20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3

```
REALVAR=        IVAR            (/P1;)

INTVAR=         IVAR            (/P1;)

BOOLVAR=        IVAR            (/P1;)

BOLARV=         ARIDEN          (/P1;)
INTARV=         ARIDEN          (/P1;)
RELARV=         ARIDEN          (/P1;)
FIDEN=          PRIND           (/P1;)
PRIDEN=         PRIND           (/P1;)
XPAR=           FORPAR          (ENI 3 /P1;
                LDA 7 *R
                SAU Ø /F1;
/F2;            SLJ 4 76567
                ØØ Ø Ø)

Ø=              DIGIT           (Ø)
1=              DIGIT           (1)
2=              DIGIT           (2)
3=              DIGIT           (3)
4=              DIGIT           (4)
5=              DIGIT           (5)
6=              DIGIT           (6)
7=              DIGIT           (7)
8=              DIGIT           (8)
9=              DIGIT           (9)
EQUAL=          REQU            (1)
UNEQ=           REQU            (Ø)

DIGIT=          NUMBER          (*F/P1;PERIOD)

NUMBER=         SVAR            (Ø /P1;)

SVAR            EXP             PRIME=          FACTOR                  (/P1;
                STA 6 /F5,*T;
                LDQ 6 /F5,*T;
                LDA /P3;
                SLJ 4 *ZEXP)
SVAR            EXP             SVAR=           FACTOR
                (LDA /P3;
                LDQ /P1;
                SLJ 4 *ZEXP)
SVAR            TIMES           FACTOR=         TERM
                (/P1;
                FMU /P3;)
SVAR            TIMES           SVAR=           TERM
                (LDA /P3;
                FMU /P1;)
SVAR            SLASH           FACTOR=         TERM
                (/P1;
```

```
                STA 6 /F5,*T;
                LDA /P3;
                FDV 6 /F5,*T;)
SVAR            SLASH       SVAR=       TERM
                (LDA /P3;
                FDV /P1;)
SVAR            +           TERM=       TSUM
                (/P1;
                FAD /P3;)
SVAR            +           SVAR=       TSUM
                (LDA /P3;
                FAD /P1;)
SVAR            -           TERM=       TSUM
                (/P1;
                SCM Ø *ZALL7
                FAD /P3;)
SVAR            -           SVAR=       TSUM
                (LDA /P3;
                FSB /P1;)
SVAR            REQU        AREX=       RELATN
                (/P1;
                FSB /P3;
                AJP /P2; *J)
SVAR            REQU        SVAR=       RELATN
                (LDA /P3;
                FSB /P1;
                AJP /P2; *J)
SVAR            RBMA        AREX=       RELATN
                (/P1;
                FSB /P3;
                AJP /P2; *J)
SVAR            RBMA        SVAR=       RELATN
                (LDA /P1;
                FSB /P3;
                AJP /P2; *J)
SVAR            RAMB        AREX=       RELATN
                (/P1;
                SCM Ø *ZALL7
                FAD /P3;
                AJP /P2; *J)
SVAR            RAMB        SVAR=       RELATN
                (LDA /P3;
                FSB /P1;
                AJP /P2; *J)

SVAR=           AXPN
                (LDA /P1;)

FACTOR          EXP         PRIME=      FACTOR
                (/P1;
                STA 6 /F5,*T;
                /P3,*T.*T*I;
                LDQ 6 /F5,*T;
                SLJ 4 *ZEXP)
FACTOR          EXP         SVAR=       FACTOR
                (/P3;
                LDQ /P1;
                SLJ 4 *ZEXP)
FACTOR=                                 TERM
                (/P1;)
TERM            TIMES       FACTOR=     TERM
                (/P3;
                STA 6 /F5,*T;
                /P1,*T.*T*I;
                FMU 6 /F5,*T;)

SVAR            DIVIDE      SVAR=       TERM
                (LDA /P1;
                LDQ /P3;
                SLJ 4 *ZDIVS)
```

```
SVAR        DIVIDE      FACTOR=     TERM
            (/P1;
            LDQ /P3;
            SLJ 4 *ZDIVS)

TERM·       DIVIDE      FACTOR=     TERM
            (/P3;
            STA 6 /F5,*T;
            /P1,*T.*T*I;
            LDQ 6 /F5,*T;
            SLJ 4 *ZDIVS)

TERM        DIVIDE      SVAR=       TERM
            (/P3;
            STA 6 /F5,*T;
            LDQ 6 /F5,*T;
            LDA /P1;
            SLJ 4 *ZDIVS)

+           SVAR=       SVAR        (/P1;)

-           SVAR=       PRIME       (LAC /P1;)


TERM        TIMES       SVAR=       TERM
            (/P3;
            FMU /P1;)
TERM        SLASH       FACTOR=     TERM
            (/P1;
            STA 6 /F5,*T;
            /P3,*T.*T*I;
            FDV 6 /F5,*T;)
TERM        SLASH       SVAR=       TERM
            (/P3;
            FDV /P1;)

TERM=                               TSUM
            (/P1;)
TSUM        +           TERM=       TSUM
            (/P3;
            STA 6 /F5,*T;
            /P1,*T.*T*I;
            FAD 6 /F5,*T;)
TSUM        +           SVAR=       TSUM
            (/P3;
            FAD /P1;)
TSUM        -           TERM=       TSUM
            (/P1;
            STA 6 /F5,*T;
            /P3,*T.*T*I;
            FSB 6 /F5,*T;)
TSUM        -           SVAR=       TSUM
            (/P3;
            FSB F1;)
TSUM=                               AREX
(/P1;)

LSTHEQ=     RBMA
(3)

GRTH=       RBMA
(2)

GRTHEQ=     RAMB
(3)

LSTH=       RAMB
(2)

BTERM       AND         BPRIME=     BTERM
            (/P3;
```

```
                /P1;)

BTERM=                                  BSUM
(/P1;)

BSUM            OR          BTERM=      BSUM
                (/P3,*J./F1;;
                SLJ Ø /F1;
/F2,-1;         /P1;
/F2;            ENI Ø Ø)

IF              BEX         THEN  AXPN  ELSE        AXPN=       AXPN
                (/P5,*J./F1;;
                /P3;
                SLJ Ø /F1;
/F2,-1;         /P1;
/F2;            ENI Ø Ø)

IF              BEX         THEN  BEX   ELSE        BEX=        BEX
                (/P5,*J./F1;;
                /P3;
                SLJ Ø /F1;
/F2,-1;         /P1;
/F2;            ENI Ø Ø)


BEX=            EXPN
                (/P1,*J./F1;;
                LDA Ø *F1PERIOD
                SLJ Ø DIVIDE+2
/F2;            ENA Ø Ø
                ENI Ø Ø)

AREX            REQU        AREX=       RELATN
                (/P3;
                STA 6 /F5,*T;
                /P1,*T.*T*I;
                FSB 6 /F5,*T;
                AJP /P2; *J)

AREX            REQU        SVAR=       RELATN
                (/P3;
                FSB /P1;
                AJP /P2; *J)

AREX            RBMA        AREX=       RELATN
                (/P3;
                STA 6 /F5,*T;
                /P1,*T.*T*I;
                FSB 6 /F5,*T;
                AJP /P2; *J)

AREX            RBMA        SVAR=       RELATN
                (/P3;
                FSB /P1;
                SCM Ø *ZALL7
                AJP /P2; *J)

AREX            RAMB        AREX=       RELATN
                (/P1;
                STA 6 /F5,*T;
                /P3,*T.*T*I;
                FSB 6 /F5,*T;
                AJP /P2; *J)

AREX            RAMB        SVAR=       RELATN
                (/P3;
                FSB /P1;
                AJP /P2; *J)
```

```
AREX=       AXPN        (/P1;)

AXPN=       EXPN        (/P1;)

LPAREN      BEX         RPAREN=     BPRIME
            (/P2;)

LPAREN      AXPN        RPAREN=     PRIME
            (/P2;)

PRIME=      FACTOR
(/P1;)

BPRIME=     BTERM
(/P1;)

RELATN=     BPRIME
(/P1;)

NOT         BOOLVAR=    BPRIME
            (LDA /P2;
            AJP 1 *J)

NOT         BPRIME=     BPRIME
            (/P2,*J./F1;;
/F2;        SLJ Ø *J)

AXPN        STEP        AXPN        UNTIL       AXPN=       FORELM
            (ENA Ø /F1;
            SAU Ø *C
            /P5;
            STA *B
            /P3;
            STA Ø *ZTEM
/F1;        /P1;
            FSB *B
/F1;        SSK Ø *ZTEM
            SCM Ø *ZALL7
            AJP Ø *S
            AJP 3 *S
            *A
/F2,-2;     /P3;
            STA Ø *ZTEM
            FAD *B
            STA *B
            SLJ Ø /F2,-1;)

INTVAR=     FORVAR      (Ø /P1;)

REALVAR=    FORVAR      (Ø /P1;)


AXPN        STEP        AXPN        WHILE       BEX=        FORELM

            (ENA Ø /F1;
            SAU Ø *C
            /P5;
            STA *B
/F1;        /P1,*J./F1;;
            SLJ Ø *S
/F2;        *A
/F2,-2;     /P3;
            FAD *B
            STA *B
            SLJ Ø /F2,-1;)

AXPN=       FORELM
            (ENA Ø /F1;
            SAU Ø *C
            /P1;
```

```
                STA *B
                SLJ Ø *S
/F2;            *A)

FORELM=         FORLIST
(/P1;)

FORLIST         COMMA       FORELM=     FORLIST
(/P3,*A.SLJ Ø /F1;;
/F2;            /P1;)

FOR             FORVAR      COLEQ       FORLIST     DO          STATEMENT=  FORST

(/P3,*A.SLJ Ø /F1;+1,*B./P5;,*C./F2;,*S./F1;;
/F2;            ENI Ø Ø
/P1;
/F2,-1;         SLJ Ø Ø
                ØØ Ø Ø)

FOR             FORPAR      COLEQ       FORLIST     DO          STATEMENT   FORST

(/P5;
                SIL 4 /F1;
                /P3,*A.SLJ Ø /F2;+1,*B.7 /F2;,*C./F1;,*S./F1;;
/F2;            ENI Ø Ø
                /P1;
/F2,-1;         SLJ Ø Ø
                ØØ Ø Ø
/F2,-2;         OCT Ø)

SVAR=           SUBLST
(LDA /P1;
                FSB 4 -/F5,*V*I*I;)

AREX=           SUBLST
                (/P1;
                FSB 4 -/F5,*V*I*I;)

SUBLST          COMMA       SVAR=       SUBLST
                (/P3,*V.*V*I*I;
                FMU 4 -/F5,*V*I*I*I;
                FAD /P1;
                FSB 4 -/F5,*V*I*I;)

SUBLST          COMMA       AREX=       SUBLST
                (/P3,*V.*V*I*I;
                FMU 4 -/F5,*V*I*I*I;
                STA 6 /F5,*T;
                SIU 4 /F1;
                /P1,*T.*T*I;
/F2;            ENI 4 Ø
                FAD 6 /F5,*T;
                FSB 4 -/F5,*V*I*I;)

LBRACK          SUBLST      RBRACK=     SUBVAR
                (/P2;
                FAD Ø *ZFIX
                SAU Ø /F1;
/F2;            INI 4 76567)

FSUBVAR=        PRIME
                (/P1;)

FSUBVAR=        BPRIME
                (/P1;
```

```
                AJP Ø *J)

VALPAR=         LSVAR
                (STA Ø /P1;)

REALVAR=        LSVAR          (STA Ø /P1;)

BOOLVAR=        LSVAR
                (AJP Ø /F1;
                LDA Ø *F1PERIOD
/F2;            STA Ø /P1;)

INTVAR =        LSVAR
                (FAD Ø *ZFIX
                FSB Ø *ZFIX
                STA Ø /P1;)

FORPAR=         LSVAR
                (STA Ø *ZTEM
                /P1;
/F1;            LRS Ø 3
                LDA Ø *ZTEM
                QJP 3 /F1;
                QLS Ø 1
                QJP 3 /F1;
                FAD Ø *ZFIX
                FSB Ø *ZFIX
                SLJ Ø /F2,-1;
/F2;            AJP Ø /F2,-1;
                LDA Ø *F1PERIOD
/F2,-1;         STA 4 Ø)

VALPAR          SUBVAR=      LSVAR
                (STA Ø *ZTEM
                LIL 4 /P2;
                /P1;
                LDA Ø *ZTEM
                STA 4 Ø)

RELARV          SUBVAR=      LSVAR
                (STA Ø *ZTEM
                LIL 4 /P2;
                /P1;
                LDA Ø *ZTEM
                STA 4 Ø)

BOLARV          SUBVAR=      LSVAR
                (STA Ø *ZTEM
                LIL 4 /P2;
                /P1;
                LDA Ø *ZTEM
                AJP Ø /F1;
                LDA Ø *F1PERIOD
/F2;            STA 4 Ø)

INTARV          SUBVAR=      LSVAR
                (STA Ø *ZTEM
                LIL 4 /P2;
                /P1;
                LDA Ø *ZTEM
                FAD Ø *ZFFIX
                FSB Ø *ZFIX
                STA 4 Ø)

FORPAR          SUBVAR=      LSVAR
                (STA Ø *ZTEM
                /P2;
/F1;            STA 6 /F5,*T;
                /P1,*T.*T*I;
                LDA Ø /F5,*T;
                LRS Ø 3
```

```
                LDA Ø *ZTEM
                QJP E /F1;
                QLS Ø 1
                QJP 3 /F1;
                FAD Ø *ZFIX
                FSB Ø *ZFIX
                SLJ Ø /F2,-1;
/F2;            AJP Ø /F2,-1;
                LDA Ø *F1PERIOD
/F2,-1;         STA 4 Ø)

LSVAR           COLEQ       ASGNST=     ASGNST
                (/P1;
                /P3;)

LSVAR           COLEQ       EXPN=       ASGNST

                (/P1;
                /P3;)


BSUM=           BIMP        (/P1;)

BIMP            IMPL        BSUM=       BIMP
                (/P3,*J./F1;;
                /P1;
/F2;            ENI Ø Ø)

BIMP=           BEQV
(/P1;)

BEQV            EQV         BIMP=       BIMP
                (ENA Ø Ø
                STA 6 /F5,*T;
                /P3,*J./F1;,*T.*T*I;
                RAO 6 /F5,*T;
/F2;            /P1,*J./F1;,*T.*T*I;
                RAO 6 /F5,*T;
/F2;            LRS Ø 1
                AJP 3 *J)

BEQV=           BEX         (/P1;)


AXPN=           ACTPAR
(*D             SLJ Ø Ø
                /P1;
/F1;            STA Ø *ZTKPO
                ENI 4 *ZTKPO
                ENA Ø Ø
                SLJ Ø *D)

BEX=            ACTPAR
(               STA Ø *ZTKPO
                ENI 4 *ZTKPO
*D              SLJ Ø Ø
                /P1,*J./F1;;
/F1;            LDA Ø *F1PERIOD
                SLJ Ø *D-1
/F2,-1;         ENA Ø Ø
                SLJ Ø *D)

DESEX=          ACTPAR
(*D             SLJ Ø Ø
                /P1;
                STA Ø *ZTKPO
                LIL 4 *ZTKPO
                ENA Ø 100
                SLJ Ø *D)

SWIDEN=         ACTPAR
(*D             SLJ Ø Ø
```

```
                ENI 4 /P1;
                ENA Ø 4ØØ
                SLJ Ø *D)

PRIND=          ACTPAR
(*D             SLJ Ø Ø
                ENI 4 /P1;*W
                ENA Ø 1ØØØ
                SLJ Ø *D)

REALVAR=        ACTPAR
(*D             SLJ Ø Ø
                ENI 4 /P1;
                ENA Ø 4
                SLJ Ø *D)

BOOLVAR=        ACTPAR
(*D             SLJ Ø Ø
                ENI 4 /P1;
                ENA Ø 2
                SLJ Ø *D)

INTVAR=         ACTPAR
(*D             SLJ Ø Ø
                ENI 4 /P1;
                ENA Ø 1
                SLJ Ø *D)

RELARV=         ACTPAR
(*D             SLJ Ø Ø
                LIL 4 /P1;
                ENA Ø 2Ø4
                SLJ Ø *D)

BOLARV=         ACTPAR
(*D             SLJ Ø Ø
                LIL 4 /P1;
                ENA Ø 2Ø2
                SLJ Ø *D)

INTARV=         ACTPAR
(*D             SLJ Ø Ø
                LIL 4 /P1;
                ENA Ø 2Ø1
                SLJ Ø *D)

RELARV          SUBVAR=     ACTPAR
(*D             SLJ Ø Ø
                LIL 4 /P2;
                /P1;
                ENA Ø 4
                SLJ Ø *D)

BOLARV          SUBVAR=     ACTPAR
(*D             SLJ Ø Ø
                LIL 4 /P2;
                /P1;
                ENA Ø 2
                SLJ Ø *D)

INTARV          SUBVAR=     ACTPAR
(*D             SLJ Ø Ø
                LIL 4 /P2;
                /P1;
                ENA Ø 1
                SLJ Ø *D)

FORPAR          SUBVAR=     ACTPAR
(*D             SLJ Ø Ø
                /P2;
/F1;            SAU Ø /F1;
                /P1;
```

```
/F2;            ENA Ø 76567
                SLJ Ø *D)

VALPAR          SUBVAR=     ACTPAR
(*D             SLJ Ø Ø
                LIL 4 /P2;
                /P1;
                ENA Ø 4
                SLJ Ø *D)

FORPAR=         ACTPAR
(*D             SLJ Ø Ø
                /P1;
                SLJ Ø *D)

REALVAR=        SVAR        (Ø /P1;)

INTVAR=         SVAR        (Ø /P1;)

BOOLVAR=        BPRIME      (LDA Ø /P1;
                AJP Ø *J)

VALPAR=         SVAR        (Ø /P1;)

RELARV          SUBVAR=     PRIME
                (LIL 4 /P2;
                /P1;
                LDA 4 Ø)

INTARV          SUBVAR=     PRIME
                (LIL 4 /P2;
                /P1;
                LDA 4 Ø)

BOLARV          SUBVAR=     BPRIME
                (LIL 4 /P2;
                /P1;
                LDA 4 Ø
                AJP Ø *J)

VALPAR          SUBVAR=     FSUBVAR
                (LIL 4 /P2;
                /P1;
                LDA 4 Ø)

FORPAR          SUBVAR=     FSUBVAR
                (/P2;
                /P1;
                LDA 4 Ø)


ACTPAR=         APLIST
                (ØØ Ø *P
                ØØ Ø /F1;
/P1,*D./F2;;)

APLIST          SEP         ACTPAR=     APLIST
                (ØØ Ø *P
                ØØ Ø /F1;
                /P3,*P.;
/P1,*D./F2;;)
PRIDEN          LPAREN      APLIST      RPAREN=     PROCST
                (SLJ 4 /P4;*W
/F1;            /P2,*P./F1;;
/F2;            ENI Ø Ø)

FORPAR          LPAREN      APLIST      RPAREN=     PROCST
                (/P4;
                SIU 4 /F1;
/F2;            SLJ 4 76567
                ØØ Ø Ø
                /P2,*P./F1;;
```

```
/F2;          ENI Ø Ø)

FORPAR=       PROCST
              (/P1;
              SIU 4 /F1;
/F2;          SLJ 4 76567
              ØØ Ø Ø
/F1;          ENI Ø /F2;)

PRIDEN=       PROCST
              (SLJ 4 /P1;*W
/F1;          ENI Ø /F2;)

FIDEN=        LPAREN      APLIST      RPAREN=     PRIME
              (INI 6 /F5,*T;
              SLJ 4 /P4;*W
/F1;          /P2,*P./F1;;
/F2;          INI 6 -/F5,*T;)

FIDEN=        PRIME
              (INI 6 /F5,*T;;
              SLJ 4 /P1;*W
/F1;          ENI Ø /F2;
              INI 6 -/F5,*T;)

FORPAR        LPAREN      APLIST      RPAREN=     PRIME
              (INI 6 /F5,*T;
              /P4;
              SIU 4 /F1;
/F2;          SLJ 4 76567
              ØØ Ø Ø
              /P2,*P./F1;;
/F2;          INI 6 -/F5,*T;)

FORPAR=       PRIME
              (INI 6 /F5,*T;
              /P1;
              LDA 4 Ø
              INI 6 -/F5,*T;)

SWIDEN        LBRACK      AXPN        RBRACK=     DESEX
              (/P2;
              FAD Ø *ZFIX
              SUB Ø *ZFIXONE
              AJP 3 *ZPANIC
              SUB Ø /P4;
              AJP 2 *ZPANIC
              INA Ø /P4;
              SAU Ø /F1;
/F2;          LDA Ø 76567)

LABEL=        DESEX       (ENA Ø /P1;)

FORPAR        LBRACK      AXPN        RBRACK=     DESEX
              (/P4;
              LRS Ø 11
              QJP 2 *ZPANIC
              SIU 4 /F1;
              SIU 4 /F2;+1
              /P2;
              FAD Ø *ZFIX
              SUB Ø *ZFIXONE
              AJP 3 *ZPANIC
/F2;          SUB Ø 76567
              AJP 2 *ZPANIC
              INA Ø 76567
              SAU Ø /F2;+2
              LDA Ø 76567)

FORPAR=       DESEX
              (/P1;
```

```
                LRS Ø 7
                @JP 2 *ZPANIC
                ENA 4 Ø)

IF              BEX         THEN        DESEX       ELSE        DESEX=      DESEX
                (/P5,*J./F1;;
                /P3;
                SLJ Ø /F1;
/F2,-1;         /P1;
/F2;            ENI Ø Ø)

GO TO           DESEX=      GOTOST
                (/P1;
                SLJ Ø *ZGOINT)

ASGNST=         STATEMENT
(               /P1;
*P              EQV *Q)

GOTOST=         STATEMENT
(               /P1;
*P              EQV *Q)

PROCST=         STATEMENT
(               /P1;
*P              EQV *Q)

FORST=          STATEMENT
(               /P1;)

BLOCK=          STATEMENT
                (/P1;)

IF              BEX         THEN        STATEMENT=  STATEMENT
(               /P3,*J./F1;;
/P1;
/F2;            ENI Ø Ø)

IF              BEX         THEN        STATEMENT   ELSE  STATEMENT=  STATEMENT
(               /P5,*J./F1;;
/P3,*Q./F1;;
                SLJ Ø /F1;
/F2,-2;         ENI Ø Ø
/P1,*P./F2,-1;;
/F2;            ENI Ø Ø)

LABEL           COLON       STATEMENT=  STATEMENT
(/P3;           ENI Ø Ø
/P1;)

STATEMENT=      STLIST      (/P1;)

STLIST          SEMIC       STATEMENT=  STLIST
                (/P3,*Q./F1;;
/P1,*P./F2;;)
```

```
INTEGER=        TNAME           ()

BOOLEAN=        TNAME           ()

REAL=                           TNAME           ()

OWN             TNAME           =               OWNLIST         ()

OWNLIST         TVAR            COMMA=          OWNLIST
                (/P3;
/P2;            OCT Ø)

OWNLIST=        OTDEC           (/P1;)
```

```
TNAME           TVAR=           TDEC
(*P             ØØ Ø *Q
                ØØ Ø /P1;
/P1;            OCT Ø)

TDEC            COMMA           TVAR=           TDEC
                (/P3,*Q./F1;;
/F2;            ØØ Ø *Q

                ØØ Ø /P1;
/P1;            OCT Ø)
```

```
AXPN            COLON           AXPN=           BNDPR
                (/P3;
                STA 6 -/F5,*V*I*I*I*I;
                /P1;
                FSB 6 -/F5,*V*I*I*I*I;
                FAD Ø *F1PERIOD
                STA 6 -/F5,*V*I*I*I*I*I;
                FMU Ø *ZTEM
                STA Ø *ZTEM)
```

```
REAL            ARRAY=          ARDECIN         ()

BOOLEAN         ARRAY=          ARDECIN         ()

INTEGER         ARRAY     =     ARDECIN         ()

ARRAY           =               ARDECIN         ()

OWN             ARDECIN=        OWNAR
(*5             LDA 6 -1
                ENQ 6 -1
```

```
                INI 6 /F5,*V*I*I;
                STA 6 -1
                STQ 6 -2
                LDA Ø *F1PERIOD
                STA Ø *ZTEM)

OWNAR           ARIDEN=     OARLIST
(/P1;           OCT Ø
/P2,*V.*V*I;
                ENA Ø /P1;
                SAL 6 -/F5,*V*I*I*I*I;)

OARLIST         COMMA=      OWNAR       (/P2;)

OARLIST         LBRACK      BNDPR=      OARSEG
                (/P3,*V.*V*I*I;
                ENA Ø -/F5,*V*I*I;
                STA 6 -3
                /P1;)

OARSEG          COMMA       BNDPR=      OARSEG
                (/P3,*V.*V*I*I;
                /P1;)

OARSEG          RBRACK=     OARDEC
                (/P2;
/F1;            LDA Ø *ZTEM
                SLJ 4 *ZFUSBDG
                SLJ Ø *M)

OARDEC          COMMA=      OWNAR
                (/P2,*M./F1;,*V.*I;
/F2;            LDA 6 -1
                ENQ 6 -1
                INI 6 /F5,*V*I*I;
                STA 6 -1
                STQ 6 -2
                LDA Ø *F1PERIOD
                STA Ø *ZTEM)

ARDECIN         ARIDEN=     ARLIST
(*P             ØØ Ø *Q
                ØØ Ø /P1;
/P1;            OCT Ø
*3              LDA 6 -1
                ENQ 6 -1
                INI 6 /F5,*V*I*I*I;
                STA 6 -1
                STQ 6 -2
                ENA Ø /P1;
                SAL 6 -/F5,*V*I*I*I*I;
                LDA Ø *F1PERIOD
                STA Ø *ZTEM)

ARLIST          COMMA       ARIDEN=     ARLIST
(/P1;           OCT Ø
*P              ØØ Ø /F1;
                ØØ Ø /P1;
/P3,*P./F2;,*V.*V*I;
                ENA Ø /P1;
                SAL 6 -/F5,*V*I*I*I*I;)

ARLIST          LBRACK      BNDPR=      ARSEG
                (/P3,*V.*V*I*I;
                ENA Ø /F5,*V*I*I;
                STA 6 -3
                /P1;)

ARSEG           COMMA       BNDPR=      ARSEG
```

```
                (/P3,*V.*V*I*I;
                /P1;)

ARSEG           RBRACK=     ARDEC
                (/P2;
                LDA Ø *ZTEM
                SLJ 4 *ZFUSBDG
/F1;            SLJ Ø *M)

ARDEC           COMMA       ARIDEN=     ARLIST
                (/P3,*Q./F1;,*M./F1;,*V.*I;
/F2,-1;         ØØ Ø *Q
                ØØ Ø /P1;
/P1;            OCT Ø
/F2;            LDA 6 -1
                ENQ 6 -1
                INI 6 /F5,*V*I*I*I;
                STA 6 -1
                STQ 6 -2
                ENA Ø /P1;
                SAL 6 -/F5,*V*I*I*I*I;
                LDA Ø *F1PERIOD
                STA Ø *ZTEM)

SWITCH          SWIDEN      COLEQ       DESEX=      SWDEC
(               ØØ Ø Ø
                ØØ Ø /F1;
/P3;            ØØ Ø Ø
                ØØ Ø /F5,*V*I;
/F2;            /P1;)

SWDEC           COMMA       DESEX=      SWDEC
(               ØØ Ø Ø
                ØØ Ø /F1;
/P3;
/F2;            /P1;)

FORPAR=         FORPARLIST
(*S             ENI Ø Ø
*P              EQV *Q)

FORPARLIST      SEP         FORPAR=     FORPARLIST

(/P3,*V.*V*I;)

VALPAR=         FORPARLIST
(/P1;           OCT Ø
*P              ØØ Ø *Q
                ØØ Ø /P1;
*S              ENI 3 /F5,*V;
                LDA 7 *R*U
                SAU Ø /F1;
                ENI Ø Ø
/F2;            SLJ 4 76567
                ØØ Ø Ø
                LRS Ø 10
                QJP 7 *ZARMOVE
                LDA 4 Ø
                STA Ø /P1;)

FORPARLIST      SEP         VALPAR=     FORPARLIST
(/P1;           OCT Ø
*P              ØØ Ø /F1;
                ØØ Ø /P1;
/P3,*P./F2;,*V.*V*I;
                ENI 3 /F5,*V;
                LDA 7 *R*U
                SAU Ø /F1;
                ENI Ø Ø
/F2;            SLJ 4 76567
                ØØ Ø Ø
                LRS Ø 10
```

```
                QJP 7 *ZARMOVE
                LDA 4 Ø
                STA Ø /P1;)

PROCEDURE=      PRDECA
(*S             RSO Ø *R*C
                LDA Ø *R*E
                ENI 2 *R
                SLJ Ø *ZPROXIT
*R              OCT Ø
*R*E            OCT Ø
*R*U            OCT 3ØØØØØ
*R*C            OCT -2
/F1;            ØØ Ø /F1;
                ØØ Ø *R*U
/F2;            ØØ Ø /F1;
                ØØ Ø *R*E
/F2;            ØØ Ø *Q
                ØØ Ø *R
*R*W            ØØ Ø Ø
                LDA Ø *ZEXIT
                STA 6 Ø
                INI 6 1
                ENQ Ø Ø
                ENA Ø *B
                LRS Ø 17
                ENA Ø *R*W
                LRS Ø 17
                STQ Ø *ZEXIT
                RAO Ø *R*C
                AJP 3 /F1;
                ENQ Ø /F2,-3;
                LDA Ø *R*W
                SLJ 4 *ZRECURS
                ØØ Ø Ø
                SLJ Ø DIVIDE+2
                ØØ Ø Ø
/F2;            LDA Ø *R*W
                ARS Ø 3Ø
                SAL Ø *R*U
                ENI 3 Ø
                LDA 7 *R*U
                ARS Ø 3Ø
                SAL Ø *R*E
                ENA 6 Ø
                STA 6 Ø
                INI 6 1
)

TNAME           PRDECA=     PRDECB      (/P1;)

PRDECA=         PRDECB      (/P1;)

PRDECB          PRIND       LPAREN      FORPARLIST  RPAREN=     PRDECC
                (/P5,*Q./F1;,*R./P4;;
                SLJ Ø /F1;
/P2,*S./F2;,*P./F2,-1;,*R./P4;;
*R              EQV /P4;*U)

PRDECC          SEMIC       SPEC=       PRDECC      (/P3;)

PRDECC          SEMIC       STATEMENT=  PRDEC
                (/P3,*R./F1;,*Q./F1;,*S./F1;,*B./F1;;
/P1,*P./F2,-2;,*Q.Ø,*R./F2,-3;;
/F2;            SLJ Ø /F2,-1;
                ØØ Ø Ø)



ARDEC=          DEC         (/P1;)

CARDEC=         DEC
```

```
                    (/P1;
*P                  EQV *Q)

TDEC=               DEC
                    (/P1;
*S                  EQV *M)

OTDEC=              DEC
                    (/P1;
*P                  EQV *Q
*S                  EQV *M)

SWDEC=              DEC
                    (/P1;
*P                  EQV *Q
*S                  EQV *M)

PRDEC=              DEC
                    (/P1;
*P                  EQV *Q
*S                  EQV *M)

DEC=                DECLIST
(/F1;               LDA Ø *ZEXIT
                    STA 6 Ø
                    INI 6 1
                    ENQ Ø Ø
                    ENA Ø *B
                    LRS Ø 17
                    ENA Ø /F2;
                    LRS Ø 17
                    STQ Ø *ZEXIT
                    ENA 6 Ø
                    STA 6 Ø
                    INI 6 1
                    SLJ Ø /F1;
                    ØØ Ø Ø
/P1,*S./F2;;)

DECLIST             SEMIC           OTDEC=          DECLIST
                    (/P3;
/P1;)

DECLIST             SEMIC           SWDEC=          DECLIST
                    (/P3;
/P1;)

DECLIST             SEMIC           PRDEC=          DECLIST
                    (/P3;
/P1;)

DECLIST=            SEMIC           ARDEC=          DECLIST
                    (/P3,*Q./F1;,*M./F1;;
/P1,*P./F2,-1;,*S./F2;;)

DECLIST             SEMIC           TDEC=           DECLIST
                    (/P3,*Q./F1;;
/P1,*P./F2;;)

DECLIST             SEMIC           OARDEC=         DECLIST
                    (/P3,*M./F1;;
/P1,*S./F2;;)

BEGIN               DECLIST         SEMIC           STLIST      END=        BLOCK
                    (/P4,*Q./F1;,*M./F1;,*B./F1;;
/F2,-1;             ENI Ø Ø
/P2,*P./F2,-2;;
/F2;                ENA Ø /F2;+1
                    SLJ Ø *ZGOINT)
```

```
BEGIN       STLIST      END         =STATEMENT
            (/P2;)

VALUE=      SPECIND     ()

STRING=     SPECIND     ()

LLABEL=     SPECIND     ()

SWITCH=     SPECIND     ()

PROCEDURE   SPECIND     ()

INAME       PROCEDURE=  SPECIND     ()

INAME=      SPECIND     ()

ARDECIN=    SPECIND     ()

SPECIND     IDEN=       SPEC        ()

SPEC        COMMA       IDEN=       SPEC        ()

LETTER=     IDEN        ()

IDEN        LETTER=     IDEN        ()

IDEN        DIGIT=      IDEN        ()

LSTG        LETTER=     LSTG        (/P2;/P1;)

RPAREN      LSTG        COLON       LPAREN=     SEP         ()

COMMA=      SEP         ()

NSX=        NTM         (/P1;)

LQUOTE=     NTM         (LQUOTE)
RQUOTE=     NTM         (RQUOTE)

END         NTM=        END         ()

BEGIN       COMMENT     NTM=        BEGIN       ()

SEMIC       COMMENT     NTM=        SEMIC       ()

NTM         NTM=        NTM         (/P2; /P1;)


STATEMENT   NTM=        STATEMENT
            (/P2;
            REM   =ERROR  /P1;=)

SEMIC       SEMIC=      SEMIC       ()

LABEL       COLON=      STATEMENT
(/P2;       ENI Ø Ø)

TRUE=       BOOLVAR     (*F1PERIOD)

FALSE=      BOOLVAR     (*FØPERIOD)

=           NSX         (* ILLEGAL *)

CARETN=     NSX         (
)
```

```
TAB=        NSX         (           )

SPACE=      NSX         ( )

TRUE        NSX         (TRUE)

FALSE=      NSX         (FALSE)

NTM=        STATEMENT
(           REM   =ERROR   /P];=
*P          EQV *Q)

PRINT=      PRIDEN      (*ZPRINT)
INPUT=      PRIDEN      (*ZINPUT)

SEMIC=      NQT         (SEMIC)

END=        NQT         (END)
NSX=        NQT         (/P1;)

ELSE=       NQT         (ELSE)

NQT         NQT=        NQT         (/P2;/P1;)

LQUOTE      NQT         RQUOTE=     ACTPAR
(*D         SLJ Ø Ø
            ENI 4 DIVIDE+2
            ENA Ø 2ØØØ
            SLJ Ø *D/F6,/P2;;)

LQUOTE      NQT         RQUOTE=     NQT         (LQUOTE/P2;RQUOTE)

A=          LETTER      (A)
B=          LETTER      (B)
C=          LETTER      (C)
D=          LETTER      (D)
E=          LETTER      (E)
F=          LETTER      (F)
G=          LETTER      (G)
H=          LETTER      (H)
I=          LETTER      (I)
J=          LETTER      (J)
K=          LETTER      (K)
L=          LETTER      (L)
M=          LETTER      (M)
N=          LETTER      (N)
O=          LETTER      (O)
P=          LETTER      (P)
Q=          LETTER      (Q)
R=          LETTER      (R)
S=          LETTER      (S)
T=          LETTER      (T)
U=          LETTER      (U)
V=          LETTER      (V)
W=          LETTER      (W)
X=          LETTER      (X)
Y=          LETTER      (Y)
Z=          LETTER      (Z)

LETTER=     LSTG        (/P1;)

LSTG=       NSX         (/P1;)

DIGIT=      LETTER      (/P1;)

TEN=        LETTER      (TEN)
```

```
PERIOD=     LETTER      (PERIOD)


DIVIDE=     NSX         (DIVIDE)
COLON=      NSX         (COLON)
OWN         NSX         (OWN)
EQV=        NSX         (EQV)
IMPL=       NSX         (IMPL)

STATEMENT=  PROGRAM
(*ZARMOVE   EQV *ZSERVCE+1
*ZEXP       EQV *ZSERVCE+2
*ZDIVS      EQV *ZSERVCE+3
*ZRECURS    EQV *ZSERVCE+4
*ZGOINT     EQV *ZSERVCE+5
*ZFUSBDG    EQV *ZSERVCE
*ZPROXIT    EQV *ZSERVCE+6
*ZPANIC     EQV *ZSERVCE+7
*ZPRINT*W   EQV *ZSERVCE+10
*ZEXIT      EQV *ZSERVCE+11
*ZINPUT*W   EQV *ZSERVCE+12
/F1;        SLJ 0 0
/P1,*Q-0;
            SLJ 0 /F2;
*F1PERIOD   OCT 2001400000000000
*F2PERIOD   OCT 2002400000000000
*F3PERIOD   OCT 2002600000000000
*F4PERIOD   OCT 2003400000000000
*F5PERIOD   OCT 2003500000000000
*F6PERIOD   OCT 2003600000000000
*F7PERIOD   OCT 2003700000000000
*F8PERIOD   OCT 2004400000000000
*F9PERIOD   OCT 2004440000000000
*FBPERIOD   OCT 0
*ZTEM       OCT 0
*ZTKPO      OCT 0
*ZALL7      OCT 7777777777777777
*ZFIX       OCT 2044600000000000
*ZFIXONE    OCT 2044600000000001
*ZSERVCE    EQV 3000
            /F7;
            END

)


COLEQ=      NSX         (COLEQ)

OR=         NSX         (OR)
AND=        NSX         (AND)
LSTH=       NSX         (LSTH)
LSTHEQ=     NSX         (LSTHEQ)
GRTH=       NSX         (GRTH)
GRTHEQ=     NSX         (GRTHEQ)
EQUAL=      NSX         (EQUAL)
NOT=        NSX         (NOT)
UNEQ=       NSX         (UNEQ)
COMMA=      NSX         (COMMA)
+=          NSX         (+)
-=          NSX         (-)
TIMES=      NSX         (TIMES)
SLASH=      NSX         (SLASH)
LPAREN=     NSX         (LPAREN)
RPAREN=     NSX         (RPAREN)
```

```
LBRACK=         NSX             (LBRACK)
RBRACK=         NSX             (RBRACK)
EXP=            NSX             (EXP)

BEGIN           NSX             (BEGIN)

IF              NSX             (IF)
THEN            NSX             (THEN)
GO TO           NSX             (GO TO)
FOR             NSX             (FOR)
STEP            NSX             (STEP)
UNTIL           NSX             (UNTIL)
WHILE           NSX             (WHILE)
DO              NSX             (DO)
ARRAY           NSX             (ARRAY)
COMMENT         NSX             (COMMENT)
PROCEDURE       NSX             (PROCEDURE)
SWITCH          NSX             (SWITCH)
INTEGER         NSX             (INTEGER)
BOOLEAN         NSX             (BOOLEAN)
REAL            NSX             (REAL)
LLABEL          NSX             (LLABEL)
VALUE           NSX             (VALUE)
STRING          NSX             (STRING))))))))

0=              DIGIT           (0)
1=              DIGIT           (1)
2=              DIGIT           (2)
3=              DIGIT           (3)
4=              DIGIT           (4)
5=              DIGIT           (5)
6=              DIGIT           (6)
7=              DIGIT           (7)
8=              DIGIT           (8)
9=              DIGIT           (9)
A=              LETTER          (A)
B=              LETTER          (B)
C=              LETTER          (C)
D=              LETTER          (D)
E=              LETTER          (E)
F=              LETTER          (F)
G=              LETTER          (G)
H=              LETTER          (H)
I=              LETTER          (I)
J=              LETTER          (J)
K=              LETTER          (K)
L=              LETTER          (L)
M=              LETTER          (M)
N=              LETTER          (N)
O=              LETTER          (O)
P=              LETTER          (P)
Q=              LETTER          (Q)
R=              LETTER          (R)
S=              LETTER          (S)
T=              LETTER          (T)
U=              LETTER          (U)
V=              LETTER          (V)
W=              LETTER          (W)
X=              LETTER          (X)
Y=              LETTER          (Y)
Z=              LETTER          (Z)

INTVAR          DIGIT=          INTVAR          (/P2;/P1;)

DIGIT=          INTVAR          (/P1;)
```

```
NUMBER=     NSX         (/P1;)

TEN         +           INTVAR=     EXPN        (+/P1;)

TEN         -           INTVAR=     EXPN        (-/P1;)

TEN         INTVAR=     EXPN        (/P1;)
DIGIT       PERIOD      INTVAR=     PRIME       (/P3;PERIOD/P1;)

DIGIT       INTVAR      PERIOD      INTVAR=     PRIME
            (/P4;/P3;PERIOD/P1;)

PERIOD      INTVAR=     PRIME       (PERIOD/P1;)

DIGIT       INTVAR=     PRIME       (/P2;/P1;*R)

PRIME=      NUMBER
(
/P1,*R.;    NUMBER      (*F'/P1,*R.PERIOD;))

EXPN=       NUMBER
(
TEN/P1;     NUMBER      (*F1PERIODD'/P1,-.B,+.A;))

PRIME       EXPN=       NUMBER
(
/P2,*R.;TEN/P1;         NUMBER      (*F'/P2,*R.PERIOD;D/P1,-.B,+.A;))

BEGIN       DECLIST     SEMIC       STLIST      END=        BLOCK
(/P4,*G./F4;;/P2,*G./F4;;)

BEGIN       STLIST      END=        BLOCK
(/P2;)

NTM=        BLOCK
(/P1;)

LABL        COLON       BLOCK=      BLOCK
(/P1;
/P3;        LABEL       ('/P3;))

STLIST      SEPARATOR   BLOCK=      STLIST
(/P3;/P1;)

BLOCK=      STLIST      (/P1;)

STLIST      PROGRAM     (/P1,*G.;)

SPECIND     NOTSC=      SPEC        ()

SPEC        SPECIND     NOTSC=      SPEC        ()

DECIDEN=    SPECIND     ()

STRING=     SPECIND     ()

PROCEDURE=  SPECIND     ()

LLABEL=     SPECIND     ()

DEC=        DECLIST     (/P1;)

DECLIST     SEMIC       DEC=        DECLIST
(/P3;
/P1;)

NOTSC       NOTSC=      NOTSC       (/P2;/P1;)
```

```
NSX             NSX=            NSX             (/P2;/P1;)

PRDECA          LPAREN          IDEN=           PROCST
(/P3;
/P1;            XPAR            ('/F5,*V;))

PROCST          SEP             IDEN=           PROCST
(/P3,*V.*V*I;
/P1;            XPAR            ('/F5,*V;))

DECBEG          COLEQ           NOTSC=          DECBEG          (/P3;)

COMMA=          SEP             ()

RPAREN          NSX             COLON           LPAREN=         SEP             ()

END             NOTSC=          END             ()
BEGIN           COMMENT         NOTSC=          BEGIN           ()

SEMIC           COMMENT         NOTSC=          SEMIC           ()
SEMIC           SEMIC=          SEMIC           ()


PROCST          RPAREN=         PRDECB
(/P2;)

PRDECA=         PRDECC          (/P1;)

PRDECB=         PRDECC          (/P1;)

PRDECB          SEMIC           VALUE           IDLIST=         PRDECC
(/P1,*M.VALPAR;
/P4;)

PRDECC          SEMIC           SPEC=           PRDECC          (/P3;)

PRDECC          SEMIC           BLOCK=          DEC
(/P1,*G./F4;;
/P3,*H.*G,*G./F4;;)

IDEN=           IDLIST
(/P1;           *M              ('/P1;))

IDLIST          COMMA           IDEN=           IDLIST
(/P3;
/P1;            *M              ('/P1;))

SWITCH=         DECIDEN         (SWIDEN)

INTEGER=        DECIDEN         (INTVAR)

BOOLEAN=        DECIDEN         (BOOLVAR)

REAL=                           DECIDEN         (REALVAR)
OWN             DECIDEN=        DECIDEN         (/P1;)

ARRAY=                          DECIDEN         (RELARV)

REAL            ARRAY=          DECIDEN         (RELARV)

INTEGER         ARRAY=          DECIDEN         (INTARV)

BOOLEAN         ARRAY=          DECIDEN         (BOLARV)

DECIDEN         IDLIST=         DECBEG
(/P1,*M./P2;;)

DECBEG=         DEC             (/P1;)

IDLIST          LBRACK          NOT             RBRACK=         IDLIST          (/P4;/P2;)
NTM             NTM=            NTM             (/P2;/P1;)
```

LBRACK	NQT	RBRACK=	NQT	(/P2;)
NSX=	NTM	(/P1;)		
NSX=	NQT	(/P1;)		
COLON=	NQT	()		
THEN=	NQT	()		
ELSE=	NQT	()		
NQT	NQT=	NQT	(/P2;/P1;)	
INTEGER	PROCEDURE=	FIND	(INTVAR)	
BOOLEAN	PROCEDURE=	FIND	(BOOLVAR)	
REAL	PROCEDURE=	FIND	(REALVAR)	
FIND	IDEN=	PRDECA		
(/P1,*G.*H;	FIDEN	('/P1,*G.*H;)		
/P1,*G.*H*G;	/P2;	('/P1,*G.*H;))		
PROCEDURE	IDEN=	PRDECA		
(/P1,*G.*H;	PRIDEN	('/P1,*G.*H;))		
LETTER=	IDEN	(/P1;*G)		
IDEN	LETTER=	IDEN	(/P2,*G.;/P1;*G)	
IDEN	DIGIT=	IDEN	(/P2,*G.;/P1;*G)	
IDEN=	LABL	(/P1;)		
LABL	COLON=	BLOCK	(/P2;	LABEL ('/P2;))
IDEN=	NSX	()		
TRUE	NSX	()		
FALSE	NSX	()		
CARETN=	NSX	()		
TAB=	NSX	()		
SPACE=	NSX	()		
=	NSX	()		
DIGIT=	NSX	()		
COLEQ=	NSX	()		
OR=	NSX	()		
AND=	NSX	()		
LSTH=	NSX	()		
LSTHEQ=	NSX	()		
GRTH=	NSX	()		
GRTHEQ=	NSX	()		
EQUAL=	NSX	()		
NOT=	NSX	()		
UNEQ=	NSX	()		
COMMA=	NSX	()		
+=	NSX	()		
-=	NSX	()		
TIMES=	NSX	()		
SLASH=	NSX	()		
LPAREN=	NSX	()		
RPAREN=	NSX	()		
LBRACK=	NTM	()		
RBRACK=	NTM	()		
EXP=	NSX	()		
IF=	NSX	()		
GO TO=	NSX	()		
FOR=	NSX	()		
STEP=	NSX	()		
UNTIL=	NSX	()		
WHILE=	NSX	()		

```
COMMENT=        NSX         ()
TEN=            NSX         ()
PERIOD=         NSX         ()
COLON=          NOTSC       ()
THEN=           NOTSC       ()
DO=             NOTSC       ()
BEGIN=          NOTSC       ()
VALUE=          NOTSC       ()
LQUOTE=         NSX         ()
RQUOTE=         NSX         ()
SPECIND=        NOTSC       ()
NOTSC=          BLOCK       ()
NTM=            NOTSC       (/P1;)
THEN=           SEPARATOR   ()
DO=             SEPARATOR   ()
ELSE=           SEPARATOR   ()
SEMIC=          SEPARATOR   ()

DIVIDE=         NSX         ()
OWN             NSX         ()
EQV             NSX         ()
```