNVC XTYPE)
   (NOT (ATOM (SET A (CONVP XTYPE))))
   (GO STF)
   (EQN A (QUOTE V))
   (SET VTYPE XTYPE)
   (EQN A (QUOTE TRU))
   (SETTRU) (EQN A (QUOTE OI)) (GO OINK) (GO STF))
  (GO ST)
  OINK (MOVLOC VTYPE XADDR XREG XIND XBYTE)
  (SET VTYPE XTYPE) (RETURN NIL)))
(FUNCTION (DSTRYC SYMBOL)
 ((XREG SYMBOL) (Y SYMBOL))
 (OR (MEMBER (SET XREG (IF (ATOM XREG) XREG (CAR XREG))) Y)
  (AND (EQN XREG (QUOTE AC)) (MEMBER 15 Y))
  (AND (EQ XREG 15) (MEMBER (QUOTE AC) Y))))
(FUNCTION (MOVSAV SYMBOL)
 ((XTYPE SYMBOL)
  (XADDR SYMBOL)
  (XREG SYMBOL) (XIND SYMBOL) (XBYTE SYMBOL) (Z SYMBOL))
 (BLOCK ((X SYMBOL) (Y SYMBOL))
  (SET X (IF (ATOM XREG) XREG (CAR XREG)))
  (IF XIND (BLOCK NIL (ATTACH (IF (AND (NUMBP XREG) (LS XREG 9))
      (LIST (QUOTE LDX) XADDR XREG XREG)
      (LIST (QUOTE LDA) XADDR 15)))
    (BLOTCH (QUOTE AC)) (SET XIND NIL) (SET XADDR 0) G01614))
  (IF (AND (EQN VCLASS (QUOTE LOC))
    (ACEQ (SET Y (IF (ATOM VREG) VREG (CAR VREG))) X))
   (COMERR (QUOTE (VREG AND XREG ARE THE SAME FOR MOVLOC))))
  (SVXREG XADDR X)
  (IF (AND (EQN VCLASS (QUOTE LOC))
    (MEMBER VADDR (QUOTE (POP. TOP. PUSHA. PUSHP.))))
   (SET VADDR (QUOTE (TOP. -1))))
  (MOVLOC XTYPE (QUOTE POP.) NIL (QUOTE INDIRECT) XBYTE)
  (IF (MEMBER XADDR (QUOTE (POP. TOP. PUSHA. PUSHP.)))
   (ATTACH (QUOTE (POP. 1)))) (RETURN (SVLOC NIL NIL NIL NIL))))
(FUNCTION (SVXREG SYMBOL)
 ((XADDR SYMBOL) (XREG SYMBOL))
 (IF (EQ 0 XADDR)
  (STXREG XREG (QUOTE PUSHP.))
  (BLOCK NIL (BLOTCH XREG) (RETURN NIL))
  NIL (IF (MEMBER XREG (QUOTE (AC 15)))
   (ATTACH (LIST (QUOTE ADD) XADDR (QUOTE (R L567.7))))
   (ATTACH (LIST (QUOTE BAX) (QUOTE (D. 1)) XREG XADDR)))
  (STXREG XREG (QUOTE PUSHP.)) NIL))
(FUNCTION (STXREG SYMBOL)
 ((XREG SYMBOL) (WHERE SYMBOL))
 (BLOCK NIL (IF (OR (EQN XREG (QUOTE AC)) (EQ XREG 15))
   (ATTACH (LIST (QUOTE STF) WHERE))
   (ATTACH (LIST (QUOTE STX) WHERE 0 XREG)))))
(FUNCTION (CANSTZ SYMBOL)
 ((XBYTE SYMBOL) (IADR SYMBOL))
 (BLOCK NIL (IF (OR (AND (EQ VADDR 0)
     (OR (MEMBER VTYPE (QUOTE (INTEGER REAL)))
      (AND (EQN VTYPE (QUOTE OCTAL))
       (NOT (EQUALN 77777777777777770 VADDR)))))
    (AND (MEMBER VTYPE (QUOTE (SYMBOL BOOLEAN)))
     (MEMBER VADDR (QUOTE (NIL FALSE)))))
   (GO X)
   (NOT (AND (EQN VTYPE (QUOTE OCTAL))
     (EQUALN VADDR 7777777777777777Q) (FULLW XBYTE)))
   (RETURN NIL))
  (ATTACH1 (IF (EQ IADR (QUOTE (A.)))
    (QUOTE (LDA 77Q (L7.7 R S))) (CONS (QUOTE STMZ) IADR)))
  (RETURN TRUE)
  X (IF (FULLW XBYTE)
   (ATTACH1 (CONS (QUOTE STZ) IADR))
   (BLOCK NIL (ATTACH1 (COMPMSK XBYTE))
    (ATTACH1 (CONS (QUOTE ANS) IADR)) (BLOTCH (QUOTE AC)) G01615))
  (RETURN TRUE)))
(FUNCTION (ACEQ SYMBOL)
 ((V SYMBOL) (X SYMBOL))
 (OR (EQ V X)
  (AND (EQN V (QUOTE AC)) (EQ X 15))
  (AND (EQN X (QUOTE AC)) (EQ V 15))))
(FUNCTION (ACT2LOC SYMBOL)
 ((XADDR SYMBOL) (XREG SYMBOL) (XIND SYMBOL) (XBYTE SYMBOL))
 (BLOCK ((X SYMBOL) (Y SYMBOL))
  C (IF (NOT (EQN VCLASS (QUOTE ACTIVE)))
   (RETURN (BLOCK NIL (COMERR (QUOTE (ACT2LOC ENTERED AND VCLASS IS
       NOT ACTIVE)))
     (RETURN (MOVLOC VTYPE XADDR XREG XIND XBYTE))))
   VINV (RETURN (BLOCK NIL (MOVACTIVE VTYPE VREG XBYTE)
     (RETURN (ACT2LOC XADDR XREG XIND XBYTE))))
   (NUMBP VREG)
   (GO STX)
   (FULLW XBYTE)
   (GO STF) (NUMBP (SET X (BMODS (QUOTE LOC) XBYTE))) (GO MSKS))
  (ATTACH1 (LIST (STOC VREG) XADDR (TAGF (ADDRMODS XIND XREG) X)))
  (GO XX)
  STF (ATTACH1 (LIST (IF (OR (EQ VREG (QUOTE AC))
      (NOT (FULLW VBYTE)))
     (BLOCK NIL (ACT2ACT (QUOTE AC) NIL) (RETURN (QUOTE STF)))
     (STOC VREG)) XADDR (ADDRMODS XIND XREG)))
  (GO XX)
  STX (IF (NULL (SET X (STXR XBYTE))) (GO CREG))
  (ATTACH1 (LIST (QUOTE STX)
    XADDR (TAGF (IF (EQN X (QUOTE RA)) 0 X)
     (ADDRMODS XIND XREG)) VREG))
  (GO XX)
  CREG (IF (NUMBP (BMODS (QUOTE LOC) XBYTE))
   (SET X (QUOTE AC)) (SET X (QUOTE L)))
  (ACT2ACT X VBYTE)
  (GO 0)
  MSKS (ACT2ACT (QUOTE L) XBYTE)
  (SET X (IF (EQ 0 (SET X (ADDRMODS XIND XREG))) NIL (LIST X)))
  (SET LISTING (NCONC (REVERSE (LIST (COMPMSK XBYTE)
      (APPEND (LIST (QUOTE ANA)
        (IF (MEMBER XADDR (QUOTE (POP. PUSHA. PUSHP.)))
         (QUOTE TOP.) XADDR)) X)
      (QUOTE (ORA L.))
      (APPEND (LIST (QUOTE STF) XADDR) X))) LISTING))
  (BLOTCH (QUOTE AC)) XX (RETURN (SVLOC XADDR XREG XIND XBYTE))))
(FUNCTION (COMPMSK SYMBOL)
 ((XBYTE SYMBOL))
 (BLOCK ((VBYTE SYMBOL))
  (RETURN (CONS (QUOTE LDA)
    (NADDR (WORDXOR 7777777777777777Q (MMSK XBYTE))
     (QUOTE OCTAL) NIL)))))
(FUNCTION (ATTACH1 SYMBOL)
 ((X SYMBOL))
 (BLOCK ((Y SYMBOL))
  (IF (AND (CDR X) (SET Y (CDDR X))) (GO A))
  (ATTACH X)
  (RETURN NIL)
  A (IF (ATOM (CAR Y))
   (IF (NULL (CDR Y))
      (IF (EQ 0 (CAR Y))
       (ATTACH (LIST (CAR X) (CADR X))) (ATTACH X)) (ATTACH X))
     (GO SBL)) (RETURN NIL) SBL (ATTACH X)))
  (FUNCTION (STXR SYMBOL)
   ((XBYTE SYMBOL))
   (IF (OR (ATOM XBYTE) (EQ 18 (CADR XBYTE))) (LXRM XBYTE) NIL))
  (FUNCTION (FULLW SYMBOL)
   ((XBYTE SYMBOL)) (OR (NULL XBYTE) (EQ XBYTE (QUOTE (0 48)))))
  (FUNCTION (STOC SYMBOL)
   ((REG SYMBOL))
   (IF (MEMBER REG (QUOTE (AC B L)))
    (CADR (SASSOC REG (QUOTE ((AC STA) (B STB) (L STL))) CADRNIL))
    (BLOCK NIL (COMER2 REG (QUOTE (NOT A LEGAL REGISTER FOR STA CLASS)
      )) (RETURN (QUOTE STORE)))))
  (FUNCTION (SVLOC SYMBOL)
   ((ADDR SYMBOL) (REG SYMBOL) (IND SYMBOL) (BYTE SYMBOL))
   (BLOCK NIL (SET VCLASS (QUOTE LOC))
    (SET VADDR ADDR) (SET VREG REG) (SET VIND IND) (SET VBYTE BYTE))))
****END OF FILE DETECTED
```

```
(SIM2 (SECTION MANIP BOOLEAN)
  (FUNCTION ((SIM . LISP) BOOLEAN)
   ((P SYMBOL) (X SYMBOL))
   (IF (ATOM P)
    (ATMFN P X)
    (ATOM (CAR P))
    (IF (EQ (QUOTE OR.) (CAR P)) (ORFN (CDR P) X) (SIMFN P X))
    (EQ (CAAR P) (QUOTE ANY.))
    (ANYFN P X)
    (EQ (CAAR P) (QUOTE C.))
    (CNTFN (CADAR P) (CADDAR P) (CADDDAR P) (CDR P) X) (SIMFN P X)))
  (FUNCTION (SIMFN BOOLEAN)
   ((P SYMBOL) (X SYMBOL))
   (IF (ATOM X)
    FALSE (AND (SIM (CAR P) (CAR X)) (SIM (CDR P) (CDR X)))))
  (FUNCTION (ATMFN BOOLEAN)
   ((P SYMBOL) (X SYMBOL))
   (BLOCK ((Z SYMBOL (FIND P METALST)))
    (IF (NULL Z) (RETURN (EQUALN P X)))
    (BLOCK ((F (FUNCTIONAL BOOLEAN SYMBOL) (CDR Z)))
     (RETURN (F X)))))
  (FUNCTION (ORFN BOOLEAN)
   ((L SYMBOL) (X SYMBOL))
   (BLOCK ((Y SYMBOL))
    (FOR Y (IN L) (IF (SIM Y X) (RETURN TRUE))) (RETURN FALSE)))
  (FUNCTION (ANYFN BOOLEAN)
   ((P SYMBOL) (X SYMBOL))
   (IF (ATOM X)
    (SIM (CDR P) X)
    (ORFN (CDAR P) (CAR X)) (SIM (CDR P) (CDR X)) (SIM (CDR P) X)))
  (FUNCTION (CNTFN BOOLEAN)
   ((I INTEGER) (J INTEGER) (M SYMBOL) (P SYMBOL) (X SYMBOL))
   (IF (EQ I 0)
    (IF (EQ J 0)
     (SIM P X)
     (AND (NOT (ATOM X)) (SIM M (CAR X)))
     (CNTFN 0 (DIFFERENCE J 1) M P (CDR X)) (SIM P X))
    (AND (NOT (ATOM X))
     (SIM M (CAR X))
     (CNTFN (DIFFERENCE I 1) (DIFFERENCE J 1) M P (CDR X)))))
  (DECLARE (METALST SYMBOL))
  (SET METALST (LIST (CONS (QUOTE A.)
     (FUNCTION ((G02427 . G02428) BOOLEAN) ((A SYMBOL)) (ATOM A)))
    (CONS (QUOTE N.)
     (FUNCTION ((G02429 . G02430) BOOLEAN) ((N SYMBOL)) (NUMBP N)))
    (CONS (QUOTE S.)
     (FUNCTION ((G02431 . G02432) BOOLEAN) ((X SYMBOL)) TRUE))
    (CONS (QUOTE L.)
     (FUNCTION ((G02433 . G02434) BOOLEAN) ((X SYMBOL)) (LISTP X)))
    (CONS (QUOTE ID.)
     (FUNCTION ((G02435 . G02436) BOOLEAN) ((X SYMBOL)) (IDP X)))
    (CONS (QUOTE V.)
     (FUNCTION ((G02437 . G02438) BOOLEAN)
      ((X SYMBOL)) (SIM (QUOTE (OR. ID. (ID. . ID.))) X)))))))
****END OF FILE DETECTED
```

```
(SLPCEC (SECTICN SYS)
  MACRO1 (((NAME2FLNC (LAMBDA (L)
      (SUBST (CACR L)
       (QUCTE V)
       (QUCTE (CHEAT INTEGER FUNCTICNAL (PLUS (S20. (GETFREE (CAR V)
            (CCR V))) 17777777Q)))))))
  (SECTICN (SUPV CCMPIL SYS LISP) SYMBCL)
  (FLNCTICN ((SIM . LISP) BOOLEAN) ((P SYMBOL) (X SYMBCL)))
  (DECLARE ((ONEPASS . LISP) BOOLEAN FLUID)
   ((LISPVALUE . LISP) SYMBOL FLUIC)
   ((TRYLC . LISP) SYMBCL FLUID)
   ((PARENFLAG . LISP) BCOLEAN FLUIC)
   ((PRNERR . LISP) BCOLEAN FLUID)
   ((PRNMAX . LISP) INTEGER OWN)
   ((BACTRC . LISP) SYMBCL FLUID)
   ((BACKTRACE . LISP) SYMBOL OWN)
   ((BADEXP . LISP) SYMBCL FLUID)
   ((PRNIL . LISP) BOOLEAN FLUID)
   ((PRNLAP . LISP) BCOLEAN FLUID) ((BINLAP . LISP) BOOLEAN FLUID))
  (FLNCTICN ((FVLIS1 . CCMPIL) SYMBCL) ((X SYMBOL)))
  (FLNCTICN ((CCMER2 . CCMPIL) SYMBCL) (A B))
  (FLNCTICN ((FUNCTIO . COMPIL) SYMBCL) (A))
  (FLNCTICN ((FNDEC . CCMPIL) SYMBCL) ((EXP SYMBOL)))
  (FLNCTICN ((DECL1 . CCMPIL) SYMBCL) ((D SYMBCL)))
  (FLNCTICN ((SECSET . CCMPIL) SYMBCL)
   ((IDLIST SYMBCL) (DFTYPE SYMBOL)))
  (FLNCTICN ((DEFALLT . COMPIL) SYMBCL) ((DFTYPE SYMBOL)))
  (DECLARE ((IRLIST . CCMPIL) FLUIC)
   ((APLIST . CCMPIL) FLUIC) ((STYPE . COMPIL) FLUID))
  (DECLARE (EXITERS SYMBCL OWN)
   (MSGFILE SYMBCL CWN)
   (INFILE FLUIC)
   (CUTFILE FLLIC)
   (FORMAT FLUIC)
   (INDEV FLUIC)
   (KEEP SYMBCL FLLID)
   (PASS INTEGER FLUIC)
   (ERRFLG BCOLEAN FLUIC) (QUOTARGS SYMBOL FLUID))
  (DECLARE ((INTERACT . LISP) BOOLEAN FLUID)
   ((OTTY . SYS) SYMBOL CWN) ((ITTY . SYS) SYMBOL OWN))
  (FLNCTICN ((LCEXP . LCCMP) SYMBOL) ((EXP SYMBOL FREE) (T SYMBOL)))
  (FLNCTICN SLREAD NIL)
  (SET MSGFILE (QUCTE (CTTY)))
  (SET EXITERS (LIST (QUOTE STOP)
    (QUOTE END) (QLOTE (STCP)) (OCT2CH 34Q) (OCT2CH 31Q)))
  (SET PRNMAX 15) (SET PRNERR FALSE) (SET BINLAP TRUE))
(SLPERVISOR (SECTICN (SLPV COMPIL LCCMP SYS))
  (FLNCTICN (LISP . LISP)
   ((INFILE FLLIC) (OUTFILE FLUID) (FORMAT FLUID))
   (IF (NCT CNEPASS)
    (BLOCK ((KEEP FLUID) ((GNLIST . SYS) FLUID))
     (RETURN ((LISP . SUPV)))) ((LISP . SUPV))))
  (FLNCTICN (LISP . SUPV)
   NIL (BLCCK (((STYPE . CCMPIL) (QUOTE SYMBOL))
     ((LISPVALUE . LISP) NIL)
     ((SLIST . CCMPIL) (QUCTE (USER LISP)))
     (INDEV FLLIC (DEVTYPE INFILE)) R)
    (BLOCK (((SNAME . CCMPIL) (CAR (SLIST . COMPIL))))
     (MESSAGE (QUCTE LISPENTRY))
     X (IF (MEMBER (SET R (SREAD)) EXITERS) (GO OUT))
     (MESSAGE (BLOCK NIL (IF (EQ FCRMAT (QUOTE IL))
        (SET R (LIST (QUOTE DUMMY) R)))
      (SET R (IF CNEPASS (ED1SUP R) (ED2SUP R)))
     (RETURN (IF (EQ FORMAT (QUOTE IL)) (CADR R) R))))
   (GO X) CUT (MESSAGE (QUOTE LISPEXIT)) (RETURN LISPVALUE))))
(FUNCTION SREAD NIL (BLOCK ((IN (INPUT INFILE)) R)
  A (BLOCK (((INTERACT . LISP) FALSE) ((PRNERR . LISP) FALSE))
   (TRY R ERR (BLOCK NIL (IF PARENFLAG (ENDIN))
     (SET R (IF (EQ FORMAT (QUOTE SL)) (SLREAD) (READ))))))
  (INPUT IN)
  (IF PRNIL (SPRINT R))
  (RETURN R)
  ERR (IF (NQ INDEV (QUOTE TTY))
   (BLOCK NIL (IF PRNERR (MESSAGE R)) (INPUT IN) (EXIT R)))
  (MESSAGE R)
  (MESSAGE (QUOTE (*STRING 'B 'A 'D '  'R 'E 'A 'D ', '  'T 'R 'Y '
     'A 'G 'A 'I 'N '.))) (GO A)))
(FUNCTION SPRINT (X)
 (BLOCK ((OUT (OUTPUT OUTFILE)))
  (PRETTYP X) (OUTPUT OUT) (RETURN X)))
(FUNCTION ED1SUP (J)
 (IF (NOT (LISTP J))
  (CONS J (QUOTE (NOT ED FORMAT)))
  (BLOCK ((PASS 1))
   (RETURN (CONS (CAR J) (MAPCAR (CDR J) OPERATE))))))
(FUNCTION ED2SUP (J)
 (IF (NOT (LISTP J))
  (CONS J (QUOTE (NOT ED FORMAT)))
  (BLOCK NIL (IF (GR (LENGTH J) 2)
    (BLOCK (((SNAME . COMPIL) SNAME)
      ((SLIST . COMPIL) SLIST) ((STYPE . COMPIL) STYPE))
     (ED1SUP J)))
   (BLOCK ((PASS 2))
    (RETURN (CONS (CAR J) (MAPCAR (CDR J) OPERATE)))))))
(FUNCTION OPERATE (X)
 (BLOCK ((ERRFLG FLUID FALSE) Z)
  RESTART (IF (EQ X (QUOTE EXIT))
   (EXIT LISPVALUE)
   (SET Z (FIND X QUOTARGS))
   (BLOCK (((PARENFLAG . LISP) FALSE))
    (TRY Z EVALER (RETURN (EVALQUOTE (CDR Z)))))
   (NOT (LISTP X))
   (GO EVAL)
   (OR (EQ (CAR X) (QUOTE FUNCTION)) (EQ (CAR X) (QUOTE ROUTINE)))
   (IF (SIM (QUOTE ((CR. V. (V. ID.)) L. (ANY. S.))) (CDR X))
    (BLOCK NIL (SET Z (IF (OR (EQ PASS 1) (NULL (CDDDR X)))
       (FNDEC X) (COMPILER X))) (GO TESTEM)))
   (OR (EQ (CAR X) (QUOTE MACRO))
    (EQ (CAR X) (QUOTE INSTRUCTIONS)))
   (BLOCK NIL (SET Z (FNAID X)) (GO TESTEM))
   (EQ (CAR X) (QUOTE DECLARE))
   (BLOCK NIL (SET Z (MAPCAR (CDR X) DECL1)) (GO TESTEM))
   (EQ (CAR X) (QUOTE SECTION))
   (IF (SIM (QUOTE ((CR. ID. ((C. 0 10000 ID.))) (ANY. ID.)))
     (CDR X))
    (BLOCK NIL (SET Z (SECSET (CADR X)
       (IF (CDDR X) (CADDR X) (QUOTE SYMBOL)))) (GO TESTEM)))
   (EQ (CAR X) (QUOTE DEFAULT))
   (IF (SIM (QUOTE (ID.)) (CDR X))
    (BLOCK NIL (SET Z (DEFAULT (CADR X))) (GO TESTEM)))
   (EQ (CAR X) (QUOTE LAP))
   (IF (SIM (QUOTE ((ID. (V. ID.)
        L. (C. 0 10000 (CR. ID. L.))) L. ID.)) (CDR X))
    (BLOCK NIL (SET Z (IF (EQ PASS 2) (LAPP X) (CAADADR X)))
     (GO TESTEM))) (LABEL EVAL (TRY Z EVALER (RETURN (EVAL X)))))
  (COMER2 X (QUOTE (SYNTAX ERROR)))
  TESTEM (IF ERRFLG (GO EVALER))
  RET (RETURN Z)
  EVALER (IF (OR (NOT INTERACT) (EQ INDEV (QUOTE TTY))) (GO RET))
  (SET ERRFLG FALSE)
  (BLOCK (((BADEXP . LISP) X))
   (IF (NOT ((LISP . LISP) ITTY CTTY (QUOTE IL))) (GO RET))
   (SET X BADEXP) (GO RESTART))))
(FUNCTION EVALQUOTE (FNAME)
 (BLOCK ((FN (FUNCTIONAL SYMBOL FUNCTIONAL) (CDR FNAME)))
  (RETURN (FN (CAR FNAME)))))
(FUNCTION LAPP (L)
 (IF (NOT BINLAP)
  (BLOCK NIL (IF PRNLAP (SPRINT L)))
  (BLOCK ((X (LAP (CADR L) (CADDR L) (CADDDR L))))
   (IF ERRFLG (RETURN (LIST X (QUOTE BAD))) PRNLAP (SPRINT L))
   (IF (EQ (QUOTE RUN) (CDR X))
    (BLOCK ((V (EVAL (LIST X))))
     (EXCISE (CAR X) (CDR X)) (RETURN V)) (RETURN X)))))
(FUNCTION KEEPER (X)
 (IF (MEMBER X KEEP) KEEP (SET KEEP (CONS X KEEP))))
(FUNCTION FNAID (X)
 (BLOCK NIL (IF (NOT (OR (SIM (QUOTE (INSTRUCTIONS (OR. V. (V.
         NOVALUE)) NIL S.)) X)
     (SIM (QUOTE (MACRO (OR. V. (V. SYMBOL)) (S.) S.)) X)))
   (GO HACK))
  (BLOCK ((Z (CADR X)) (MFLAG BOOLEAN (EQ (CAR X) (QUOTE MACRO))))
   (IF (EQ PASS 1)
    (RETURN Z)
    (AND MFLAG (BLOCK ((Y (CADDR X)))
      (IF (NQ (LENGTH Y) 1)
       (RETURN TRUE)
       (SIM (QUOTE V.) (CAR Y))
       (SET (CAADDR X) (LIST (CAR Y) (QUOTE SYMBOL)))
       (SIM (QUOTE (V. SYMBOL . S.)) (CAR Y))
       (RETURN NIL)
       (SET (CDAADDR X) (CONS (QUOTE SYMBOL) (CDAR Y))))))
    (GO HACK)
    (BLOCK NIL (IF (SIM (QUOTE V.) (CADR X))
      (SET (CADR X)
       (LIST (CADR X) (IF MFLAG (QUOTE SYMBOL) (QUOTE NOVALUE)))))
     (RETURN (COMPILER X)))))) HACK (COMER2 X (QUOTE (HACK DEF)))))
(FUNCTION COMPILER (X)
 (BLOCK ((IRLIST . COMPIL) (APLIST . COMPIL))
  (BLOCK ((F (FUNCTIO X)))
   (RETURN (IF ERRFLG (LIST (CADR X) (QUOTE BAD)) (LAPP F))))))
(FUNCTION (COMPILER . LISP)
 (J) (BLOCK ((PASS 2)) (RETURN ((COMPILER . SUPV) J))))
(FUNCTION (EVAL . LISP)
 (S D E)
 (BLOCK ((SNAME . COMPIL)
   (SLIST . COMPIL)
   (STYPE . COMPIL) (PASS 2) ((ERRFLG . SUPV) FALSE))
  (OPERATE (LIST (QUOTE SECTION) S D))
  (IF ERRFLG (RETURN NIL)) (RETURN ((EVAL . SUPV) E))))
(FUNCTION ((EVAL . SUPV) SYMBOL)
 ((J SYMBOL))
 (BLOCK ((V SYMBOL (CONS (GENID) (QUOTE RUN))) (L SYMBOL))
  (IF (EQ (PASS . SUPV) 1) (RETURN NIL) (NOT TRYLC) (GO BC))
  LC (BLOCK (((LISTING . LCOMP) SYMBOL FREE))
   (TRY L BCP (LCEXP J (QUOTE SYMBOL)))
   (SET L (SUBST (DREVERSE (APPEND (QUOTE ((RETURN) (END)))
       (LISTING . LCOMP)))
     (QUOTE E)
     (SUBST V (QUOTE V)
      (QUOTE (LAP (FUNCTION (V SYMBOL)
         NIL (BEGIN) . E) NIL RUN))))))
  (GO RUN)
  BCP (IF (EQ TRYLC (QUOTE ONLY)) (GO ERROR))
  BC (SET L ((FUNCTIC . COMPIL)
    (LIST (QUOTE FUNCTION) (LIST V (QUOTE SYMBOL)) NIL J)))
  RUN (IF (ERRFLG . SUPV) (GO ERROR))
  (LAP (CADR L) (CADDR L) (CADDDR L))
  (IF (ERRFLG . SUPV) (GO ERROR))
  (BLOCK ((G (FUNCTIONAL SYMBOL) (NAME2FUNC V))
    ((PRNERR . LISP) FALSE)) (TRY L EXERR (SET L (G))))
  (EXCISE (CAR V) (CDR V))
  (RETURN L)
  EXERR (MESSAGE (LIST J (QUOTE (*STRING '  'E 'V 'A 'L '  'E 'X 'I
     'T '  'V 'A 'L 'U 'E '. '. '. ' )) L))
  (MESSAGE (LIST (QUOTE (*STRING 'B 'A 'C 'T 'R 'A 'C 'E '. '. '. '
     )) BACKTRACE))
  (EXCISE (CAR V) (CDR V))
  ERROR (EXIT (QUOTE (*STRING 'B 'A 'D '  'E 'V 'A 'L '.)))))
 (FUNCTION EXCISE (N S)
  (BLOCK NIL (DELE (GETFREE N S) KEEP)
   (RETURN ((EXCISE . LISP) N S)))))
****END OF FILE DETECTED
```

```
(LCOMP (SECTION (LCOMP SYS) SYMBOL)
  (DECLARE (LISTING SYMBOL FREE)
   (EXP SYMBOL FREE)
   (INV SYMBOL FREE)
   ((TRYLC . LISP) SYMBOL FREE TRUE)
   ((SLIST . COMPIL) SYMBOL FREE)
   ((STYPE . COMPIL) SYMBOL FREE)
   ((ERRFLG . SUPV) BOOLEAN FREE)
   ((PASS . SUPV) INTEGER FREE)
   ((PRNERR . LISP) BOOLEAN FREE))
  MACRO1 (((NAME2FUNC (LAMBDA (L)
      (SUBST (CADR L)
       (QUOTE V)
       (QUOTE (CHEAT INTEGER FUNCTIONAL (PLUS (S20. (GETFREE (CAR V)
            (CDR V))) 17777777Q)))))))
  (FUNCTION (LCEXP SYMBOL)
   ((EXP SYMBOL FREE) (T SYMBOL)) (LCCONV (LCE EXP) T))
  (FUNCTION (LCE SYMBOL)
   ((EXP SYMBOL FREE))
   (BLOCK ((X SYMBOL))
    (IF (AND (ATOM EXP) (NOT (IDP EXP)))
     (RETURN (LCE (QUOTER EXP))) (LISTP EXP) (RETURN (LCFORM)))
    (SET X (LCVAR EXP FALSE))
    (IF (NULL X) (LCEXIT))
    (ATTACH (CONS (QUOTE LDA) (CDR X))) (RETURN (CAR X))))
  (FUNCTION (LCFORM SYMBOL)
   NIL (BLOCK ((V SYMBOL (CAR EXP))
     (D SYMBOL)
     (C SYMBOL) (X SYMBOL) (L SYMBOL) (A SYMBOL) (T SYMBOL))
    (IF (NULL (SET D (LCFREE V))) (LCEXIT))
    (IF (EQ (CAR D) (QUOTE MACRO))
     (BLOCK ((G (FUNCTIONAL SYMBOL SYMBOL) (NAME2FUNC V)))
      (RETURN (LCE (G EXP))))
     (EQ (CAR D) (QUOTE INSTRUCTIONS))
     (IF (NQ (CDR V) (QUOTE LLISP))
      (LCEXIT)
      (BLOCK ((G (FUNCTIONAL SYMBOL) (NAME2FUNC V))
        (INV SYMBOL FREE)) (G) (RETURN INV)))
     (OR (EQ (CAR D) (QUOTE FUNCTION)) (EQ (CAR D) (QUOTE ROUTINE)))
     (SET C (LIST (LIST (QUOTE CALL) V)))
     (EQ (CAR (SET X (LCVAR V FALSE))) (QUOTE FUNCTIONAL))
     (SET C (SUBST (CDR X)
       (QUOTE A)
       (QUOTE ((LDB . A)
         (STB (FMCALL . SYS)) (CALL (FMCALL . SYS))))))
     (LCEXIT))
    CALL (ATTACH (QUOTE (ARGS)))
    (SET X (CDADR D))
    (LCNELS (LENGTH X))
    (IF (NULL (SET L (CDR X))) (GO F2))
    (SET A (CDR EXP))
    F1 (SET T (IF (EQ (CADAR L) (QUOTE LOC))
      (BLOCK NIL (LCLOC (CAR A) (CAAR L)) (RETURN (QUOTE SYMBOL)))
      (LCEXP (CAR A) (CAAR L))))
    (SET A (CDR A))
    (IF (SET L (CDR L)) (BLOCK NIL (LCPUSH T) (GO F1)))
    F2 (ATTSEQ C)
    (SET T (CAR X))
    (RETURN (IF (NQ T (QUOTE NOVALUE))
      T (BLOCK NIL (ATTACH (QUOTE (STZ A.)))
       (RETURN (QUOTE SYMBOL))))))
  (INSTRUCTIONS ((QUOTE . LLISP) NOVALUE)
   NIL (BLOCK NIL (LCNELS 2)
  (ATTACH (LIST (QUOTE LDA) (QUOTER (CADR EXP))))
  (SET INV (QUOTE SYMBOL))))
(INSTRUCTIONS ((SET . LLISP) NOVALUE)
 NIL (BLOCK ((X SYMBOL) (T SYMBOL))
  (LCNELS 3)
  (SET X (LCVAR (CADR EXP) FALSE))
  (IF (NULL X) (LCEXIT))
  (SET T (LCE (CADDR EXP)))
  (IF (NQ T (CAR X)) (LCPUSH T))
  (LCCONV T (CAR X))
  (ATTACH (CONS (QUOTE STF) (CDR X)))
  (IF (NQ T (CAR X)) (ATTACH (QUOTE (LDA POP.)))) (SET INV T)))
(FUNCTION ((CAR . LLISP) SYMBOL) ((X SYMBOL)) (CAR X))
(FUNCTION ((CDR . LLISP) SYMBOL) ((X SYMBOL)) (CDR X))
(FUNCTION (LCLOC SYMBOL)
 ((E SYMBOL) (T SYMBOL))
 (BLOCK ((D SYMBOL (LCVAR E TRUE)))
  (IF (OR (NULL D) (NQ (CAR D) T)) (LCEXIT))
  (ATTACH (CONS (QUOTE LDA) (CDR D)))))
(FUNCTION (LCVAR SYMBOL)
 ((V SYMBOL) (MLOC BOOLEAN))
 (BLOCK ((D SYMBOL (LCFREE V)) (K SYMBOL) (A SYMBOL))
  (IF (NULL D) (RETURN NIL))
  (SET K (CAR D))
  (IF (AND (EQ K (QUOTE OWN))
    (NQ (CADDR D) (QUOTE LOC)) (NOT (SET MLOC (NOT MLOC))))
   (SET A (QUOTE (R L4567.7)))
   (AND (EQ K (QUOTE FUNCTION)) (NOT MLOC))
   (SET A (QUOTE (2Q1 R L4567.7)))
   (MEMBERN K (QUOTE (OWN FLUID FREE)))
   (SET A (IF MLOC 0 (QUOTE I))) (RETURN NIL))
  (RETURN (LIST (FTYPER (CADR D)) V A))))
(FUNCTION (LCFREE SYMBOL)
 ((V SYMBOL LOC))
 (BLOCK ((X SYMBOL) (N SYMBOL V) (S SYMBOL (SLIST . COMPIL)))
  (IF (ATOM V)
   (IF (NOT (IDP V)) (RETURN NIL))
   (BLOCK NIL (SET N (CAR V)) (SET S (LIST (CDR V)))))
  (FOR S (IN S)
   (IF (OR (AND (EQ S (QUOTE LISP))
      (SET X (GETFREE N (QUOTE LLISP)))) (SET X (GETFREE N S)))
    (BLOCK NIL (SET V (VARNAME X))
     (SET X (FVLIST X))
     (IF (NQ (CAR X) (QUOTE MEANS)) (RETURN X))
     (SET V (CONS (CADR X) (CADDR X))) (RETURN (LCFREE V)))))))
(FUNCTION (LCCONV SYMBOL)
 ((T1 SYMBOL) (T2 SYMBOL))
 (BLOCK ((X SYMBOL))
  (IF (EQN T1 T2) (GO R))
  (IF (SET X (FINDN T1 (QUOTE ((OCTAL . OCT2SYM)
       (INTEGER . INT2SYM)
       (REAL . REAL2SYM) (FUNCTIONAL . FORM2SYM)))))
   (ATTCON (CDR X)))
  (IF (EQ T2 (QUOTE BOOLEAN))
   (ATTACH (QUOTE (BUC (ENTRY STBENT) 0 4)))
   (SET X (FINDN T2 (QUOTE ((OCTAL . SYM2OCT)
       (INTEGER . SYM2INT)
       (REAL . SYM2REAL) (FUNCTIONAL . SYM2FORM)))))
   (ATTCON (CDR X))) R (RETURN T2)))
(FUNCTION (LCNELS SYMBOL)
 ((N INTEGER))
 (IF (AND (LISTP EXP) (EQ (LENGTH EXP) N)) TRUE (LCEXIT)))
(FUNCTION (LCEXIT SYMBOL) NIL (EXIT NIL))
```

```
  (FUNCTION (LOPUSH SYMBOL)                                                    0012700
   ((T SYMBOL))                                                                0012800
   (ATTACH (LIST (QUOTE STF)                                                   0012900
     (IF (OR (EQ T (QUOTE SYMBOL)) (EQ T (QUOTE FUNCTIONAL)))                  0013000
      (QUOTE PUSHP.) (QUOTE PUSHA.)))))                                        0013100
  (FUNCTION (ATTCON SYMBOL)                                                    0013200
   ((V SYMBOL))                                                                0013300
   (BLOCK NIL (ATTACH (QUOTE (ARGS)))                                          0013400
    (ATTACH (LIST (QUOTE CALL) (CONS V (QUOTE LISP))))))                       0013500
  (FUNCTION (ATTSEQ SYMBOL)                                                    0013600
   ((X SYMBOL)) (BLOCK NIL (FOR X (IN X) (ATTACH X))))                         0013700
  (FUNCTION (ATTACH SYMBOL)                                                    0013800
   ((I SYMBOL)) (BLOCK NIL (SET LISTING (CONS I LISTING))))                    0013900
  (FUNCTION (QUOTER SYMBOL) ((X SYMBOL)) (LIST (QUOTE QUOTE) X)))              0014000
****END OF FILE DETECTED
```

```
(COMETA (SECTION (COMETA LISP COMPIL SUPV SYS) SYMBOL)
  (DECLARE (COMLST SYMBOL FLUID)
   ((PRNIL . LISP) BOOLEAN FREE FALSE))
  (FUNCTION (COMETA SYMBOL)
   (X)
   (BLOCK NIL (IF (PRNIL . LISP) (SPRINT X) (SUPOTY (CADR X)))
    (RETURN (IF (EQ (CAR X) (QUOTE SECTION)) (SECSET X) (COMP X)))))
  (FUNCTION (SECSET SYMBOL)
   (S)
   (BLOCK NIL (SET (STYPE . COMPIL) (CADDR S))
    (IF (ATOM (SET (SLIST . COMPIL) (CADR S)))
     (SET (SLIST . COMPIL) (LIST (SLIST . COMPIL))))
    (IF (NOT (MEMBER (QUOTE LISP) (SLIST . COMPIL)))
     (SET (SLIST . COMPIL)
      (APPEND (SLIST . COMPIL) (QUOTE (LISP)))))
    (RETURN (SET (SNAME . COMPIL) (CAR (SLIST . COMPIL))))))
  (FUNCTION (COMP SYMBOL)
   (F)
   (BLOCK ((VCLASS . COMPIL)
     (VADDR . COMPIL)
     ((LISTING . COMPIL) (LIST (QUOTE FUNCTION)))
     (REFLIST . COMPIL) (TGO . COMPIL) (FGO . COMPIL))
    (MAKEFREE (CADR F)
     (SNAME . COMPIL)
     (QUOTE FUNCTION) (QUOTE (FUNCTIONAL SYMBOL)) (QUOTE VALUE))
    (BLOCK ((N (CONS (CADR F) (SNAME . COMPIL))))
     (ATTACH (LIST N (QUOTE SYMBOL)))
     (ATTACH NIL)
     (ATTACH (QUOTE (BEGIN)))
     (COMVAL (CADDDR F))
     (ATTACH (QUOTE (END)))
     (ATTACH (QUOTE (RETURN)))
     (SET (LISTING . COMPIL) (DREVERSE (LISTING . COMPIL)))
     (IF (PRNLAP . LISP)
      (SPRINT (LIST (QUOTE LAP)
        (LISTING . COMPIL) (REFLIST . COMPIL) (SNAME . COMPIL))))
     (IF (BINLAP . LISP)
      (LAP (LISTING . COMPIL) (REFLIST . COMPIL) (SNAME . COMPIL)))
     (RETURN N))))
  (FUNCTION (ATTACHG SYMBOL)
   (L) (ATTACH (LIST (QUOTE BUC) (LABELER L))))
  (FUNCTION (LABELER SYMBOL) (L) (LIST (QUOTE LABEL) L))
  (FUNCTION ATTACHL (L)
   (BLOCK NIL LOOP (SCOTCH L LISTING)
    (IF (NOT (REFLAB L LISTING)) (RETURN LISTING))
    (BLOCK ((K LISTING))
     A (IF (ATOM (CAR K)) (BLOCK NIL (SET K (CDR K)) (GO A)))
     (IF (AND (EQ (CAAR K) (QUOTE BUC))
       (NOT (ATOM (CADR K)))) (GOHERE L (CADR K)))
      (BLOCK NIL (SET (CAAR K)
        (CDR (FINDN (CAADR K) (QUOTE ((BOZ . BNZ) (BNZ . BOZ))))))
       (SET (CDR K) (CDDR K)) (GO LOOP))))
    (RETURN (SET LISTING (CONS L LISTING)))))
  (FUNCTION (ATTACH SYMBOL)
   (I)
   (BLOCK NIL (IF (ATOM I)
     (SCOTCH I (LISTING . COMPIL))
     (AND (NOT (ATOM (CAR (LISTING . COMPIL))))
      (MEMBER (CAAR LISTING) (QUOTE (BUC BSX)))
      (NOT (MEMBER I (QUOTE ((ARGS) (END) (CALL)))))))
     (RETURN (LISTING . COMPIL)))
    (RETURN (SET (LISTING . COMPIL) (CONS I (LISTING . COMPIL))))))
  (FUNCTION (SCOTCH BOOLEAN)
 (L LST)
 (IF (ATOM (CAR LST))
  (BLOCK ((B BOOLEAN (SCOTCH L (CDR LST))))
   (IF (NOT B)
    (RETURN FALSE) (SCOTCH (CAR LST) (CDR LST)) (SCOTCH L LST))
   (RETURN TRUE))
  (BLOCK NIL (IF (NOT (GOHERE L (CAR LST))) (RETURN FALSE))
   (SET (CAR LST) (CADR LST))
   (SET (CDR LST) (CDDR LST)) (RETURN TRUE))))
(FUNCTION (REFLAB BOOLEAN)
 (L LST)
 (BLOCK NIL (FOR LST (IN LST)
   (UNLESS (OR (ATOM LST)
     (NOT (OR (GOHERE L LST)
       (AND (EQ (CAR LST) (QUOTE BSX)) (EQ L (CADADDDR LST)))))))
   (RETURN TRUE))))
(FUNCTION (GOHERE BOOLEAN)
 (L I)
 (AND (MEMBER (CAR I) (QUOTE (BUC BOZ BNZ))) (EQN L (CADADR I))))
(FUNCTION (COMIF NOVALUE)
 NIL (BLOCK ((X (CDR EXP))
   (S (XGO . COMPIL))
   ((XGO . COMPIL) (IF (XGO . COMPIL) (XGO . COMPIL) (GENID))))
  (IF (NULL X)
   (GO M) (NULL (CDR X)) (BLOCK NIL (COMSTAT (CAR X)) (GO M)))
  L (BLOCK ((FG (GENID)))
   (BLOCK ((F (IF (NULL (CDDR X))
       (XGO . COMPIL) (CDDDR X) FG (BGO (CADDR X) FG)))
     (T (BGO (CADR X) NIL)))
    (BLOCK ((TERGO . COMPIL) ((PCLASS . COMPIL) TRUE))
     (COMBOL (CAR X) T F))
    (IF (NULL T)
     (BLOCK NIL (COMSTAT (CADR X))
      (IF (NOT (OR (RETP (CADR X)) (AND (NULL (CDDR X)) (NOT S))))
       (ATTACHG XGO))))
    (ATTACHL FG)
    (IF (NULL (SET X (CDDR X)))
     (GO M)
     (NOT (NULL (CDR X))) (GO L) (EQ F FG) (COMSTAT (CAR X)))))
  M (IF (NULL S) (ATTACHL (XGO . COMPIL)))))
(FUNCTION (RETP BOOLEAN)
 (I) (AND (NOT (ATOM I)) (EQ (CAR I) (QUOTE RETURN))))
(FUNCTION (BGO SYMBOL)
 (I L)
 (IF (ATOM I)
  L (EQ (CAR I) (QUOTE GO))
  (CADR I)
  (AND (PCLASS . COMPIL) (RETP I))
  (IF (NULL (CADR I))
   (FGO . COMPIL) (EQ TRUE (CADR I)) (TGO . COMPIL) L) L))
(FUNCTION (COMOR NOVALUE) NIL (COMLOG FALSE))
(FUNCTION (COMAND NOVALUE) NIL (COMLOG TRUE))
(FUNCTION (COMNOT NOVALUE)
 NIL (IF (OR (SCLASS . COMPIL) (NOT (PCLASS . COMPIL)))
  (MAKEPRED)
  (COMBOL (CADR (EXP . COMPIL)) (FGO . COMPIL) (TGO . COMPIL))))
(FUNCTION (MAKEPRED NOVALUE)
 NIL (IF SCLASS NIL (BLOCK ((TER TERGO)
    ((TERGO . COMPIL) (IF TERGO TERGO (GENID)))
    (T (GENID)) (F (GENID)) ((PCLASS . COMPIL) TRUE))
   (COMBOL EXP T F)
   (BLOCK ((L (LABELER TERGO)))
    (BLOCK ((T1 (LIST (QUOTE BSX) (QUOTE (ENTRY ONENT)) 4 L))
      (F1 (LIST (QUOTE BSX) (QUOTE (ENTRY STZENT)) 4 L)))
     (IF (SCOTCH T LISTING)
      (BLOCK NIL (ATTACHL T) (ATTACH T1) (ATTACHL F) (ATTACH F1))
      (BLOCK NIL (ATTACHL F) (ATTACH F1) (ATTACHL T) (ATTACH T1)))
     (IF (NOT TER) (ATTACHL TERGO))
     (SET VCLASS (QUOTE ACTIVE)))))))
(FUNCTION (COMLOG NOVALUE)
 ((B BOOLEAN))
 (IF (NULL (CDR (EXP . COMPIL)))
  (COMPILE B)
  (OR (SCLASS . COMPIL) (NOT (PCLASS . COMPIL)))
  (MAKEPRED)
  (BLOCK (T (L (GENID)))
   (FOR T (ON (CDR (EXP . COMPIL)))
    (COMBOL (CAR T)
     (IF (NULL (CDR T))
      (TGO . COMPIL) B NIL (TGO . COMPIL) (TGO . COMPIL) L)
     (IF (NULL (CDR T))
      (FGO . COMPIL)
      (NULL B) NIL (FGO . COMPIL) (FGO . COMPIL) L)))
   (ATTACHL L))))
(FUNCTION (COMBOL NOVALUE)
 (X (TGO . COMPIL) (FGO . COMPIL))
 (BLOCK NIL (BLOCK ((VCLASS . COMPIL) (VADDR . COMPIL))
   (COMEXP X)
   (IF (NQ VCLASS (QUOTE PRED))
    (BLOCK NIL (MOVAC)
     (IF TGO (ATTACH (LIST (QUOTE BNZ) (LABELER TGO))))
     (IF FGO (ATTACH (LIST (IF TGO (QUOTE BUC) (QUOTE BOZ))
        (LABELER FGO))))))) (SET VCLASS (QUOTE PRED))))
(FUNCTION (MOVAC NOVALUE)
 NIL (IF (NQ (VCLASS . COMPIL) (QUOTE ACTIVE))
  (BLOCK NIL (ATTACH (CONS (QUOTE LDA) (VADDR . COMPIL)))
   (SET (VCLASS . COMPIL) (QUOTE ACTIVE)))))
(FUNCTION (COMGO NOVALUE) NIL (ATTACHG (CADR (EXP . COMPIL))))
(FUNCTION (COMQUO NOVALUE)
 NIL (BLOCK NIL (ATTACH (IF (IDP (CADR (EXP . COMPIL)))
    (SUBST (CADR EXP) (QUOTE I) (QUOTE (LDA (ID I) (R L4567.7))))
    (LIST (QUOTE LDA) (EXP . COMPIL))))
  (SET (VCLASS . COMPIL) (QUOTE ACTIVE))))
(FUNCTION (COMSET NOVALUE)
 NIL (BLOCK NIL (COMEXP (CADR (EXP . COMPIL)))
  (IF (AND (NULL (CADDR (EXP . COMPIL)))
    (OR (SCLASS . COMPIL) (PCLASS . COMPIL)))
   (BLOCK NIL (ATTACH (CONS (QUOTE STZ) (VADDR . COMPIL)))
    (IF (NOT (SCLASS . COMPIL))
     (BLOCK NIL (IF (FGO . COMPIL) (ATTACHG (FGO . COMPIL)))
      (SET (VCLASS . COMPIL) (QUOTE PRED)))))
   (BLOCK ((L (VADDR . COMPIL)))
    (COMVAL (CADDR (EXP . COMPIL)))
    (ATTACH (CONS (QUOTE STF) L))
    (SET (VCLASS . COMPIL) (QUOTE ACTIVE))))))
(FUNCTION (COMRET NOVALUE)
 NIL (IF (PCLASS . COMPIL)
  (COMBOL (CADR (EXP . COMPIL)) (TGO . COMPIL) (FGO . COMPIL))
  (BLOCK NIL (COMACT (CADR (EXP . COMPIL)))
   (ATTACHG (TERGO . COMPIL)))))
(FUNCTION (COMBLK NOVALUE)
 NIL (BLOCK ((XGO . COMPIL))
  (IF (TERGO . COMPIL)
   (BLOCK (T) (FOR T (IN (CDDR (EXP . COMPIL))) (COMSTAT T)))
   (BLOCK (((TERGO . COMPIL) (GENID)))
    (BLOCK (((TGO . COMPIL)
       (IF (TGC . CCMPIL) (TGO . CCMPIL) (TERGO . COMPIL)))
      ((FGC . COMPIL)
       (IF (FGC . CCMPIL) (FGO . CCMPIL) (TERGO . COMPIL))))
     (CCMBLK)
     (IF (PCLASS . CCMPIL)
      (IF (NCT (EQN (FGO . COMPIL) (TERGO . COMPIL)))
       (ATTACHG (FGC . CCMPIL))) (COMVAL NIL))
     (ATTACHL (TERGC . CCMPIL)))))
  (SET (VCLASS . CCMPIL)
   (IF (PCLASS . COMPIL) (QUOTE PRED) (QUOTE ACTIVE)))))
(FLNCTICN (CCMSTAT NCVALUE)
 (X)
 (IF (IDP X)
  (ATTACH X)
  (BLOCK (((SCLASS . COMPIL) TRUE)
    (VADDR . CCMPIL) (VCLASS . CCMPIL)) (COMPILE X))))
(FLNCTICN (CCMVAL NCVALUE)
 (X) (BLCCK ((TERGO . COMPIL) (PCLASS . COMPIL)) (COMACT X)))
(FLNCTICN (CCMACT NOVALUE)
 (X)
 (BLOCK NIL (CCMEXP X)
  (IF (NQ (VCLASS . CCMPIL) (QUOTE ACTIVE)) (MOVAC))))
(FLNCTICN (CCMEXP NOVALUE)
 (X) (BLCCK ((SCLASS . CCMPIL)) (CCMPILE X)))
(FLNCTICN (CCMPILE NCVALUE)
 ((EXP . CCMPIL))
 (IF (IDP (EXP . COMPIL))
  (COMVAR)
  (ATOM (EXP . CCMPIL))
  (COMDAT)
  (BLOCK ((X (FIND (CAR (EXP . CCMPIL)) COMLST)))
   (IF X (BLCCK ((F (FUNCTIONAL NCVALUE) (CDR X))) (F))
    (COMCAL)))))
(FLNCTICN (CCMDAT NOVALUE)
 NIL (IF (PCLASS . CCMPIL)
  (BLOCK NIL (IF (NULL (EXP . COMPIL))
    (IF (FGO . CCMPIL) (ATTACHG (FGO . COMPIL)))
    (IF (TGO . CCMPIL) (ATTACHG (TGO . CCMPIL))))
   (SET (VCLASS . COMPIL) (QUOTE PREC)))
  (BLOCK NIL (ATTACH (IF (NULL (EXP . COMPIL))
     (QUCTE (STZ A.))
     (EQN (EXP . CCMPIL) TRUE)
     (QUCTE (LDA 1 (R L567.7)))
     (LIST (QUCTE LDA) (LIST (QUCTE QUOTE) (EXP . COMPIL)))))
   (SET (VCLASS . CCMPIL) (QUOTE ACTIVE)))))
(FLNCTICN (CCMVAR NCVALUE)
 NIL (BLCCK ((D (FINDEC (EXP . CCMPIL) FALSE)))
  (SET (VADDR . COMPIL) (LIST D (QUOTE I)))
  (SET (VCLASS . COMPIL) (QUOTE LCC))))
(FLNCTICN (CCMCAL NCVALUE) NIL (CCMCALL (FINDEC (CAR EXP) EXP)))
(FLNCTICN (CCMCALL NCVALUE)
 (D)
 (BLOCK (T)
  (ATTACH (QUOTE (ARGS)))
  (FOR T (CN (CDR EXP))
   (BLOCK NIL (CCMVAL (CAR T))
    (IF (NCT (NULL (CDR T))) (ATTACH (QUOTE (STF PUSHP.))))))
  (ATTACH (LIST (QUOTE CALL) D)) (SET VCLASS (QUOTE ACTIVE))))
(FLNCTICN (FINCEC SYMBCL)
 (VAR X)
 (BLOCK (S)
  (FOR S (IN (SLIST . COMPIL))
   (BLOCK ((D (SRDEC VAR S))) (IF D (RETURN D))))
  (MAKEFREE VAR (SNAME . COMPIL)
   (IF (NULL X) (QUOTE FLUID) (QUOTE FUNCTION))
   (IF (NULL X)
    (QUOTE SYMBOL)
    (CONS (QUOTE FUNCTIONAL) (QUOTE SYMBOL) (MAKSYM (CDR X))))
   (QUOTE VALUE)) (RETURN (FINDEC VAR X))))
(FUNCTION (MAKSYM SYMBOL)
 (L) (IF (NULL L) NIL (CONS (QUOTE SYMBOL) (MAKSYM (CDR L)))))
(FUNCTION (SRDEC SYMBOL)
 (VAR S)
 (BLOCK ((T (GETFREE VAR S)))
  (IF (NULL T) (RETURN FALSE))
  (BLOCK ((D (FVLIST T)))
   (IF (EQ (CAR D) (QUOTE MEANS))
    (RETURN (SRDEC (CADR D) (CADDR D))))
   (BLOCK ((N (CONS VAR S)))
    (IF (NOT (FIND N (REFLIST . COMPIL)))
     (SET (REFLIST . COMPIL)
      (CONS (CONS N D) (REFLIST . COMPIL)))) (RETURN N)))))
(SET COMLST (LIST (CONS (QUOTE CONS) COMCONS)
  (CONS (QUOTE IF) COMIF)
  (CONS (QUOTE OR) COMOR)
  (CONS (QUOTE AND) COMAND)
  (CONS (QUOTE NOT) COMNOT)
  (CONS (QUOTE NULL) COMNOT)
  (CONS (QUOTE GO) COMGO)
  (CONS (QUOTE QUOTE) COMQUO)
  (CONS (QUOTE SET) COMSET)
  (CONS (QUOTE BLOCK) COMBLK)
  (CONS (QUOTE RETURN) COMRET)
  (CONS (QUOTE CAR) COMCAR)
  (CONS (QUOTE CDR) COMCDR)
  (CONS (QUOTE CADR) COMCADR)
  (CONS (QUOTE CDAR) COMCDAR) (CONS (QUOTE CADDR) COMCADDR)))
(FUNCTION (COMCAR NOVALUE) NIL (CARCDR (QUOTE (A))))
(FUNCTION (COMCDR NOVALUE) NIL (CARCDR (QUOTE (D))))
(FUNCTION (COMCADR NOVALUE) NIL (CARCDR (QUOTE (A D))))
(FUNCTION (COMCDAR NOVALUE) NIL (CARCDR (QUOTE (D A))))
(FUNCTION (COMCADDR NOVALUE) NIL (CARCDR (QUOTE (A D D))))
(FUNCTION (CARCDR NOVALUE)
 (L) (BLOCK NIL (COMVAL (CADR EXP)) (COMNODE L)))
(FUNCTION (COMNODE NOVALUE)
 (L)
 (IF (NULL L)
  NIL (BLOCK NIL (COMNODE (CDR L))
   (ATTACH (IF (EQ (CAR L) (QUOTE A))
     (QUOTE (LDA 0 (L7.123 15))) (QUOTE (LDA 0 (L567.7 15))))))))
(FUNCTION (COMCONS NOVALUE) NIL (COMLNK (QUOTE (CONS2 . SYS))))
(FUNCTION (COMLNK NOVALUE)
 (S)
 (COMCALL (BLOCK (((SLIST . COMPIL) (LIST (CDR S)))
    ((SNAME . COMPIL) (CDR S))) (RETURN (FINDEC (CAR S) EXP))))))

****END OF FILE DETECTED
```