```
#1 LOGO,20
LOGO STRING LANGUAGE  RJG+PMW  5/21/1970  (LOGO,20) - PASS 1

#2 LOGO,20
LOGO STRING LANGUAGE  RJG+PMW  5/21/1970  (LOGO,20) - PASS 2
PNT   5727    SEG+2    SEG
PNT 1   7225    BEFORC+55    COMMANDS 1
PNT 1   7215    TURTH+4    COMMANDS 2
PNT 1   7353    LISTAB+2    LIST
PNT 1   7152    SVDRA+2    SAVE
PNT 1   7103    HOARDD+1    GET FILE IN
PNT 1   7351    GDBYEB+1    EDIT ERASE DEFINITION
PNT 1   7210    INTXTD+3    TT INPUT
PNT 1   7153    GTGETA+2    FILE INPUT
PNT 1   7341    LTALT2+2    PRINT AND REQUEST
PNT 1   6664    SUPDOG+34    SUPERDO
PNT 1   7164    DTEM+2    RANDOM FILE STUFF
PNT 1   6261    FINITB+3    Q-RANDOM FILE STUFF
PNT 1   7003    EGETX+1    LINE EDITOR
PNT 1   7122    ERTBL+757    ERRORS
PNT 1   7271    ETXTB2+2    SECOND ERRORS
USW   6076    EFIN3+2    CHARAC    PDL
USW   6101    EFIN3+5    CHARAC    CAP
PNT 1   6123    ETXTB3+2    THIRD ERRORS
PNT 1   6511    INCON1+1    INITIALIZATION
PNT   7656      FOO

#J

#B LOGO,20

#S LOGO,20

#A

CONSTANTS AREA, INCLUSIVE
FROM        TO
7401        7656

#H
```

```
DRA IF DRUM: 275730
277304  COMMAND TABLE
277505  SYSTEM TABLE
301021  LIST TABLE
300015  COMMANDS 1
300746  COMMANDS 2
300145  LIST
300447  SAVE
300110  GET FILE IN
300152  EDIT ERASE DEFINITION
300742  TT INPUT
300543  FILE INPUT
301042  PRINT AND REQUEST
300236  SUPERDO
301677  RANDOM FILE STUFF
300512  Q-RANDOM FILE STUFF
300541  LINE EDITOR
301062  ERRORS
301122  SECOND ERRORS
301102  THIRD ERRORS
301041  INITIALIZATION
SEG TOC 301342
MAIN PROGRAM 301401
```

LOGO™

Copywrite 1969
BBN®

LOGO STRING LANGUAGE RJG+PMW 7/29/1969 (LOGO,16)

```
RESTSU=73
IOPMAX=74
TISMAX=75
TTNO=76
TRAPPC=77
ERCODE=102
OWNWD=104
TTTSU=105
IOPTSU=106
FSA=107
STS=110
JMODE=111
ORG=123
TYIHNG=IOT 100
TYI=IOT 200
TYO=IOT 300
TIS=IOT 400
TOS=IOT 500
TTCKS=IOT 1000
SUPGO=IOT I 4100
TTMODE=IOT 1100
RSMC=IOT 1200
DELAY=IOT 1600
WPP=IOT 3200
GTD=IOT 3400
PEEK=IOT 3700
GRDR=IOT 4100
RPA=IOT 4200
RRDR=IOT 4300
GPUN=IOT 4400
PPA=IOT 4500
RPUN=IOT 4600
RCK=IOT 4700
RAI=IOT 6100
RAB=IOT 6140
EAI=IOT 6200
EAB=IOT 6240
WNIF=IOT 6300
WNIH=IOT 6320
WNBF=IOT 6340
WNBH=IOT 6360
SGI=IOT 5300
WAI=IOT 6400
WAB=IOT 6440
IVNR=IOT 6500
IVNW=IOT 6600
IVNRW=IOT 6640
IVNE=IOT 6660
```

```
EDIT=IOT 10100
RPB=IOT 10200
DNM=IOT 10600
SNM=IOT 11000
TDNUM=IOT 11200
STD=IOT 11300
DTM=IOT 11415
DDT=IOT 11515
HALT=IOT 12401

RAIS=RAI"U"2
RAIL=RAI"U"2
WAIP=WAI"U"22
RAIP=RAI"U"2
WNIP=WNIH"U"2
WNIL=WNIF"U"2
WAIL=WAI"U"2
SGIL=SGI"U"2
EAIL=EAI"U"2
RAIFL=RAI"U"2
WAIFL=WAI"U"2
SGIFL=SGI"U"2
WNIFL=WNIF"U"2
EAIFL=EAI"U"2
SEGNUM=276
PASNUM=300
NAMNUM=301

BCNT=20
BLNG=98.
PDLNG=20000.
&L
```

```
DEFINE SJSP A
          JSP .SJSP
           A
TERMINATE SJSP

DEFINE SJMP A
          JDA .SJMP
           A
TERMINATE SJMP

DEFINE LODE A
          LIO A
          LAC (JMP-1)
TERMINATE LODE

DEFINE STORE A,B
          LIO I A+1
          DIO B
          LAC A
          SUB A+5
          DAC B+1
TERMINATE STORE

DEFINE LOAD A
          LIO A
          LAC A+1
TERMINATE LOAD

DEFINE UNLOAD A
          DIO A
          DAC A+1
TERMINATE UNLOAD

REPEAT ØIF P.EQUALS LDCOM,NULL
DEFINE REMLD A,B
EQUALS B, LDCOM
DEFINE LDCOM
EQUALS LDCOM,B
EXPUNGE B
LDCOM
TEXT !A#!
TERMINATE LDCOM
TERMINATE REMLD

DEFINE COLON A
          A=SEGNO"T"I+.
TERMINATE COLON
&L
```

```
DEFINE UNSTEP A
          LAC A
          IDC
          IDC
          SUB (1)
          DAC A
TERMINATE UNSTEP

DEFINE USED A
          LAC (010000)
          IOR I A+4
          DAC I A+4
TERMINATE USED

DEFINE UNREAD A
          LAC A+4
          JDA RUNFRZ
TERMINATE UNREAD

DEFINE NEWSEG COMMENT/C
          REPEAT 1IF VP .-CON,CON=.
          REPEAT 1IF P,PRINT !COMMENT!
          REPEAT 0IF P,REMLD COMMENT,C
          SEGNO=SEGNO+1
          SEG/            CON-.
           0
TERMINATE NEWSEG

EQUALS HERE,NULL
DEFINE REMOTE A/B
EQUALS B,HERE
DEFINE HERE
EQUALS HERE,B
EXPUNGE B
HERE
A
TERMINATE HERE
TERMINATE REMOTE

DEFINE SYMBOL NAME,DATA/A,B
A,        0
          TEXT /NAME#/
          DATA
B,        A/              B-A
B/        EXPUNGE A,B
TERMINATE SYMBOL

SEGNO=0
REPEAT 0IF P,CON=0

/SYMBOLS TEMPORARILY DEFINED : TO BE CHANGED LATER
CNAME=000001
CTHING=000002
&L
```

```
100/        JMP LOADER
OWNWD/      TEXT .LOG.
ORG/        /LOADER FOR LOGO (PLAGERIZED FROM STRINGCOMP)
TOCBK,      . 50./
DATABK,     . 50./
TOCP,       0
DATAP,      0

/SUBROUTINES AND STORAGE FIRST SO CAN EASILY ZERO MOST OF LOADER
TGWORD,     DAP TGWX          /GET NEXT WORD FROM BINARY FILE
            LAW DATABK 50.    /ARE WE OFF TOP OF DATA BLOCK
            SAS DATAP
            JMP GWORD2        /NO
            LAW TOCBK 50.     /YES: ARE WE OFF TOP OF TOC
            SAS TOCP
            JMP GWORD1        /NO
            LAW TOCBK         /YES: GET NEXT TOC BLOCK
            LIO TOCBK
            RAB 2
            LAW TOCBK 1       /RESET POINTERS
            DAC TOCP
GWORD1,     LIO I TOCP        /GET NEXT DATA B
            IDX TOCP
            LAW DATABK
            DAC DATAP
            RAB 2
GWORD2,     LIO I DATAP
            IDX DATAP
TGWX,       JMP .

T21,        DAP T22           /TYPE OUT NUMBER
            LAI
            LIO TCC
            SNM+43
            CLA
            TYO               /TYPE 2 SPACES
            TYO
            LCH I TXTCOM
            SAD TEOM
            JMP .+3
            TYO
            JMP .-4
            LAC TJMP
            IOR TXTCOM
            DAC TXTCOM
            LAC TCD
            TYO 1
TCD,        760000
T22,        JMP .
```

```
TEOM,       CHARACTER L#
TJMP,       JMP
TXTCOM,     Ø
            TEXT /COMMAND TABLE#/
            TEXT /SYSTEM TABLE#/
            TEXT /LIST TABLE#/
            REPEAT 1IF P,LDCOM
            REPEAT 1IF P,[REPEAT 4,CHARACTER L#]
TXTCOM+2ØØ./
&L
```

```
TXT5,     TEXT .SEG TOC #.
TXT6,     TEXT .DRA IF DRUM: #.
TXT2,     TEXT .CHECKSUM ERROR#.
TXT4,     TEXT .MAIN PROGRAM #.
TCRPB,    RPB
TCA,      DZM CON
TCG,      JSP TGWORD
T16,      SEGNUM
          TEXT .LOG.
           Ø

T3Ø,      Ø
T36,      Ø
T32P,     Ø

T32,      98.
          REPEAT 97.,Ø

T31,      Ø
          DZM T36
          DAP T37
          LAW T31B        /OUTPUT COMTBL, SYSTBL, ETC.
          DAP T35X
T31B,     LAW T32+3
          DAC T32P
T31A,     LAC I T31
          DAC I T32P
          SZA I
          JMP T37         /END OF LIST
          IDX T31
          IDX T32P
          SAD T39
          JMP T35         /END OF BUFFER
          JMP T31A
&L
```

```
T35,    LAW T32
        DZM T32+2
        WNIL+1          /WRITE OUT A BUFFER OF TABLE
        Ø
        LAC T36
        DIO T36
        SZA I
        JMP T38
        LIA
        LAW T32
        RAIL+1
        Ø
        LAC T36
        DAC T32+2
        LAW T32
        WAIL+1
        Ø
T35X,   JMP T318

T37,    LAW .           /RETURN ADR DAPPED INTO HERE
        DAP T35X
        JMP T35

T38,    DIO I TV1
        IDX TV1
        JSP T21         /TYPE OUT ADR OF TABLE
        JMP I T35X

T39,    T32+98.
&L
```

```
DEFINE C NAME,DATA,BITS/A,B
A,          Ø
            TEXT /NAME#/
            DATA+7ØØØØØ
B,          A/              BITS"T"1ØØØØ+B-A
B/          EXPUNGE A,B
TERMINATE C



/4Ø←WHEEL ONLY COMMAND
/2Ø←(SOME PARSING INFO, NOT USED NOW)
/1Ø←LEGAL AS STORED COMMAND
/4←LEGAL AS DIRECT COMMAND
T33,        C TO,TO,4
            C CALL,MAKE,14
            C RETURN,RETURN,1Ø
           —C HOARD,HOARD,54
           —C SHARE,SHARE,54
            C TITLE,TITLE,14
           -C BURY,BURY,54
            C WAIT,WAIT,14
          —C DIGUP,DIGUP,54
            C IF,IF,14
           —C UNLOCK,UNLOCK,14
          —C LOCK,LOCK,14
            C EDIT,.EDIT,4 /EDIT ILLEGAL IN ANY FORM FROM FILE
            C END,END,14
            C TRACE,TRACE,14
            C ERASE,ERASE,14
            C LIST,LIST,14
            C GOODBYE,GOODBYE,14
            C PRINT,PRNT,14
            C TYPE,TYPE,14
            C OUTPUT,RETURN,1Ø
            C MAKE,MAKE,14
            C STOP,STP,1Ø
            C DO,SUPDO,14
            C LOCAL,LOCAL,1Ø
            C SAVE,SAVE,14
            C GET,GET,14
            C GO,GO,1Ø
            C DDTBBN,DDTA,54
            C RESET,RESET,14
            C ABBREVIATE,ABBT,14
            C TEST,TEST,14
            C PASSWORD,PSWORD,14
/"Q"COMMANDS ARE FOUND ONLY IN READING FROM FILES
            C "Q"TO,GTTO,4
            C "Q"BTO,GTBTO,4            /HIDDEN PROCEDURRE
            C "Q"END,GTEND,4
            C "Q",GTFINI,4
            Ø
&L
```

```
DEFINE S NAME,DATA
A,          Ø
            TEXT /NAME#/
            6ØØØØØ DATA
B,          A/              B-A
B/          EXPUNGE A,B
TERMINATE S

DEFINE SØ NAME,DATA
A,          Ø
            TEXT /NAME#/
            5ØØØØØ DATA
B,          A/              B-A
B/          EXPUNGE A,B
TERMINATE SØ

T34,        S FIRST,FIRST
            S EMPTYQ,EMPTQ
            S ZEROQ,ZEROQ
            S BEFOREP,BEFOREP
            S EMPTYP,EMPTQ
            S ZEROP,ZEROQ
            S DATE-GOTTEN,DATEGT
            S SIZE,SIZE
            S INITIALS,FINIT
            S OWNER,OWNER
            S WORDP,WORDQ
            S DATE-SAVED,DATESV
            S SENTENCEP,SENTQ
            S NUMBERP,NUMQ
            S GREATERP,GREATQ
            S IS,IS
            S ENTRIES,ENTRIES
            S BUTFIRST,BUTF
            S LAST,LAST
            S BUTLAST,BUTL
            S COUNT,COUNT
            S THING,THING
            S WORDQ,WORDQ
            S SENTENCEQ,SENTQ
            S NUMBERQ,NUMQ
            S WORD,WRD
            S MAXIMUM,MAXIMUM
            S SENTENCE,SENT
            S SUM,SUM
            S ASK,ASK
            S MINIMUM,MINIMUM
            S GREATERQ,GREATQ
            S BOTH,BOTH
            S EITHER,OR
```

```
S OR,OR          /TEMPORARY
S DIFFERENCE,DIFF
SØ ABBREVIATIONS,ABBRS
SØ ABBREVIATION,ABBR
SØ ALL,ALL
SØ AND,CAND
SØ COMMENT,COMMENT
SØ PROCEDURES,PRCDS
SØ NAMES,NAMES
SØ ENTRY,ENTRY
SØ FILE,FILE
SØ FILES,FILES
SØ CONTENTS,CONTENTS
SØ REQUEST,REQUEST
SØ CLOCK,CLOCK
SØ TIME,TIME
SØ DATE,DATE
SØ RANDOM,RANDOM
SØ AS,AS
SØ TRACES,TRACES
SØ LINE,LINE
SØ OF,OF
SØ ON,ON
SØ TRUE,ATRUE
SØ FALSE,AFALSE
Ø
```

```
&L
```

```
DEFINE LC TYPE,NAME,ADDR/A,B,C
          C=0
IRPC[,,NAME]
          C=C+1
          REPEAT 1IF VZ C-3,C=0
ENDIRPC
          TYPE"T"100000 ADDR
A,        0
          TEXT /NAME#/
B,        A/              B-A+1IF VP C-1/"T"200000+1IF VP C-2/"T"100000
B/        REPEAT 1IF P,EXPUNGE A,B,C
TERMINATE LC


T40,      LC 7,IF,IF      /COMMANDS TO BE LISTED (ALL COMMANDS)
          LC 7,WAIT,WAIT
          LC 7,OUTPUT,RETURN
          LC 7,UNLOCK,UNLOCK
          LC 7,LOCK,LOCK
          LC 7,HOARD,HOARD
          LC 7,SHARE,SHARE
          LC 7,DDTBBN,DDTA
          LC 7,TITLE,TITLE
          LC 7,BURY,BURY
          LC 7,DIGUP,DIGUP
          LC 7,DO,SUPDO
          LC 7,EDIT,,EDIT
          LC 7,END,END
          LC 7,TRACE,TRACE
          LC 7,ERASE,ERASE
          LC 7,LIST,LIST
          LC 7,GOODBYE,GOODBYE
          LC 7,MAKE,MAKE
          LC 7,SAVE,SAVE
          LC 7,GET,GET
          LC 7,PRINT,PRNT
          LC 7,RETURN,RETURN
          LC 7,TYPE,TYPE
          LC 7,TEST,TEST
          LC 7,RESET,RESET
          LC 7,ABBREVIATE,ABBT
          LC 7,PASSWORD,PSWORD
          LC 7,STOP,STP
          LC 7,LOCAL,LOCAL
          LC 7,TO,TO
          LC 7,GO,GO
          LC 6,IS,IS
          LC 6,BOTH,BOTH
          LC 6,ASK,ASK
          LC 6,BEFOREP,BEFOREP
&L
```

```
LC 5,AND,CAND
LC 6,ENTRIES,ENTRIES
LC 5,TRACES,TRACES
LC 5,OF,OF
LC 5,TRUE,ATRUE
LC 5,FALSE,AFALSE
LC 6,FIRST,FIRST
LC 6,BUTFIRST,BUTF
LC 6,LAST,LAST
LC 6,INITIALS,FINIT
LC 5,LINE,LINE
LC 6,BUTLAST,BUTL
LC 6,COUNT,COUNT
LC 6,DATE-GOTTEN,DATEGT
LC 6,SIZE,SIZE
LC 6,OWNER,OWNER
LC 6,THING,THING
LC 6,WORDP,WORDQ
LC 6,SENTENCEP,SENTQ
LC 6,NUMBERP,NUMQ
LC 6,EMPTYP,EMPTQ
LC 6,ZEROP,ZEROQ
LC 6,WORD,WRD
LC 6,GREATERP,GREATQ
LC 6,DATE-SAVED,DATESV
LC 6,MAXIMUM,MAXIMUM
LC 6,EITHER,OR
LC 6,MINIMUM,MINIMUM
LC 6,SENTENCE,SENT
LC 6,SUM,SUM
LC 6,DIFFERENCE,DIFF
LC 5,ABBREVIATIONS,ABBRS
LC 5,CONTENTS,CONTENTS
LC 5,NAMES,NAMES
LC 5,ENTRY,ENTRY
LC 5,FILE,FILE
LC 5,FILES,FILES
LC 5,ALL,ALL
LC 5,COMMENT,COMMENT
LC 5,PROCEDURES,PRCDS
LC 5,ABBREVIATION,ABBR
LC 5,REQUEST,REQUEST
LC 5,RANDOM,RANDOM
LC 5,CLOCK,CLOCK
LC 5,TIME,TIME
LC 5,DATE,DATE
LC 5,ON,ON
LC 5,AS,AS
Ø
```

&L

```
LOADER,   CLF 7
          LAW TXTCOM+1
          DAC TXTCOM
          LAW TXT7          /START OF LOADER
          TOS
          LAW TXT6
          TOS               /"DRA IF DRUM"
          LAW TOCBK         /TYPE INTO EXTRA SPACE
          DAC FSA
          TIS
          LAW TOCBK
          EDIT
          JMP LOADER
          DNM 16            /TYPE IN OCTAL NUMBER WITH NO SIGN
          STF 3             /OR OTHERWISE ASSUME PAPER TAPE
          LIA
          LAC TCD
          TYO
          LAC TCG           /CHOSE INSTRUCTION FOR WHICH TYPE OF FILE
          SZF 3
          LAC TCRPB
          DAC T1
          SZF I 3           /IF PAPER TAPE GET READER
          JMP .+3
          GRDR 40
           XX
          DIO T30           /SAVE IO
          LAW T33           /COMTBL
          JDA T31
          LAW T34           /SYSTBL
          JDA T31
          LAW T40           /LIST TABLE
          JDA T31
          LIO T30
          LAW TOCBK         /READ IN FIRST TOC BLOCK IF DRUM
          SZF I 3
          RAB 2
          LAW TOCBK 2       /AND SET POINTERS
          DAC TOCP
          LAW DATABK 50.    /AND SET SO DATA BLOCK WILL AUTOMATICALLY LOAD
          DAC DATAP
          STF 4             /IF FROM DRUM: SKIP TO AFTER FIRST JMP BLOCK
          SZF 3
T2,       CLF 4             /BEGINNING OF LOADER PROPER
          LAW SEG           /ZERO OUT SEGMENT AREA
          DAP .+1
          DZM .
          IDX .-1
          SAS TCA
          JMP .-3
          CLF 1             /SET TO BE LOADING SEGMENTS
&L
```

```
T1,       .             /GET FIRST WORD
          DIO T4        /FIRST ADDRESS
          DIO T5        /CHECK SUM
          SPI           /JUMP BLOCK?
T4,       Ø             /IF SO DISPATCH
          XCT T1        /READ ADDRESS OF LAST WORD+1
          DIO T6
          LAI
          ADD T5        /RESET CHECKSUM TO INCLUDE IT
          DAC T5
          CLA
          DIP T4        /SET SO USING ONLY 12 BIT ADDRESS
          DIP T6
T7,       XCT T1        /GET NEXT WORD
          SZF I 4
          DIO I T4      /STORE IF NOT SKIPPING
          LAI
          ADD T5        /AND UPDATE THE CHECKSUM
          DAC T5
          IDX T4        /SET FOR NEXT WORD
          AND TC7777
          SAS T6        /WAS THIS THE LAST WORD
          JMP T7        /NO: LOOP
          XCT T1        /YES: CHECK CHECKSUM
&L
```

```
          LAI
          SAD T5
          JMP T1          /OK
          LAW TXT2        /"CHECKSUM ERROR"
          TOS 1
T6,        Ø
          RRDR            /RELEASE READER AND QUIT
          JMP T15

T11,      STF 1           /SEGMENT LOADER : FLAG 1 SAYS MAIN SEGMENT
T8,       LAW 1Ø          /WAIT 1Ø SECONDS IF FROM TAPE
          SZF 3
          DELAY
          LAW SEG
          WNIL 1
T5,        Ø              /WRITE OUT SEGMENT
TV1,      DIO COMTBL      /AND SAVE DRA'S
          DIO SEGDRA
          IDX TV1
          JSP T21         /TYPE OUT NUMBER ALSO
          SZF I 1
          JMP T2          /GO GET REST OF SEGMENTS
          LAW TEME        /DONE: WRITE OUT SEGMENT TOC
          WNIL 1
TC7777,    7777
          LAW T16
          IVNRW+1
           Ø
          LAW TXT5        /TYPE OUT ADDRESS OF SEGMENT TOC
          TOS
          JSP T21
          LAW 36          /WRITE OUT MAIN SEGMENT
          WNIL 1
T12,       Ø
&L
```

```
           DIO T14
           LAW TXT4
           TOS 1
TCC,        6
           JSP T21
TCC+2,     RRDR
           LAW TOCBK        /ZERO OUT AREA OCCUPIED BY LOADER
           DAP .+1
T13,       DZM .
           IDX .-1
           SAS TCB          /FEW WORDS STILL THERE
           JMP T13
           LAW 36
           LIO T14
           DIO 42
           WAIL             /REWRITE MAIN SEGMENT WITH AC AND IO
T15,       LAW TXT7
           TOS
           DZM 66           /HALT IN DDT WITHOUT TYPEOUT
           JMP 66

TCB,       DZM T13
T14,       Ø
TXT7,      771477
           157715
           771577
           157715
           767674

WORD JMP T2
&L
```

```
71/         JMP GDBYEA      JMP GDBYEA
100/        JMP LOGO        JMP ERRS
RESTSU/     JMP MARKBK      /DZM BRKCNT
TTTSU/      CAL BRKERR
IOPTSU/     CAL IOPERR
ORG/        BASE.
ORG+BLNG"T"BCNT/
/FIND BUFFER TO USE AND SET UP POINTERS (TRANSPARENT TO IO)
FBUF.       DAP FITMX
            DIO DRUMI
            LAC (10000)     /SET TO LARGER THAN POSSIBLE
            DAC DRUMT
            LAW BCHK
            DAP FBUFA+1     /SEARCH COUNT TABLE FOR OLDEST REFERENCE
FBUFA.      LAW 7777        /GET COUNT OF NEXT DRA
            AND .
            SUB DRUMT       /ARE WE LESS THAN PREVIOUS
            SMA
            JMP FBUFB
            LAC I FBUFA+1   /YES!-IS THIS LOCKED IN CORE
            AND (760000)
            SZA
            JMP FBUFB       /YES!-SO FORGET IT
            LAW 7777        /NO!-GET ORIGINAL COUNT
            AND I FBUFA+1
            DAC DRUMT       /RESET NEW LIMIT
            LAW I BCHK-BTBL
            ADD FBUFA+1     /RESET DRA WHICH WILL HAVE TO GO
            DAP BDRA
FBUFB.      IDX FBUFA+1     /LOOK AT NEXT COUNT
            SAS (AND BCHK+BCNT)
            JMP FBUFA       /LOOK TA NEXT NUMBER
            JMP FITMC       /DONE-SO HAVE FOUND LEAST
&L
```

```
/FIND ITEM WHOSE DRA IS IN IO (TRANSPARENT TO IO)
/R1-NOT IN CORE,R2-IN CORE AND POINTERS SET
FITM,     DAP FITMX
          DIO DRUMI
          LAW BTBL-1      /SEARCH TABLE FOR DRUM ADDRESSES
          DAP BDRA
FITMA,    IDX BDRA
          SAD (SAS BTBL+BCNT)         /OFF END OF TABLE
          JMP FITMX-1     /YES!-GIVE R1 FOR DRA NOT FOUND
          LAI
BDRA,     SAS .           /WILL POINT AT DRA AT END
          JMP FITMA
          IDX FITMX       /HAVE FOUND DRA SO R2
FITMC,    LAW BCHK-BTBL
          ADD BDRA
          DAP BCOUNT      /SET POINTER TO COUNT TABLE
          SUB (SAS BCHK) /CALCULATE ADDRESS OF BUFFER
          MUL (BLNG)
          SCL 9S
          SCL 8S
          ADD (BASE)
          DAC BBPTR
          ADD (2)
          DAC BBPTR1      /TO FWD PTR
          IDA
          DAC BBPTR2      /TO FIRST WORD
          LIO DRUMI
FITMX,    JMP .
&L
```

```
/GET ITEM IN CORE (DRA IN IO) (C(IO)=Ø MEANS FORCE ITEM OUT)
          CLI             /JSP GITM+1 FORCES ITEM OUT OF BUFFER
GITM,     DAP GITMX
          DIO DRUMT
          LAC I BDRA      /IS THE BLOCK ALREADY THERE?
          SAD DRUMT
          JMP GITMB       /YES!-SKIP THE READING JAZZ
          LIA
          LAC I BCOUNT    /IS THE BLOCK CHANGED
          AND (10ØØØ)
          SZA I
          JMP GITMA       /NO!-SO DON'T WRITE OUT
          LAC BBPTR       /YES!-SO WRITE IT OUT
          SNI I
          WAIP
          DZM I BCOUNT
          DZM I BDRA
GITMA,    LIO DRUMT       /READ IN BLOCK WANTED
          LAC BBPTR
          SNI I
          RAIP
          DIO I BDRA
GITMB,    IDX BTIME       /UPDATE QUEING ALGORITHM
          DAP I BCOUNT    /AND SET FOR THIS WORD
          SAS (10ØØØ)     /DID WE OVERFLOW COUNT
GITMX,    JMP .
          LAW BCHK        /YES!-SET ALL COUNTS TO Ø
          DAP GITMC+1
GITMC,    CLA
          DAP .
          IDX .-1
          SAS (DAP BCHK+BCNT)
          JMP GITMC
          DZM BTIME
          JMP GITMX

/GET ITEM (Ø MEANS GET NEW ITEM)
          CLI
GETIT,    DAP GETITX
          JSP FITM        /IS ITEM IN CORE
          JSP FBUF        /NO!-GO FIND BUFFER
          JSP GITM        /READ IN IF NECESSARY
GETITA,   LAC BBPTR       /GET ADDRESS OF BEGINNING OF BUFFER
          SNI I
GETITX,   JMP .
          DZM I BBPTR1    /Ø THE LINKING WORD
          DZM I BBPTR2    /Ø FIRST WORD
          JDA WNPCNT      /WNIP AND COUNT ITEMS TO ASSURE < 4 QTRKS
          LAC BBPTR
          DIO I BDRA      /SET UP POINTER IN TABLE
          JMP GETITX
&L
```

```
NEWITM,    DAP NEWITX      /GET NEW ITEM INTO SAME BUFFER
           JSP GETIT       /SET UP POINTERS (C(AC)=C(BBPTR))
           JDA WNPCNT      /WRITE NEW ITEM AND COUNT SO ONLY 4 QTRKS
           DIO I BBPTR1    /SET FORWARD CHAIN WORD
           JSP MARK        /MARK OLD BLOCK AS CHANGED
           JSP GITM-1      /FORCE OLD ITEM OUT OF CORE
           LIO I BBPTR1    /FORCE NEW ITEM TO BE ONE IN CORE
           DIO I BDRA
           JSP MARK        /MARK IT CHANGED ALSO
           DZM I BBPTR1
           DZM I BBPTR2
NEWITX,    JMP .

RUNFRZ,    Ø               /POINTER IN AC : READ UNFREEZE
           DAP RUFRZX
RUNFRZ+2,  CLA
           SAD RUNFRZ
           JMP RUFRZX
           LAC (-4ØØØØØ)
           AND I RUNFRZ
           DAC I RUNFRZ
RUFRZX,    JMP .

/MARK PRESENT ITEM AS USED
MARK,      DAP .+4
           LAC (1ØØØØ)
           IOR I BCOUNT
           DAC I BCOUNT
           JMP .

WNPCNT,    Ø               /ROUTINE TO WRITE ON HELD QTRKS AND
           DAP WNPCNX      /ASSURE THAT NO MORE THAN FOUR
           LAC WNPCNT      /QUARTER TRACKS ARE HELD
           WNIP
           IDX ITMCNT
           SAD (1ØØØØ)
           CAL QTRERR      /YOU LOSE AND HALT
WNPCNX,    JMP .
&L
```

```
/ROUTINE TO SET UP BUFFER
/C(AC) IS RELATIVE CHARACTER POINTER INTO BLOCK
/C(IO) IS DRA OF THE BLOCK
/WORD FOLLOWING JDA IS ADDRESS OF SIX WORD BLOCK CONSISTING OF :
/TEXT POINTER,DRUM ADDRESS,BOTTOM OF BUFFER,TOP OF BUFFER, POINTER TO
/TABLE OF COUNTS,POINTER TO BEGINNING OF TEXT BUF.
SETUP,    0
          DAP SETUPX
          LAC I SETUPX    /GET ADDRESS OF FIVE WORD BLOCK
          DAP SETUPA
          JSP GETIT       /GET ITEM DESIRED INTO CORE
          LAC BBPTR2
          ADD SETUP       /CALCULATE PHYSICAL TEXT POINTER
SETUPA,   DAC .           /AND SAVE IN TEXT POINTER
          IDX SETUPA
          LAC BDRA
          DAC I SETUPA
          IDX SETUPA
          LAC BBPTR1      /SET EXT POINTER TO BOTTOM OF BUFFER
          IOR (600000)
          DAC I SETUPA
          IDX SETUPA
          LAC BBPTR       /SET POINTER TO TOP OF BUFFER
          ADD (JMP+BLNG-1)
          DAC I SETUPA
          IDX SETUPA
          LAC I BCOUNT
          IOR I SETUPX
          DIP I BCOUNT
          IDX SETUPX
          LAC BCOUNT      /SAVE POINTER TO CHECK TABLE
          DAC I SETUPA
          IDX SETUPA
          LAC BBPTR2
          DAC I SETUPA
SETUPX,   JMP .

MARKBK,   DZM BRKCNT      /MARK EXISTENCE OF BREAK
          RSMC+3          /AND DEBREAK

CHKBRK,   DAP CHKBRX      /HAS THERE BEEN A BREAK
          CLA
          SAD BRKCNT      /0 IF BREAK
          CAL BRKERR
CHKBRX,   JMP .
&L
```

```
GETPDL,   DAP GETPDX      /SET UP POINTERS FOR PUSH-DOWN
          JSP GETIT       /GET DESIRED ITEM INTO CORE
          DIO PDLDRA
          LAC (-40000)    /UNFREEZE PREVIOUS ITEM
          AND I PLCOUNT
          DAC I PLCOUNT
          LAC BCOUNT
          DAC PLCOUNT     /SET FREEZE PTR TO THIS ITEM
          LAC (40000)     /FREEZE THIS ONE
          IOR I PLCOUNT
          DAC I PLCOUNT
          LAC BBPTR1
          DAP PUSHF       /RESET FORWARD POINTER
          IDA
          DAP PUSHG       /RESET BACK POINTER
          IDA
          DAP PUSHP       /RESET POINTER TO PRESENT LOCATION
          ADD (BLNG-4)
          DAP PUSHT       /RESET POINTER TO TOP
          SUB (1)         /RETURN IN AC FOR PULL ROUTINE
GETPDX,   JMP .

PUSH,     0               /PUSH DOWN ROUTINE (TRANSPARENT TO AC)
          DAP PUSHX
          DIO PUSHI
PUSHA,    LAC I PUSHX
          DAP PUSHB
          AND (770000)
          SZA
          JMP PUSHC       /NOTHING TO PUSH
          LAC PUSHP       /ARE WE OFF TOP OF BUFFER?
          SAS PUSHT
          JMP PUSHD       /NO
PUSHF,    LIO .           /YES- IS THE FORWARD PTR 0
          SNI I
          JMP PUSHD-1
          JSP GETIT-1     /GET A NEW ITEM
          DIO I PUSHF     /SET UP FORWARD POINTER
          LAC PDLDRA
          DAC I BBPTR2    /AND BACK POINTER
PUSHD-1,  JSP GETPDL      /SET UP POINTERS FOR NEW
PUSHD,    LAC (10000)     /MARK THIS BLOCK USED
          IOR I PLCOUNT
          DAC I PLCOUNT
          IDX PDLCNT      /KEEP COUNT OF ENTRIES ON PDL
          SAD (PDLNG)     /IS PDL TOO LONG
          CAL PDLERR
PUSHB,    LAC .           /GET WORD AND STORE IT
PUSHP,    DAC .
          IDX PUSHX
          IDX PUSHP
          JMP PUSHA
```

```
PUSHC,     LAC PUSH
           LIO PUSHI
PUSHX,     JMP .

PUSHG,     DAC .           /PTR TO BACK PTR
&L
```

```
PULL,     0
          DAP PUSHX
          DIO PUSHI
PULLA,    LAC I PUSHX
          DAP PULLB       /GET ADDRESS OF WHERE TO PUT
          AND (770000)    /IS THERE ANYTHING MORE TO PUT
          SZA
          JMP PULLC       /NO
          LAW I 1         /YES-BACK UP COUNT 1
          ADD PDLCNT
          DAC PDLCNT
          LIO I PUSHG     /BACK PTR TO IO (USED IF OFF BOTTOM)
          LAW I 1         /BACK UP PDL PTR 1
          ADD PUSHP
          SAD PUSHG       /ARE WE OFF BEGINNING OF BUFFER
          JSP GETPDL      /YES-GO GET PREVIOUS BLOCK
          DAP PUSHP
          LAC I PUSHP
PULLB,    DAC .
          IDX PUSHX
          JMP PULLA

PULLC,    LAC PULL
          JMP PUSHC+1

ACPUSH,   0
          JDA PUSH
           ACPUSH
          JMP PUSH+1

ACPULL,   DAP .+3
          JSP PULL+1
           PULL
          LAW .
          JMP PULL+1

PDLCLR,   DAP PDLCLX      /CLEAR PDL: CLEAR TO POSITION IN IO
          LAI
          SAD PDLCNT
PDLCLX,   JMP .
          JDA PULL
           TEMA
          JMP PDLCLX-1

SGET,     DAP SGETX       /ROUTINE TO GET CHARACTER FROM SYMBOL
          IDX SGETP       /STORED WITH FIRST 15. CHARACTERS IN
          SUB (LAC SYM+5)             /SYM-SYM+5
          SMA
          JMP SGETA       /NOW LOOKING IN VIRTUAL MEMORY FOR CHARACTER
          DZM FWORDP+4    /MARK THAT STILL LOOKING IN CORE
SGETP,    LAC .
SGETX,    JMP .
&L
```

```
SGETA,    SZA             /IS THIS FIRST TIME TO LOOK AT DRUM
          JMP SGETB       /NO
          LODE NDRA       /YES. SET UP REGISTERS
          JDA SFWORD
SGETB,    JSP FWORD
          JMP SGETX

GWORD,    DAP GWORDX      /GET A WORD FOR SEARCH ROUTINES
          LAC GWORDP
          SAD GWORDP+3
GWORDF,   JSP GWORDA
          IDX GWORDP
GWORDC,   LAC I GWORDP
GWORDX,   JMP .

NTHWD,    0
          DAP GWORDX
          LAC NTHWD       /COMPARE TOP AND C(GWORDP)+N
          AND (7777)
          ADD GWORDP
          SUB GWORDP+3
          SPO
          JMP NTHWDA      /REMAIN WITHIN THIS BLOCK
          DAC NTHWD       /RESET N AND GO TO NEXT BLOCK
          LAW NTHWD+2
GWORDA,   DAP GWORDB      /GET NEXT BLOCK FOR GWORD
          JSP UGWORD
          LIO I GWORDP+2
          SNI I
          JMP GWORDE
          LIO I GWORDP+1 /DRA
          JSP NEWITM
GWORDE,   LAC (JMP-1)
          JDA SGWORD
GWORDD,   NOP
GWORDB,   JMP .

NTHWDA,   ADD GWORDP+3
          DAC GWORDP
          JMP GWORDC

USEDG,    DAP USEDGX      /MARK GWORDP AS USED
          LAC (010000)
          IOR I GWORDP+4
          DAC I GWORDP+4
USEDGX,   JMP .
&L
```

```
GWORDS,   Ø
          DAP GWRDSX
          JSP GWORD
          JSP USEDG
          LAC GWORDS
          DAC I GWORDP
GWRDSX,   JMP .

SGWORD,   Ø               /SET UP GWORDP
          DAP SGWRDX
          LAC SGWORD
          JDA SETUP
           4ØØØØØ GWORDP
SGWRDX,   JMP .

RGWORD,   DAP RGWRDX      /CALCULATE RELATIVE POINTER
          LIO I GWORDP+1
          LAC GWORDP
          SUB GWORDP+5
RGWRDX,   JMP .

UGWORD,   DAP RUFRZX      /FREE GWORDP
          LAC GWORDP+4
          DAC RUNFRZ
          JMP RUNFRZ+2

FWORD,    DAP FWORDX      /ANOTHER READ ROUTINE
          LAC FWORDP
          SAD FWORDP+3
          JSP FWORDA
FWORDB,   IDX FWORDP
          LAC I FWORDP
FWORDX,   JMP .

FWORDA,   DAP FWORDD
          JSP UFWORD
          LIO I FWORDP+2
          SNI I
          JMP FWORDE
          LIO I FWORDP+1
          JSP NEWITM
FWORDE,   LAC (JMP-1)
          JDA SFWORD
FWORDD,   JMP .
```

```
SFWORD,   0               /SET FWORDP
          DAP SFWRDX
          LAC SFWORD
          JDA SETUP
           200000 FWORDP
SFWRDX,   JMP .

UFWORD,   DAP UFWRDX      /RELEASE FWORDP
          CLA
          SAD FWORDP+4
          JMP UFWRDX
          LAC (-200000)
          AND I FWORDP+4
          DAC I FWORDP+4
UFWRDX,   JMP .

USEDF,    DAP USEDFX      /MARK FWORDP AS USED
          LAC (010000)
          IOR I FWORDP+4
          DAC I FWORDP+4
USEDFX,   JMP .

FWORDS,   0               /AUXILLARY ROUTINE TO STORE THRU FWORDP
          DAP FWORDT
          JSP FWORD
          JSP USEDF
          LAC FWORDS
          DAC I FWORDP
FWORDT,   JMP .
&L
```

```
TLINE,    DAP SLINEX
          LIO CHARNO
          SNI
          JMP SLINEX
SLINE,    DAP SLINEX
          LAC (760000)
          TYO
          DZM CHARNO
SLINEX,   JMP .
TWORD,    DAP TWORDX      /GET NEXT WORD FROM A COMMAND
          LAW I 1         /DECREMENT COUNT LEFT
          ADD NPTR
          DAC NPTR
          LAC TEXTP
          SAD TEXTP+3
          JSP TWORDA
          IDX TEXTP
TWORDB,   LAC I TEXTP
TWORDX,   JMP .

GNS,      DAP GNSX        /GET NEXT SYMBOL: FORGET COMMENTS
          JSP TWORD
          AND (700000)
          SAD (100000)
          JMP GNSA        /COMMENT
          LAC I TEXTP     /NOT COMMENT
GNSX,     JMP .
GNSA,     JSP EVALG
          JMP GNS+1

GNST,     0
          DAP GNSV        /SKIP OVER N WORDS
          LAC NPTR        /UPDATE REGISTERS LEFT COUNTER
          SUB GNST
          DAC NPTR
GNSB,     LAC TEXTP
          SUB TEXTP+3
          ADD GNST
          SPQ
          JMP GNSC
          DAC GNST
          LAW GNSB
TWORDA,   DAP TWORDC
          JSP UTWORD
          LIO I TEXTP+2
          SNI I
          JMP TWORDD
          LIO I TEXTP+1
          JSP NEWITM
TWORDD,   LAC (JMP-1)
          JDA STWORD
          LAC TEXTP
TWORDC,   JMP .
&L
```

```
GNSC,       ADD TEXTP+3
            DAC TEXTP
GNSV,       JMP .

UTWORD,     DAP UTWRDX      /RELEASE TEXTP
            LAC (-100000)
            AND I TEXTP+4
            DAC I TEXTP+4
UTWRDX,     JMP .

STWORD,     0
            DAP STWRDX
            LAC STWORD
            JDA SETUP
             100000 TEXTP
STWRDX,     JMP .
&L
```

```
/DRUM SEARCH ROUTINE
/R1 MEANS ENTRY IS NOT IN TABLE
/R2 MEANS ENTRY IS IN TABLE AND VARIABLES SET UP

DLOOK,    DAP DLOOKX
          LAC (JMP-1)
DLOOKF,   JDA SGWORD
          JMP DLOOKH

DLOOKB,   LAI             /GET NEXT RELATIVE COUNT
          SUB (1)
          JDA NTHWD
DLOOKH,   JSP RGWORD
          DIO SDRA
          DAC SDRA+1
          JSP GWORD
          DIP POINT       /SAVE TOP BITS
          AND (7777)
          SZA I           /ARE WE DONE
          JMP DLOOKG      /YES
          LIA             /RELATIVE COUNT TO IO IN CASE FAIL
          SUB (3)         /COMPARE WE WORD COUNT OF SYMBOL
          SZF 6           /SET MEANS ONLY ONE INFO WORD
          IDA
          SAS WRDCNT
          JMP DLOOKB      /NOT SAME LENGTH
DLOOKA,   LAW SYM-1       /PREPARE TO COMPARE WORDS
          DAP SGETP
          DZM TEMA        /SET COUNTER OF WORDS
DLOOKC,   JSP GWORD       /GET WORD FROM TABLE
          DAC TEMB
          JSP SGET
          SAD TEMB
          JMP DLOOKD      /SO FAR SO GOOD
DLOOKE,   JSP UFWORD      /FREE VIRTUAL MEMORY-DIDN'T MAKE IT
          LAC WRDCNT      /GO GET NEXT RELATIVE COUNT
          SZF I 6
          IDA
          SUB TEMA
          JMP DLOOKB+2

&L
```

```
DLOOKD,   IDX TEMA        /COUNT WORD THAT WON
          SAS WRDCNT      /ARE WE DONE WITH SYMBOL YET
          JMP DLOOKC
          JSP UFWORD
          JSP GWORD       /WON-SO GET INFO
          DAC POINT+1
          SZF I 6         /DON'T GET ANOTHER IF WORD IF THERE ISN'T ONE
          JSP GWORD
          DAC POINT+2
          IDX DLOOKX      /GIVE RETURN 2
DLOOKG,   JSP UGWORD
          LAC (NOP)       /RESET SO SURE THAT GWORD IS RIGHT
          DAC GWORDD
          CLF 6           /RESET TO TWO WORDS OF INFO
DLOOKX,   JMP .

DLOOKI,   DAP DLOOKX      /SEPARATE ENTRY FOR VARIABLE LOOK-UP
          LAC (IDX GWORDP)
          DAC GWORDD
          LOAD VDRA
          JMP DLOOKF
&L
```

```
/IN CORE LOOK-UP ROUTINE
/WORD FOLLOWING IS ADDRESS OF TABLE
/NEXT RETURN MEANS THAT IT WAS NOT FOUND
/OTHER RETURN MEANS FOUND
LOOK,     DAP LOOKX
          LAC I LOOKX     /ADDRESS OF TABLE
LOOKA,    DAP LOOKP
LOOKP,    LAC .           /GET RELATIVE POINTER
          DAC TEME        /SAVE LEGAL BITS IN CASE COMMAND
          AND (7777)
          SZA I
          JMP LOOKB       /SYMBOL NOT FOUND
          SUB (2)         /CALCULATE WORD COUNT OF SYMBOL
          SAD WRDCNT      /IS IT SAME LENGTH
          JMP LOOKC
          ADD (2)         /NØ-GO GET NEXT SYMBOL
LOOKF,    ADD LOOKP
          JMP LOOKA

LOOKC,    LAW SYM-1       /SAME LENGTH : COMPARE TEXT
          DAP SGETP
          DZM TEMA        /COUNT OF WORDS LOOKED AT IN SYMBOL
LOOKD,    IDX LOOKP
          JSP SGET
          SAD I LOOKP
          JMP LOOKE       /STILL SAME
          JSP UFWORD      /LOST: GO TO THE NEXT SYMBOL
          LAW 1
          ADD WRDCNT
          SUB TEMA
          JMP LOOKF

LOOKE,    IDX TEMA        /ARE WE DONE
          SAS WRDCNT
          JMP LOOKD       /NOT DONE YET
LOOKG,    JSP UFWORD
          IDX LOOKP
          IDX LOOKX
          LIO I LOOKP
          DIO POINT+1
          DZM POINT+2
LOOKB,    IDX LOOKX
LOOKX,    JMP .

CHWRD,    Ø               /GIVEN CHARACTER COUNT: GET WORD COUNT
          DAP CHWRDX
          LAW 7777
          AND CHWRD
          SCR 9S
          SCR 8S
          DIV (3)
TEMB,      Ø
          SNI I
          IDA
          DAC WRDCNT
CHWRDX,   JMP .
&L
```

```
/FIND STEP GIVEN IN SNUM
/C(IO) IS DRA OF PROCEDURE
/R1←FOUND
/R2←NOT FOUND

SNUM,      Ø
FSTEP,     DAP FSTEPX
           LAC (JMP-1)
           JDA SGWORD      /USE GWORD TO GET WORDS
           JSP GWORD
           SUB (1)
           JMP FSTEPA+2

FSTEPA,    LAW I 2
           ADD TEMA
FSTEPA+2,  JDA NTHWD
           JSP RGWORD      /GET RELATIVE POINTER
           DIO SDRA
           DAC SDRA+1
           JSP GWORD       /GET RELATIVE POINTER
           DAC TEMA
           SZA I
           JMP FSTEPC-1    /OFF END
           JSP GWORD
           SUB SNUM
           SPA
           JMP FSTEPA      /NOT FOUND YET
           SZA             /FIND IT?
           IDX FSTEPX      /NO
FSTEPC,    JSP UGWORD
           LOAD SDRA
FSTEPX,    JMP .

/DISPATCH ON COMMAND: ENTER WITH TEXTP SET
DISPATCH,  JDA ACPUSH
           JSP CHKBRK      /HAS THERE BEEN A BREAK?
           JSP GNS
           SZA I
           JMP NILL        /CONSTANT
           AND (700000)    /IS IT A VERB
           SAS (700000)
           JMP DO          /NO: SO IMPLIED DO
           XOR I TEXTP
           DAC .+2
           SJMP .
COMRTN,    JSP CLINE
POP,NILL,  JDA PULL
            .+2
POPX,      SJMP .

ERRS,      SUB (1)         /CALCULATE ERROR CODES
           DAP .+2
           LAW 7777
ERRSA,     AND .
           SJMP ERROR
&L
```

```
COLON DO  LAW COMRTN      /IMPLIED DO: JUST EXECUTE EVAL
CALC,     JDA ACPUSH      /EVAL WITH FIRST SYMBOL GOTTEN
          JMP EVALD

EVAL,     JDA ACPUSH      /EVALUATE FOLLOWING STRING
          JSP GNS
EVALD,    LAC (200000)    /SET UP POSSIBLE REGISTERS
          DAC POINT
          STORE TEXTP,POINT+1
          LAC I TEXTP
          AND (700000)    /WHAT KIND OF SYMBOL
          SZA I
          CAL EVER1       /"THERE IS SOMETHING MISSING ON THIS LINE."
          SAD (200000)
          JMP EVALA       /CONSTANT
          SAD (300000)
          JMP EVALB       /VARIABLE NAME
          SAD (400000)
          JMP EVALC       /DEFINED PROCEDURE
          SAD (500000)
          JMP EVALO       /NO ARGUMENT BUILT-IN
          SAS (600000)
          CAL EVER2       /"X" NEEDS A MEANING
          XOR I TEXTP     /GET DISPATCH ADDRESS
          JDA ACPUSH      /AND SAVE
          JSP GNS         /IGNORE "OF" IF THERE
          SAD (500000 OF)
          JSP GNS
          JSP CALC        /EVALUATE FIRST ARGUMENT
          JSP VALUEQ      /MUST HAVE RETURNED A VALUE
EVALO,    LAW TSYM        /SET UP REGISTERS POSSIBLY NEEDED
          DAC FWORDP
          DZM FWORDP+4
          LIO POINT
          LAW POINT
          JMP POP         /AND DISPATCH TO CODE

EVALQ,    XOR I TEXTP     /DISPATCH ON ZERO ARG PRCS
          JDA ACPUSH
          JMP EVALO

2ARG,     JDA ACPUSH      /ROUTINE TO GET SECOND ARGUMENT FOR MACHINE CODE
          JSP PPUSH       /SAVE FIRST ARGUMENT
          JSP GNS         /IGNORE "AND" IF THERE
          SAD (500000 CAND)
          JSP GNS
          JSP CALC        /GET SECOND ARGUMENT
          JSP VALUEQ      /MUST HAVE RETURNED A VALUE
          JDA PULL
           TPOINT+2
           TPOINT+1
           TPOINT
          JMP EVALO

&L
```

```
EVALA,    LIO I TEXTP     /SEE IF WORD OR SENTENCE
          RIL 3S
          LAW 7777        /ARE WE LESS THAN SIX CHARACTERS
          AND I TEXTP
          SUB (7)
          SPQ
          JMP EVALP       /SIX CHARACTER MODE
          LAC (600000)    /SENTENCE?
          SPI
          DAC POINT
          LAW EVALO
EVALG,    DAP EVALGX
          LAC I TEXTP
          JDA CHWRD
          JDA GNST
EVALGX,   JMP .

EVALP,    DAC TEMA        /SIX CHARACTER MODE CONSTANT: SAVE COUNT
          LAC (700000)    /SENTENCE?
          SPI I
          LAC (300000)
          DAC POINT
          JSP TWORD       /GET FIRST WORD OF CONSTANT
          DAC POINT+1
          LAW 3           /IS THERE ONLY ONE WORD IN CONSTANT
          ADD TEMA
          SPA
          JMP EVALO       /YES: SO DONE
          JSP TWORD       /NO: GET OTHER WORD
          DAC POINT+2
          CLA             /IS THERE ALT. MODE TO SKIP ALSO
          SAD TEMA
          JSP TWORD
          JMP EVALO

EVALB,    JSP EVALG       /MOVE ON TOO NEXT SYMBOL
COLON THING               JSP VGET    /SET UP VARIABLE IN STANDARD FORM
          JSP LOOK        /BUILT-IN NAME?
           SVTBL
          JMP EVALH       /NO, CHECK OTHER NAMES
          JDA PUSH
           POINT+1
          JMP EVALO       /DISPATCH TO CODE FOR SYSTEM SYMBOLS

EVALH,    JSP DLOOKI      /LOOK UP IN NAMES TABLE
           JMP CEMPTY     /NOT THERE, TREAT AS EMPTY
          JMP EVALO       /FOUND IT SO ALL SET
&L
```

```
EVALC,    JSP FPROD       /LOOK UP PROCEDURE
          DAC TEMA        /DRA OF PROCEDURE LOCATION
          SZA I
          CAL EVER3       /"X" NEEDS A MEANING
          SAD DDRA
          CAL EVER4       /"X" HAS NOT BEEN COMPLETELY DEFINED
          JDA PUSH
           TRCR           /TRACE FLAG
          JSP GWORD
          DAC TEMB        /NUMBER OF ARGUMENTS
          DZM TEMC        /NUMBER OF ARGUMENTS PROCESSED SO FAR
          JSP UGWORD      /FINISHED WITH DIRECTORY
          CLA
          SAD TEMB        /ANY ARGUMENTS
          JMP EVALJ       /NO: SO SET
          JSP GNS         /YES: IS THERE AN OF
          SAD (500000 OF)
          JSP GNS         /IF "OF" IGNORE IT
EVALK,    JDA PUSH
           TEMA
           TEMB
           TEMC
           TEMD
           TEMD+1
           TEME
          JSP CALC        /EVALUATE NEXT ARGUMENT
          JDA PULL
           TEME
           TEMD+1
           TEMD
           TEMC
           TEMB
           TEMA
          IDX TEMC        /HAVE WE PROCESSED ALL ARGUMENTS
          SAD TEMB
          JMP EVALJ       /YES
          JSP GNS         /NO: SEE IF "AND" BETWENN
          SAD (500000 CAND)
          JSP GNS
          LAW EVALK
PPUSH,    JDA PUSH        /SAVE POINT ON PUSH DOWN LIST
           POINT
           POINT+1
           POINT+2
          JMP I PUSH
&L
```

JSP VALUEQ

```
EVALJ,    DZM TRCR
          LAC TEME
          SPA
          IDX TRCR        /THIS PROCEDURE TRACED
          DZM TEMC        /DEFINE BOUND VARIABLES (BACKWARDS)
          JSP UTWORD      /RELEASE PRESENT STEP
          STORE TEXTP,TPOINT+1
          LODE TEMA       /GET NEW PROCEDURE IN CORE
          JDA STWORD
          LOAD VDRA       /GET VARIABLE DIRECTORY IN CORE
          JDA SGWORD
          LAC (600000)    /SET UP POINTER TO STOP
          IOR GWORDP+5
          DAC TEMF
          JSP TWORD       /NEGLECT RELATIVE COUNT
          CLA
          SAD TEMB        /ANY BOUND VARIABLES
          JMP EVALL       /NO: GO ON AND EXECUTE
EVALM,    JSP TWORD
          DAC WRDCNT      /-NO. OF WORDS IN SUMBOL
          DAC TEME
          LAC POINT+2     /SET SECOND WORD OF VALUE
          JDA VSTORE
          LAC POINT+1
          JDA VSTORE
          JSP TWORD       /PUT NAME IN BACKWARDS
          JDA VSTORE
          ISP WRDCNT
          JMP .-3
          LAW 3
          SUB TEME
          IOR POINT       /BOTTOM 15 BITS 0?
          JDA VSTORE      /SAVE RELATIVE POINTER
          IDX TEMC        /HAVE WE HANDLED ALL ARGUMENTS
          SAD TEMB
          JMP EVALL       /YES
          JDA PULL        /NO: GET NEXT ONE
           POINT+2
           POINT+1
           POINT
          JMP EVALM       /GO PROCESS IT
&L
```

```
EVALL,    JDA PUSH        /SAVE QUANTITIES FROM THIS ROUTINE
           XDRA
           RNUM
           VDRA
           VDRA+1
           TRUTH
           PROD
           PROD+1
           TPOINT+1
           TPOINT+2
          LAC TEMA
          DAC XDRA
          LOAD TEMD       /SAVE ADDRESS OF NAME
          DIO PROD
          DAC PROD+1
          DZM TRUTH       /RESET TRUTH VALUE
          CLA
          SAD TRCR        /ARE WE TRACED
          JMP .+3         /NO
          SJMP TRACP      /YES, PRINT ARGS
          JSP UGWORD      /RELEASE VARIABLE DIRECTORY
          LOAD TEMD       /RESET PROD
          UNLOAD PROD
          JSP RGWORD      /ARE RESET START OF VARIABLE DIRECTORY
EVALF,    DIO VDRA
          DAC VDRA+1
          JSP TWORD       /REL PTR FOR TO LINE
          SUB (1)
          JDA GNST        /SKIP OVER TO LINE
EVALN,    JSP TWORD       /SKIP RELATIVE POINTER FOR STEP
          SZA I
          JMP STPA        /DONE WITH PROCEDURE: NO "RETURN"
          SUB (1)
          DAC NPTR        /SET WORDS LEFT IN LINE COUNT
          JSP TWORD       /GET STEP NUMBER
          DAC RNUM
          JSP DISPATCH    /GO EXECUTE COMMAND
          JMP EVALN
```

STP B

```
VSTORE,   Ø               /STORE WORD AT BEGINNING OF VARIABLE DIRECTORY
          DAP VSTORX
          LAC GWORDP      /ARE WE AT BEGINNING OF BLOCK
          SAS TEMF
          JMP VSTORA      /NO: GO STORE WORD
          JSP UGWORD      /YES: GO TO CHANGE BLOCKS
          LIO I GWORDP+5  /IS THERE A PREVIOUS BLOCK
          SNI I
          JMP VSTORB      /YES: GO READ IT
          LAW I 3         /NO: MUST ADD A BLOCK
          ADD GWORDP+5
          JDA WNPCNT      /WNIP AND COUNT (FOR QTRK ERROR)
          DIO I GWORDP+5  /SET POINTER IN OLD BLOCK
          JSP USEDG
          JSP GETIT       /GET THE NEW ITEM IN CORE
          DZM I BBPTR2
          LAC I GWORDP+1
          DAC I BBPTR1
          JSP USEDG
```

```
VSTORB,   LAC (JMP BLNG-4)          /GET PREVIOUS BLOCK
          JDA SGWORD
          LAC (600000)
          IOR GWORDP+5
          DAC TEMF
VSTORA,   LAC VSTORE
          DAC I GWORDP
          LAW I 1
          ADD GWORDP
          DAC GWORDP
          JSP USEDG
VSTORX,   JMP .

FPROD,    DAP FPRODX      /FIND PROCEDURE
          JSP TWORD       /GET DRA OF NAME
          DAC TEMD
          JSP TWORD
          DAC TEMD+1
          LIO TEMD
          JDA SGWORD
          JSP GWORD       /GET RELATIVE POINTER
          DAC TEME        /BIT 0←TRACED,1←HIDDEN
          SUB (2)
          JDA NTHWD
FPRODX,   JMP .

CEMPTY,   LAW EMPTY       /HANDLE VARIABLE NAME AS EMPTY
          JDA SET
          JMP EVALO
&L
```

```
COLON RETURN            JSP ACPULL  /RETURN-REMOVE DISPATCH RETURN
          JSP EVAL
          JSP VALUEQ    /MAKE SURE SOMETHING RETURNED
          CLF 1         /FLG 1=STOP (USED IN TRACE PRINTOUT)
RETB,     JSP CLINE
          JSP UTWORD    /RELEASE PRESENT CODE
          LOAD PROD     /SAVE NAME OF RETURNING PROCEDURE
          UNLOAD EXPROD /IN CASE OF ERROR
          CLA
          SAD TRCR      /TRACED?
          JMP .+3       /NO
          SJMP TRACR    /YES PRINT "FOO RETURNS .."
RETC,     JSP ACPULL
           TEMD
           PROD+1
           PROD
           TRUTH
           VDRA+1
           VDRA
           PNUM
           XDRA
           TRCR
          LIO TEMD
          JDA STWORD    /RESET CURRENT PROCEDURE
          JMP POP

COLON STP JSP ACPULL    /STOP, RETURN /EMPTY/
          JSP CLINE
STPA,     CLC
          DAC POINT     /MARK AS NOTHING RETURNED
          STF 1
STPC,     JMP RETB+1

COLON TEST              JSP EVAL    /SET IF FLAG
          JSP VALUEQ
          JSP VGET
          JSP LOOK
           TFTABLE
          CAL TSTERR    /NOT TRUE OR FALSE
          DIO TRUTH
          JMP COMRTN

TFTABLE,  SYMBOL TRUE,0
          SYMBOL FALSE,-0
          0

VALUEQ,   DAP VALUEX    /MAKE SURE SOMETHING WAS RETURNED
          CLC
          SAD POINT
          CAL EVER6     /NOTHING THERE
VALUEX,   JMP .
&L
```

```
COLON IF  JSP GNS          /"TRUE" OR "FALSE"
          LIO IF           /NOT Ø OR -Ø
          SAD (5ØØØØØ ATRUE)
          CLI
          SAD (5ØØØØØ AFALSE)
          CLI"U"CMI
          LAI
          SAD IF
          CAL IFERR        /NOT "YES" OR "NO"
          SAD TRUTH
          JMP DISPATCH+1 /OK: EXECUTE WHAT FOLLOWS
IFA,      LAC NPTR
          JDA GNST
          JMP POP

CLINE,    DAP CLINEX       /IS THERE ANYTHING ELSE ON LINE
          JSP GNS
          SZA
          CAL DISERR       /YES: ERROR
CLINEX,   JMP .

TSET,     Ø                /SET UP TEXT HANDLING ROUTINE
          DAP TSETX        /C(AC)=ADDRESS OF 3 WORD BLOCK
          LIO I TSET       /WHAT TYPE OF VARIABLE STORAGE
          IDX TSET
          RIL 2S
          SPI
          JMP TSETC        /6 CHARACTERS OR LESS
          LIO I TSET       /1Ø: CONSTANT IN STEP
          IDX TSET
          LAC I TSET
          JDA SGWORD
          LAW GWORDA
TSETB,    DAP TGETF
TSETX,    JMP .

TSETC,    DZM GWORDP+4     /11: 6 CHARACTERS OR LESS
          DZM GWORDP+5
          DAC GWORDP
          ADD (JMP 1)
          DAC GWORDP+3
          LAW (4ØØØØØ)
          DAC GWORDP+1     /MARK AS NOT FROM DRUM IF 6 CHAR MODE
          LAW TGETB
          JMP TSETB
&L
```

```
TGET,     DAP TGETX       /GET A CHARACTER FROM MEMORY
          LAC GWORDP
          SAD GWORDP+3    /AT END OF BLOCK?
TGETF,    JSP .           /YES. GWORDA IF STRING; TGETB IF 6 CHARACTER
          LCH I GWORDP
TGETX,    JMP .

TGETB,    LAC (CHARACTER L#)          /SIX CHARACTER MODE IMPLIED #
          JMP TGETX
&L
```

```
TSTORE,   Ø                 /GENERAL STORE CHARACTER IN VARIABLE MEM,
          DAP TSTORX
          LAC FWORDP        /ARE WE DONE WITH 6 CHAR MODE
          SAS (JMP TSYM+1)
          JMP TSTORB
          LIO TFREE
          DIO TPLACE
          LAC TFREE+1
          DAC TPLACE+1
          JDA SFWORD
          LAC TSYM          /AND PUT THE TWO WORDS IN IT
          JDA FWORDS
          LAC TSYM+1
          JDA FWORDS
TSTORB,   LAC FWORDP        /ARE WE OFF END OF 6 WORD BLOCK
          SAD FWORDP+3
          JSP FWORDA
          LAC TSTORE
          DCH I FWORDP
TSTORX,   JMP .

VGET,     DAP VGETX         /SET UP VARIABLE NAME IN STANDARD FORM
          LAW POINT
          JDA TSET
          JSP FSET
          JSP TGET
          SAD (CHARACTER L#)          /AN EMPTY NAME?
          CAL NMERR5        /YES
          JMP .+2
          JSP TGET
          JDA FSTORE
          SAS (CHARACTER R#)
          JMP .-3
          JSP UFWORD
          JSP UGWORD
VGETX,    JMP .
&L
```

```
TDONE,    DAP TDONX       /FINISHED: TELL WHAT TYPE
          LAC FWORDP      /CHECK IF IMPLIED ALT. MODE
          SAD (JMP TSYM+1)
          JMP TDONC
          LAC (CHARACTER L#)          /OTHERWISE STORE AN ALT. MODE
          JDA TSTORE
TDONC,    CLA
          SAS FWORDP+4    /6 CHARACTER MODE?
          JMP TDONF       /NO: STRUNG BLOCKS
TDOND,    LAC TSYM        /YES
          DAC POINT+1
          LAC (300000)
          LIO TSYM+1
TDONB,    DIO POINT+2
          XOR POINT
          AND (377777)
          XOR POINT
          DAC POINT
TDONX,    JMP .

TDONF,    LIO I FWORDP+1
          DIO TFREE
          LAC FWORDP
          IOR (JMP)
          SUB FWORDP+5
          DAC TFREE+1
          JSP UFWORD
          CLA
          LIO TPLACE      /COPY DRA AND REL. PTR. INTO POINT
          DIO POINT+1
          LIO TPLACE+1
          JMP TDONB
&L
```

```
SET,      Ø                /SUBROUTINE TO GIVE SET ANSWERS
          DAP SETX
          LAC I SET
          DAC POINT
          IDX SET
          LAC I SET
          DAC POINT+1
          IDX SET
          LAC I SET
          DAC POINT+2
SETX,     JMP .

TRUE,     3ØØØØØ           TEXT .TRUE# .
FALSE,    3ØØØØØ           TEXT .FALSE#.
EMPTY,    3ØØØØØ           TEXT .#     .

EMPTYQ,   Ø                /IS VALUE EMPTY: R1 MEANS YES, R2 NO
          DAP EMPTYX       /TRANSPARENT TO IO
          LAC (3ØØØØØ)
          AND I EMPTYQ
          SAS (3ØØØØØ)
          JMP EMPTYA
          IDX EMPTYQ
          LCH I EMPTYQ
          SAS (CHARACTER L#)
EMPTYA,   IDX EMPTYX
EMPTYX,   JMP .
&L
```

```
FSTORE,   0               /STORE CHARACTERS IN VIRTUAL MEMORY
          DAP FSTORX
          IDX CHCNT
          LAC FWORDP      /ARE WE AT END OF BUFFER
          SAS FWORDP+3
          JMP FSTORA
          SAS (JMP SYM+4)           /IS THIS THE FIRST TIME
          JMP FSTORB      /NO
          LODE NDRA
          DIO POINT+1
          DAC POINT+2
          JDA SFWORD
FSTORA,   LAC FWORDP
          AND (600000)
          SAS (600000)
          JMP FSTORC      /NOT AT BEGINNING OF WORD
          LAW 1
          ADD FWORDP
          DAP .+1
          DZM .
          IDX WRDCNT
FSTORC,   CLA
          SAS FWORDP+4
          JSP USEDF
          LAC FSTORE
          DCH I FWORDP
FSTORX,   JMP .

FSTORB,   JSP FWORDA
          JMP FSTORA

FSET,     DAP FSETX
          LAC (JMP SYM-1)
          DAC FWORDP
          ADD (5)         /INITIALIZE FSTORE
          DAC FWORDP+3
          DZM FWORDP+4
          DZM CHCNT
          DZM WRDCNT
FSETX,    JMP .
```

```
SVTBL,      SYMBOL EMPTY,CEMPTY
            SYMBOL CONTENTS,ACNTNT
            SYMBOL LINE FEED,LFSCR
            SYMBOL CARRIAGE RETURN,ACRSLF
            SYMBOL FILES,LFILE
            SYMBOL FORM FEED,FORMF
            SYMBOL BLANK,BLANK
            SYMBOL BELL,BELL
            SYMBOL QUOTE,DQ
            SYMBOL SKIP,CR
            SYMBOL TTNO,TTLINE          /SCANNER LINE CONNECTED TO
            SYMBOL USER,USERA
            0

COLON FINIT                 SJMP FINITA
COLON DATESV                SJMP DATSAV
COLON DATEGT                SJMP DATGET
COLON SIZE                  SJMP SIZEA
COLON OWNER                 SJMP OWNERA
COLON PRNT                  LAC (PRNT1)
          JMP .+2
COLON TYPE                  LAC (TYPE1)
          SJMP EVAL
COLON SUPDO                 LAC (SUPDOF)
          JMP TYPE+1
COLON DDTA                  CAL 7777
COLON REQUEST               SJMP REQUE1
COLON ASK SJMP AASK
&L
```

```
ITMCNT,    10              /NUMBER OF ITEMS WRITTEN
SYM,       REPEAT 5,0      /STORE FIRST 15 CHARACTERS OF SYMBOL HERE
WRDCNT,    0               /COUNT OF WORDS IN SYMBOL
RELPTR,    0               /RELATIVE POINTER FOR STEP
NPTR,      0               /COUNT LEFT IN LINE (SOMETIMES)
CHCNT,     0               /CHARACTER COUNT IN SYMBOL
CHARNO,    0               /TT CURSOR POSITION ON PAGE
TRUTH,     0               /IS FLAG FOR TRUTH AND FALSEHOOD
TCHAR,     0               /ACTUAL CHARACTER TO TERMINATE LAST COMMAND
TDRA,      0               /DRUM ADDRESS FOR VARIABLE LINKING
TFREE,     REPEAT 2,0      /FREE LIST POINTER
TSYM,      REPEAT 2,0      /BUFFER FOR SIX CHARACTER MODE
TPLACE,    REPEAT 2,0      /DRA AND RELATIVE OF TEXT BEING STORED
GWORDP,    REPEAT 6,0      /SET OF POINTERS FOR SEARCH
FWORDP,    REPEAT 6,0      /SECOND SET OF POINTERS FOR SEARCH
TEXTP,     REPEAT 6,0      /POINTERS FOR COMMAND SCAN
DNAME,     REPEAT 2,0      /NAME OF PROCEDURE
DDRA,      0               /DRA OF PROCEDURE BEING DEFINED
IDRA,      0               /DRA OF INPUT STRING
GCDRA,     0               /DRA OF BEGINNING OF CDRA
CDRA,      REPEAT 2,0      /DRA OF DIGESTER CODE AND DIRECT COMMANDS
DCDRA,     REPEAT 2,0      /CDRA FOR DO COMMAND
PDRA,      0               /DRA OF PROCEDURE DIRECTORY
VDRA,      REPEAT 2,0      /DRA OF VARIABLE DIRECTORY
GVDRA,     0               /GLOBAL VARIABLES DRUM ADDRESS
ADRA,      0               /DRA OF ABBREVIATION DIRECTORY
AVALUE,    REPEAT 2,0      /DRA OF VALUE OF ABBREVIATIONS
NDRA,      0               /DRA FOR LONG NAMES
SDRA,      REPEAT 2,0      /DRA OF LAST SYMBOL FOUND IN SEARCH
PROD,      REPEAT 2,0      /PTR. TO NAME OF PROD BEING EXECUTED
EXPROD,    REPEAT 2,0      /NAME OF PROCEDURE JUST RETURNED
XDRA,      0               /DRA OF RUNNING PROCEDURE
RNUM,      0               /STEP NUMBER BEING EXECUTED
SECOND,    0               /CLOCK REGISTER
PDLDRA,    0               /DRA OF PUSH DOWN LIST-THIS ITEM
OPDLDRA,   0               /DRA OF BEGINNING OF PDL
PLCOUNT,   0               /POINTER TO CHANGE WORD FOR THIS ITEM OF PDL
PDLCNT,    0               /PDL COUNT OF ENTRIES
PRCNT,     0               /PROCEED PDL DEPTH
PUSHT,     DAC .           /POINTER TO TOP OF PDL BUFFER
PUSHI,     0               /TEMP STORAGE FOR IO IN PUSH AND PULL
POINT,     REPEAT 3,0      /MAIN TEXT POINTER FOR STRINGS
           740000          /ALT. MODE SOMETIMES USEFUL
TPOINT,    REPEAT 3,0      /SECONDARY TEXT POINTER FOR STINGS
&L
```

```
BTBL,       REPEAT BCNT,0   /TABLE OF DRA'S FOR BUFFERS
BCHK,       REPEAT BCNT,0   /TABLE OF CHANGE REGISTERS FOR BUFFERS
BBPTR,      0               /ADDRESS OF CURRENT BUFFER
BBPTR1,     0               /ADDRESS OF CURRENT BUFFER+1
BBPTR2,     0               /ADDRESS OF CURRENT BUFFER+2
DRUMT,      0               /TEMP. STORAGE FOR DRUM ROUTINES
DRUMI,      0               /TEMP STORAGE FOR DRA IN FITM,FBUF
BCOUNT,     0               /ADDRESS OF CHANGE REGISTER-THIS BUFFER
BTIME,      0               /NUMBER OF ITEMS REFERENCED SO FAR
FNUM,       0
VERB,       0
SSBASE,     REPEAT 6,0
TEMD,       REPEAT 2,0      /TEMP STORAGE FOR DRA'S
TEMF,       0               /STORAGE USED BY EVAL FOR VSTORE
FLPTR,      REPEAT 2,0      /PTRS FOR LOADING FROM FILES
TRCR,       0               /0 IF PROD NOT TRACED
NTRCS,      0               /DEPTH OF TRACES
FLSFLG,     0               /FLG FOR SKIPPING PRCS IN GET
DDTSEG,     0               /DRUM ADDR OF DDT IF THERE IS ONE
PERMIN,     0               /PERMANENT DRA OF INPUT SEG
RANDA,      0               /FOR RANDOM NUMBERS
            FLEXO PMW
HOARDB,     0               /HOARDING BIT
BRKCNT,     1               /ZERO IF A BREAK HEARD
USER,       0               /USER'S ID FOR FILES (BIT 0←WHEEL)
INITFL,     0               /DRA OF INITIALIZING FILE
FILDRA,     0               /DRA OF FILE DIRECTORY
OFREE,      0               /DRA OF BEGINING OF VARIABLE STORAGE
&L
```

TTPTR, 10:0 /TT POINTERS

```
/SEGMENT ROUTINES
/ADDRESS SEGMENT BY EXTENDED ADDRESS: SEG NUM IN TOP 6 BITS
/PHYSICAL CORE ADDRESS IN BOTTOM 12 BITS
/SEG NUMBER Ø MEANS DO NOT READ IN SEG

.SJSP,     DAP SJMPX       /JSP SIMULATOR
           LAW I STBL-1
           ADD SEGPTR
           AND (37)
           RAR 6S
           XOR SJMPX
           XOR (600000)
           IDA
           DAC .SJMP
           JMP SJMPA

.SJMP,     Ø               /JMP SIMULATOR
           DAP SJMPX
SJMPA,     LAC I SJMPX
           DAP SJMPX
           RAL 6S
           AND (37)
           SZA I           /IF Ø SEGMENT MEANS DON'T READ IN
           JMP SJMPB
           ADD (STBL-1)
           DAP SEGPTR
SEGPTR,    LAC .
           SAD SEGDRA
           JMP SJMPB
           DIO DRUMI
           LIA
           LAW SEG
           RAIS 1
TEMC,       Ø
           DIO SEGDRA
           LIO DRUMI
SJMPB,     LAC .SJMP
SJMPX,     JMP .

SEGDRA,    .+1/
SEG,       CON-.
           Ø
REPEAT 1IF P,PRINT !SEG!
SEGNO=1
&L
```

```
COLON FIRST             JDA EMPTYQ  /FIRST OF
          JMP EVALO
          LAW POINT
          JDA TSET
          LIO POINT
          SPI
          JMP FIRSTA    /SENTENCE
          JSP TGET
          JDA TSTORE
          SAS (77)
          JMP .+3
          JSP TGET
FIRSTE,   JDA TSTORE    /WORD: STORE FIRST CHARACTER (ALSO LAST STORE)
FIRSTB,   DZM POINT     /FIRST IS ALWAYS A WORD
          JSP UGWORD
FIRSTC,   JSP TDONE
          JMP EVALO

FIRSTA,   JSP TGET      /SENTENCE
          SAS (CHARACTER L#)
          SZA I
          JMP FIRSTB
          JDA TSTORE
          JMP FIRSTA

COLON BUTF              JDA EMPTYQ  /BUT FIRST
          JMP EVALO
          LAW POINT
          JDA TSET
          LIO POINT
          SPI
          JMP BUTFA
          JSP TGET      /WORD: PASS FIRST CHARACTER
          SAD (770000)
          JSP TGET
BUTFB,    JSP TFIN      /DO RIGHT THINGS FOR POINTERS
          JSP UGWORD
          JMP EVALO

BUTFA,    JSP TGET
          SZA I
          JMP BUTFB
          SAS (CHARACTER L#)
          JMP BUTFA
          JSP UGWORD
          JMP CEMPTY
&L
```

```
TFIN,     DAP TDONX
          LAC (300000)
          AND POINT
          SAD (300000)
          JMP TFINA
          JSP RGWORD
          DIO POINT+1
          DAC POINT+2
          LAW TSYM
          DAC FWORDP
TCHKC,    JSP TGET
          LAC (JMP TSYM+1)
          SAD FWORDP
          JMP TCHKB       /DON'T THINK MADE IT
          LCH GWORDP
          DCH I FWORDP
          SAS (CHARACTER R#)
          JMP TCHKC
TCHKD,    JSP UGWORD
          JMP TDOND       /6 CHARACTER MODE: CHANGE POINTER

TCHKB,    LCH GWORDP      /FAILED: IS IT ALT MODE
          SAD (CHARACTER L#)
          JMP TCHKD
TCHKE,    JSP UGWORD
          JMP TDONX

          JDA TSTORE
TFINA,    JSP TGET
          SAS (CHARACTER L#)
          JMP TFINA-1
          JSP UGWORD
          JMP TDONE+1
```

```
WORDS,    0              /SUBROUTINE FOR SENTENC  AND WORD
          DAP WORDSX
          LAC WORDS
          JDA TSET
WORDSA,   JSP TGET
          SAD (CHARACTER L#)
          JMP WORDSB
          JDA TSTORE
          JMP WORDSA

WORDSB,   JSP UGWORD
WORDSX,   JMP .

WORDC,    0              /ANOTHER SUCH SUBROUTINE
          DAP WORDCX
          LAW TPOINT
          JDA EMPTYQ
          JMP WORDCA
          LAW POINT
          JDA EMPTYQ
          JMP WORDCB
WORDCX,   JMP .

WORDCB,   LAW TPOINT
          JDA SET
WORDCA,   LAW POINT
          JDA EMPTYQ
          JMP EVALO
          LAC WORDC
          IOR POINT
          DAC POINT
          JMP EVALO

COLON WRD SJSP 2ARG      /WORD: GET BOTH ARGUMENTS
          LAC TPOINT
          SMA
          SPI
          CAL WRDERR     /BOTH ARGUMENTS MUST BE WORDS
          CLA
          JDA WORDC
          LAW TPOINT
          JDA WORDS
          LAW POINT
          JDA WORDS
          JMP FIRSTC
&L
```

```
COLON SENT              SJSP 2ARG
          LAC (400000)
          JDA WORDC
          LAW TPOINT
          JDA WORDS
          CLA
          JDA TSTORE
          LAW POINT
          JDA WORDS
SENTA,    LAC (400000)
          DAC POINT
          JMP FIRSTC
COLON WORDQ             CMI          /WORDQ
COLON SENTQ             JDA EMPTYQ   /SENTENCEQ
          CLI"U"CMI     /IF EMPTY AUTOMATICALLY TRUE
          LAW TRUE
          SPI I
SENTQA,   LAW FALSE
          JMP CEMPTY+1

COLON EMPTQ
          LAW POINT
          JDA EMPTYQ
          JMP EMPTQA    /TRUE
          JMP SENTQA    /FALSE

EMPTQA,   LAW TRUE
          JMP CEMPTY+1

COLON ZEROQ
          LAW POINT
          JDA TSET
          JSP TGET
          SAS (CHARACTER L+)
          SAD (CHARACTER L-)
          JSP TGET      /SKIP THE SIGN
          SAS (CHARACTER L0)
          JMP ZEROQA    /FALSE
          JSP TGET
          SAD (CHARACTER L0)
          JMP .-2       /SKIP ANY NUMBER OF ZEROS
          SAS (CHARACTER L#)
          JMP ZEROQA
          JSP UGWORD
          JMP EMPTQA

ZEROQA,   RAL 6S
          XOR (20)
          SUB (10.)
          SMA
          CAL ZERERR    /NOT A NUMBER
          JSP TGET
          SAS (CHARACTER L#)
          JMP ZEROQA
          JMP SENTQA    /RETURN FALSE
&L
```

```
COLON NUMQ              SPI I       /NUMBERQ
          JDA EMPTYQ
          JMP SENTQA    /MUST BE NON-EMPTY WORD
          LAW POINT
          JDA TSET
          JSP TGET      /CHECK FOR SIGN
          SAS (CHARACTER L-)
          SAD (CHARACTER L+)
          JMP NUMQC     /SIGN EXISTS
          JMP NUMQA+1

NUMQA,    JSP TGET
          SAD (CHARACTER L#)
          JMP NUMQB     /MUST BE NUMBER SINCE HASN'T FAILED
NUMQA+3,  RAL 6S
          XOR (20)
          SUB (12)
          SPA
          JMP NUMQA
NUMQD,    JSP UGWORD    /NOT A NUMBER
          JMP SENTQA

NUMQC,    JSP TGET      /GOT SIGN: MAKE SURE FOLLOWED BY NUMBER
          SAD (CHARACTER L#)
          JMP NUMQD     /NOT A NUMBER
          JMP NUMQA+3

NUMQB,    JSP UGWORD
          LAW TRUE
          JMP CEMPTY+1

NWORD,    DAP NWORDX    /GET NEXT (WORD OR CHARACTER)
          IDX TEME
          JSP TGET
          SAD (770000)
          JSP TGET
NWORDA,   LIO POINT
          SPI I
          JMP NWORDX    /WORD: SO RETURN EACH CHARACTER
          SAS (CHARACTER L#)
          SZA I
NWORDX,   JMP .
          IDX TEME
          JSP TGET
          JMP NWORDA
&L
```

```
WAITA,     CLA
           DELAY            /WAIT FOR END OF OUTPUT
COLON WAIT                                        LAC TTPTR
           CLA
           PEEK
           AND (40)         /TYPEING OUT NOW
           SZA
           JMP WAITA        /YES
           SJSP EVAL
           CLF 7
           LAC POINT
           AND (700000)
           SAS (300000)     /6 CHARACTER NUMBER?
           CAL WAITER       /TOO LARGE
           LAW POINT+1
           DAC FSA
           DNM
           CAL WAITER
           SPO
           CAL WAITER
           DAC TEMB
           SUB (60."T"60."T"24.)
           SMA
           CAL WAITER       /TOO LONG A WAIT
           LCH I FSA        /NUMBER IS OK
           SAS (CHARACTER L#)
           CAL WAITER       /BUT THERE'S SOMETHING ELSE
           JSP SEC          /CURRENT TIME
           DAC TEMC
           LAC TEMB
WAITB,     DELAY
           JSP CHKBRK
           JSP SEC
           SUB TEMC
           SPA              /INTO NEXT DAY?
           ADD (60."T"60."T"24.)       /YES
           SUB TEMB
           CMA
           SPQ
           JMP COMRTN
           JMP WAITB        /WAIT SOME MORE
COLON COUNT                 JDA TSET    /COUNT
           DZM TEMB
COUNTB,    JSP NWORD
           SAD (CHARACTER L#)
           JMP COUNTA
           IDX TEMB
           JMP COUNTB
```

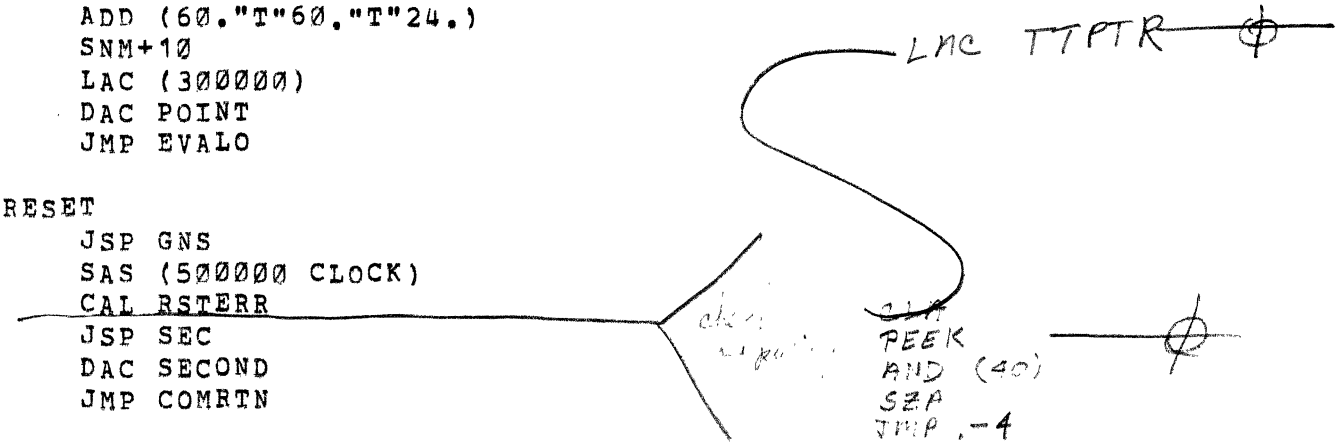
```
COUNTA,    SPI
           IDX TEMB
           LAC (300000)    /WILL FIT IN 6 CHARACTER MODE
           DAC POINT
           JSP UGWORD
           LAW POINT+1
           DAC STS
           LAC TEMB        /CHECK TO SEE THAT NUMBER NOT TOO LONG
           SUB C10E5       /100000.
           SMA
           CAL CNTERR
           LAC TEMB
           SNM"U"10
           JMP EVALO

C10E5,     100000.

COLON LAST                 JDA EMPTYQ
           JMP EVALO
           LAW POINT
           JDA TSET
           LIO POINT
           SPI
           JMP LASTB
LASTC,     JSP TGET        /A WORD SO FIND LAST CHARACTER
           SAD (CHARACTER L#)
           JMP LASTD
           SAS (770000)    /IS WARNING?
           JMP LASTE       /NO JUST SAVE CHARACTER
           JSP TGET        /YES, GET SECOND HALF
           IOR (77)        /MARK WARNING
LASTE,     DAC TEME        /SAVE IT
           JMP LASTC

LASTD,     LAC TEME        /STORE LAST CHARACTER
           RAR 6S
           SPA             /WARNING?
           JMP .+3         /YES
           RAL 6S          /NO UNROTATE
           JMP FIRSTE
           JDA TSTORE      /YES, STORE BOTH HALVES
           JMP FIRSTE
&L
```

```
LASTB,    JSP RGWORD
          DIO TPOINT+1
          DAC TPOINT+2
          JSP NWORD
          SAS (CHARACTER L#)
          JMP LASTB
          LAC (377777)
          AND POINT
          DAC POINT       /MUST BE A WORD
          SAD (300000)
          JMP LASTA       /6 CHARACTER MODE
          DAC TPOINT
          LAW TPOINT
          JDA TSET
          JMP BUTFB

LASTA,    LAC TPOINT+2
          DAC GWORDP
          JMP BUTFB

COLON BUTL                JDA EMPTYQ
          JMP EVALO
          LAW POINT
          JDA TSET
          DZM TEME

BUTLA,    LAC TEME
          DAC TEMA
          JSP NWORD
          SAS (CHARACTER L#)
          JMP BUTLA
          JSP UGWORD
          LAW I 1
          ADD TEMA
          SPQ
          JMP CEMPTY
          CMA
          DAC TEME
          LAW POINT
          JDA TSET
BUTLB,    JSP TGET
          JDA TSTORE
          SAD (77)
          JMP BUTLB
          ISP TEME
          JMP BUTLB
          JMP FIRSTB+1
&L
```

```
CSET,     Ø                 /SECOND SET OF TEXT HANDLING ROUTINES
          DAP CSETX         /JUST LIKE TSET AND TGET
          LIO I CSET
          IDX CSET
          RIL 2S
          SPI
          JMP CSETC
          LIO I CSET
          IDX CSET
          LAC I CSET
          JDA SFWORD
          LAW FWORDA
CSETB,    DAP CGETF
CSETX,    JMP .

CSETC,    DZM FWORDP+4
          DAC FWORDP
          ADD (JMP 1)
          DAC FWORDP+3
          LAW CGETB
          JMP CSETB

CGET,     DAP CGETX         /GET CHARACTER FROM TEXT
          LAC FWORDP
CGETA,    SAD FWORDP+3
CGETF,    JSP .
CGETD,    LCH I FWORDP
CGETX,    JMP .

CGETB,    LAC (CHARACTER L#)
          JMP CGETX

&L
```

```
COLON DQ  LAC DQUOTE
NFILLA,   DAC NFILL
          LAW NFILL-1
          JMP CEMPTY+1
COLON USERA               LAW NFILL
          DAC STS
          LAW 7777
          AND USER
          SNM+10
          JMP NFILLA+1
COLON BELL                LAC DBELL
          JMP NFILLA

COLON BLANK               LAC DBLANK
          JMP NFILLA

COLON CR  LAC CRLF
          JMP NFILLA

COLON LFSCR               LAC DLFSCR
          JMP NFILLA
COLON ACRSLF              LAC DCRSLF
          JMP NFILLA
COLON FORMF               LAC DFORM
          JMP NFILLA

DQUOTE,   TEXT ."# .
DBELL,    770774
DBLANK,   770274
CRLF,     767400
DLFSCR,   771274
DCRSLF,   771574
DFORM,    771474
          300000
NFILL,    0
          0
```

```
SEC,      DAP SECX        /CALCULATE SECONDS IN DAY SO FAR
          GTD 1           /GET MINUTES
          CLA"U"SWP
          MUL (60.)
          SCR 1S
          DIO TEMA        /MINUTES IN SECONDS
          RCK 20          /GET MILLISECOND CLOCK
          SCL 1S
          DIV .+1
           1000.
          ADD TEMA
SECX,     JMP .

COLON CLOCK               LAW POINT+1
          DAC STS
          JSP SEC
          SUB SECOND
          SPA
          ADD (60."T"60."T"24.)
          SNM+10
          LAC (300000)
          DAC POINT
          JMP EVALO

COLON RESET
          JSP GNS
          SAS (500000 CLOCK)
          CAL RSTERR
          JSP SEC
          DAC SECOND
          JMP COMRTN
&L
```



```
COLON IS  SJSP 2ARG       /COMPARE TWO STRINGS
          LAW POINT
          JDA TSET
          LAW TPOINT
          JDA CSET

ISA,      JSP TGET
          DAC TEMB
          SAD (CHARACTER L#)
          JMP ISB
          JSP CGET
          SAD (CHARACTER L#)
          JMP ISF
          SAD TEMB
          JMP ISA
ISF,      JSP UGWORD
          JSP UFWORD
          LAW FALSE
          JMP CEMPTY+1

ISB,      JSP CGET
          SAS (CHARACTER L#)
          JMP ISF
          JSP UGWORD
          JSP UFWORD
          LAW TRUE
          JMP CEMPTY+1
&L
```

```
COLON MAKE
          JSP GNS
          SAS (CNAME)     /WHICH VARIETY OF MAKE
          JMP MAKEA       /ONE LINE MAKE
          SJSP EVAL       /GET THE NAME
          JSP VALUEQ      /MAKE SURE SOMETHING RETURNED
          JSP GNS
          SAS (CTHING)
          CAL NMERR2      /SOMETHING EXTRA IN THE NAME
MAKEC,    JSP PPUSH       /SAVE THE NAME
          SJSP EVAL       /GET THE THING
          JSP VALUEQ
          JSP GNS
          SZA
          CAL NMERR1      /SOMETHING EXTRA IN THE THING
          LAC POINT
          AND (300000)
          SAS (200000)    /IS THING IN A STEP
          JMP MAKED       /NO, SO OK
          LAW POINT       /YES, SO COPY IT INTO STORAGE
          JDA TSET
MAKEH,    JSP TGET
          SAD (CHARACTER L#)
          JMP MAKEG
          JDA TSTORE
          JMP MAKEH

MAKEG,    JSP UGWORD
          JSP TDONE
MAKED,    LAC POINT
          DAC TPOINT      /EXCHANGE POINT AND TPOINT
          LOAD POINT+1
          UNLOAD TPOINT+1
          JDA PULL
           POINT+2
           POINT+1
           POINT
          JSP VGET
          JSP DLOOKI      /LOOK UP IN TABLE
           JMP MAKEB      /NOT FOUND
          CLF 1
```

```
MAKEF,    LOAD SDRA       /RESET (OR SET) VARIABLE NAME AND THING
          JDA SGWORD
          LAW 3
          ADD WRDCNT
          IOR TPOINT
          JDA NGWORD      /SAVE WORD COUNT
          DZM TEMA
          LAW SYM-1
          DAP SGETP
MAKEE,    JSP SGET
          JDA NGWORD
          IDX TEMA
          SAS WRDCNT
          JMP MAKEE
          LAC TPOINT+1
          JDA NGWORD      /SAVE VALUE
          LAC TPOINT+2
          JDA NGWORD
          CLA             /PUT Ø AT END IF ON END OF TABLE
          SZF 1
          JDA NGWORD
          JSP UFWORD
          JSP UGWORD
          JMP NILL        /HAVE ALREADY CHECKED END
&L
```

```
MAKEA,    SJSP CALC       /ONE LINE MAKE
          JSP VALUEQ      /MAKE SURE A VALUE RETURNED
          JMP MAKEC

MAKEB,    STF 1           /NOT DEFINED YET
          JSP LOOK        /IS IT SYSTEM SYMBOL
           SVTBL
           JMP MAKEF      /NO-SO GO DEFINE IT
          CAL TOERR3      /CAN'T REDEFINE A BUILT-IN NAME

NGWORD,   Ø               /STORE THRU GWORDP
          DAP NGWRDX
          LAC GWORDP
          SAS GWORDP+3
          JMP NGWRDA
          JSP UGWORD
          LIO I GWORDP+2
          SNI I
          JMP NGWRDB
          LIO I GWORDP+1
          JSP NEWITM
          LAC I GWORDP+1
          DAC I BBPTR2
NGWRDB,   LAC (JMP Ø)
          JDA SGWORD
NGWRDA,   JSP USEDG
          IDX GWORDP
          LAC NGWORD
          DAC I GWORDP
NGWRDX,   JMP .
&L
```

```
COLON RANDOM              LAC RANDA
          LIO RANDA+1     /GENERATE A RANDOM DIGIT
          RCR 7S
          XOR RANDA+1
          LIO RANDA
          DAC RANDA
          DIO RANDA+1
          SCR 9S
          SCR 8S
          DIV .+1
           10.
          SPI
          CMI
          LAI
          IOR (20)
          RAR 6S
          DIP RANDB+1
          LAW RANDB
          JMP CEMPTY+1


COLON TIME                LAW TBLOCK  //TIME/
          DAC STS
          DAC FSA
          GTD 1           /GET PRESENT TIME AND DATE
          STD 10          /SET UP ONLY TIME
          LAC (400000)    /TIME IS ALWAYS A SENTENCE
          DAC POINT
TDCOPY,   LCH I FSA
          SAD (CHARACTER L#)
          JMP TDCOPA
          JDA TSTORE
          JMP TDCOPY

COLON DATE                LAW TBLOCK  //DATE/
          DAC STS
          DAC FSA
          GTD 1
          STD 20
          DZM POINT       /DATE IS ALWAYS A WORD
          JMP TDCOPY

TDCOPA,   JSP UGWORD
          JSP TDONE
          JMP EVALO

TBLOCK,   REPEAT 7,0
RANDB,    300000          007400      000000
&L
```

```
COLON BEFOREP
          SJSP 2ARG       /COMPARE TWO TIME AND DATES
          LAW POINT
          JDA TSET
          JSP BEFORB      /DECODE TIME AND DATE
          UNLOAD TEMD
          LAW TPOINT
          JDA TSET
          JSP BEFORB      /DECODE FIRST TIME AND DATE
          SUB TEMD+1      /COMPARE DATES
          SPA
          JMP EMPTQA      /TRUE
          SZA
          JMP SENTQA      /FALSE
          LAI             /DATES THE SAME, CHECK TIMES
          SUB TEMD
          SPA
          JMP EMPTQA
          JMP SENTQA

BEFORB,   DAP BEFORX
          LAW TBLOCK
          DAC FSA
          DAC STS
BEFORA,   LAC STS
          SAD (JMP TBLOCK+7)
          CAL BEFERR
          JSP TGET
          DCH I STS
          SAS (CHARACTER R#)
          JMP BEFORA
          JSP UGWORD
          DTM             /DECODE TIME
          JMP BEFORC      /MAYBE JUST DATE
          LIA
          LCH I FSA
          SAD (CHARACTER L#)
          JMP BEFORF      /JUST A TIME, SUPPLY TODAY'S DATE
          SZA             /SKIP THE SPACE
          JMP BEFORC
          DDT
          JMP BEFORC
BEFORD,   DAC TEMA
          LCH I FSA
          SAS (CHARACTER L#)
          CAL BEFERR
          LAC TEMA
BEFORX,   JMP .
```

```
BEFORF,   DIO TEMA
          GTD+1
          LIO TEMA
          JMP BEFORX

BEFORC,   CLI             /MIDNIGHT
          LAW TBLOCK
          DAC FSA
          DDT
          CAL BEFERR
          JMP BEFORD      /JUST A DATE
&L
```

```
COLON GO  JSP GNS          /GO TO LINE X
          SAS (700000 TO)
          CAL GOERR1
          JSP GNS
          SAS (500000 LINE)
          CAL GOERR1
          LAC PROD
          SZA I
          CAL GOERR2
          SJSP EVAL        /GET LINE NUMBER
          JSP VALUEQ
          LAW POINT
          JDA EMPTYQ       /MUST BE WORD NON-EMPTY
          CAL LSTER3
          LAC POINT        /MUST BE <= 6 CHARACTERS
          AND (700000)
          SAS (300000)
          CAL LSTER3
          LAW POINT+1
          DAC FSA
          DNM
          CAL LSTER3
          SPQ
          CAL LSTER3
          DAC SNUM
          LCH I FSA
          SAS (CHARACTER L#)
          CAL LSTER3
          LIO XDRA
          JSP FSTEP
          JMP .+2
          CAL ERERR2       /NO SUCH LINE
          JSP CLINE
          JSP UTWORD       /RELEASE PRESENT STEP
          LOAD SDRA
          JDA STWORD
          JMP NILL

WORD JMP T8                NEWSEG COMMANDS 1
&L
```

```
COLON DIFF              SJSP 2ARG
          LIO (600000 DIFF)          /FOR ERROR
          CLL"U"CML     /TRICK SUM INTO SUBTRACTING
          JMP DIFFA

COLON SUM SJSP 2ARG
          CLL
          LIO (600000 SUM)
DIFFA,    LAW SUMD
SUMM,     DAP SUMMX     /SUBRUOTINE FOR GREATER
          DIO TEMC      /SO ERR CAN TELL SUM,DIFF,OR GREATER
          DZM TEME      /COUNTS NEGATIVE ARGUMENTS
          LAC PDLCNT
          DAC SUM1      /TO CLEAR PDL ON RETURN
          LAW POINT
          JDA TSET
          JSP SUMB      /PUT SECOND ARG ON PDL
          DAC SUM1+1    /SAVE PDLCNT
          DAC SUM2
          CLL"U"SZL"U"SCF
          IDX TEME      /SECOND ARG COMPLEMENTED
          DIO SUM1+2    /PDL DRA
          LAC PUSHP
          DAC SUM1+3
          LAC PLCOUNT
          DAC SUM1+4
          LAC SUM1+5    /FREEZE BUFFER IN CORE
          IOR I SUM1+4
          DAC I SUM1+4
          LAW TPOINT
          JDA TSET
          JSP SUMB      /PUT FIRST ARG ON PDL
          DAC SUM2+1
          DIO SUM2+2
          LAC PUSHP
          DAC SUM2+3
          LAC PLCOUNT
          DAC SUM2+4
          LAC SUM2+5
          IOR I SUM2+4
          DAC I SUM2+4
          JDA ACPUSH    /PUSH A NEG NUMBER (400000)
           ACPUSH       /TWICE
          CLL"U"SZL"U"SCF
          IDX TEME      /FIRST ARG COMPLEMENTED
          STF 3
&L
```

```
SUMC,     LAW SUM1
          JDA SPULL
          STF 2
          DAC TEMB
          LAW SUM2
          JDA SPULL
          STF 5
          ADD TEMB
          SZF 1
          IDA"U"CLL"U"CML
          CLF 1
          SUB (10.)
          SMA
          STF 1
          SPA
          ADD (10.)
          JDA ACPUSH
          LAC ACPUSH
          SZL
          JMP SUMG
          SZF 3
          IDA
SUMG,     CLF 3
          CLL
          SAD (10.)
          CLA"U"STF 3
          JDA ACPUSH
          SZF 2
          SZF I 5
          JMP SUMC
          JSP ACPULL
          SZA
          CLL"U"CML
&L
```

```
SUMH,     ADD TEME
          RAR 1S
          SPA
          JSP ACPULL
SUMMX,    JMP .          /SUM AND DIFF JUST CONTINUE

SUMD,     JSP ACPULL
           TEME
          SPA
          JMP SUMZ
          SZL I
          JMP .+3
          CMA
          ADD (9.)
          SZA I
          JMP  SUMD
          DAC TEMB
          LAC (CHARACTER L-)
          SZL
          JDA TSTORE
          LAC TEMB
SUME,     IOR (20)
          RAR 6S
          JDA TSTORE
          JSP ACPULL
           TEME
          SPA
          JMP SUMF
          SZL I
          JMP SUME
          CMA
          ADD (9.)
          JMP SUME
&L
```

```
SUMF,     LAC SUM2+4
          JDA RUNFRZ
          LAC (-200000)
          AND I SUM1+4
          DAC I SUM1+4
          LIO SUM1
          JSP PDLCLR
          DZM POINT
          JSP TDONE
          JMP EVALO

SUMB,     DAP SUMBX
          JSP TGET
          SAD (CHARACTER L-)
          CML             /SET TO PUSH 9'S COMPLEMENT
          SAS (CHARACTER L-)
          SAD (CHARACTER L+)
          JSP TGET        /SKIP SIGN IF THERE
          CLI"U"SWP
          SZL
          LAW 9.
          JDA ACPUSH      /PUSH 0 OR 9 FOR + OR -
          LAI
SUMBA,    RAL 6S
          XOR (20)
          SUB (10.)
          SMA
          CAL SUMERR      /NOT A DIGIT
          ADD (10.)
          SZL I
          JMP .+3
          CMA
          ADD (9.)
          JDA ACPUSH      /PUSH A DIGIT OR 9-DIGIT
          JSP TGET
          SAS (CHARACTER L#)
          JMP SUMBA
          JSP UGWORD
          LAC PDLCNT
          LIO PUSHG
SUMBX,    JMP .

&L
```

```
SPULL,    Ø
          DAP SPULLX
          LAC SPULL
          DAP SPULLA
          IDX SPULL
          DAP SPULLB
          IDX SPULL
          DAP SPULLC
          IDX SPULL
          DAP SPULLD
          IDX SPULL
          DAP SPULLE
          IDA
          DAP SPULLG
SPULLB,   LAC .           /PDL COUNT NOW AT
SPULLA,   SAD .           /PDL COUNT TO STOP AT
          JMP SPULLH
          SUB (1)
          DAC I SPULLB
          IDX SPULLX
          LAW I 1
SPULLD,   ADD .           /CURRENT PTR INTO PDL
          DAC I SPULLD
SPULLC,   SAS .           /PTR TO END OF BUFFER
          JMP SPULLX-2
          DAC TEMD
SPULLG,   LAC .           /FRZ BIT
          CMA
SPULLE,   AND I .         /PTR INTO BCHK FOR FRZING
          DAC I SPULLE
          LIO I TEMD
          JSP GETIT
          LAC BCOUNT
          DAC I SPULL
          LAC I BCOUNT
          IOR I SPULLG
          DAC I BCOUNT
          LAC BBPTR2
          DAC I SPULLC
          ADD (BLNG-4)
          DAC I SPULLD
SPULLH,   LAC I SPULLD
          DAC SPULL
          LAC I SPULL
SPULLX,   JMP .
&L
```

```
SUMZ,     LAC (CHARACTER LØ)          /RETURN +Ø
          JDA TSTORE
          JMP SUMF

SUM1,     Ø
          Ø
          Ø
          Ø
          Ø
          2ØØØØØ
SUM2,     Ø
          Ø
          Ø
          Ø
          Ø
          4ØØØØØ          /FREEZE BIT
&L
```

```
MAXA,     DAP MAXX
          CLL"U"CML
          JSP SUMM        /SUBTRACT THE TWO ARGS
          CLF 1
MAXD,     JSP ACPULL      /CHECK IF ANSWER IS ZERO
           TEME
          SPA
          JMP MAXC        /ZERO
          SZL I
          JMP .+3
          CMA
          ADD (9.)
          SZA I
          JMP MAXD
MAXE,     LAC SUM2+4
          JDA RUNFRZ      /FREE THE STUFF HELD BY SUM
          LAC (-200000)
          AND I SUM1+4
          DAC I SUM1+4
          LIO SUM1
          JSP PDLCLR
MAXX,     JMP .

MAXC,     STF 1           /MEANS ZERO
          JMP MAXE

COLON MAXIMUM
          SJSP 2ARG
          LIO (600000 MAXIMUM)
          JSP MAXA
MAXB,     SZL             /LINK SAYS DIFF NEGATIVE
          JMP EVALO
          LAC TPOINT
          DAC POINT
          LAC TPOINT+1
          DAC POINT+1
          LAC TPOINT+2
          DAC POINT+2
          JMP EVALO

COLON MINIMUM
          SJSP 2ARG
          LIO (600000 MINIMUM)
          JSP MAXA
          CML
          JMP MAXB

COLON GREATQ
          SJSP 2ARG
          LIO (600000 GREATQ)
          JSP MAXA
          LAW TRUE
          SZF I 1
          SZL
          LAW FALSE
          JMP CEMPTY+1
&L
```

```
COLON LOCAL             CLA         /LOCAL COMMAND
          JDA ACPUSH
LOCALC,   JSP GNS
          SZA I
          CAL LOCERR    /LOCAL WHAT?
          SJSP CALC     /EVALUATE NEXT ARGUMENT
          JSP VALUEQ    /MAKE SURE ARG RETURNED SOMETHING
          JDA PULL      /GET BACK COUNT
           TEMD
          JSP VGET      /SET UP VARIABLE NAME IN STANDARD FORM
          JSP LOOK      /IS IT A BUILT-IN NAME
           SVTBL
          JMP .+2
          CAL TOERR3    /YES, ERROR
          DZM TEMB      /PREPARE TO PUT ON PUSH-DOWN LIST
          LAW SYM-1     /BACKWARDS
          DAP SGETP
LOCALA,   JSP SGET
          JDA ACPUSH
          IDX TEMB
          SAS WRDCNT
          JMP LOCALA
          JDA ACPUSH    /SAVE WORD COUNT ON PUSH-DOWN ALSO
          IDX TEMD      /COUNT NUMBER OF VARIABLES
          JDA ACPUSH
          JSP GNS
          SZA I
          JMP LOCALB    /ALL DONE
          SAS (500000 CAND)
          JMP LOCALC+1
          JMP LOCALC

LOCALB,   JDA PULL      /GET NUMBER OF VARIABLES
           TEMA
          LOAD VDRA     /GET VARIABLE DIRECTORY
          JDA SGWORD
          LAC (600000)
          IOR GWORDP+5
          DAC TEMF
          DZM TEMB      /COUNT NUMBER OF VARIABLES OFF LIST
          LAC TEMB
&L
```

```
LOCALD,   SAD TEMA
          JMP LOCALE
          JSP ACPULL      /GET WORD COUNT
          DAC TEME
          CMA
          DAC TEMC
          LAC EMPTY+2     /PUT OUT EMPTY VALUE
          JDA VSTORE
          LAC EMPTY+1
          JDA VSTORE
          JSP ACPULL      /PUT OUT NAME
          JDA VSTORE
          ISP TEMC
          JMP .-3
          LAW 3           /PUT OUT WORD COUNT
          ADD TEME
          IOR EMPTY
          JDA VSTORE
          IDX TEMB
          JMP LOCALD

LOCALE,   JSP UGWORD
          JSP RGWORD      /RELEASE GWORD AND SET VDRA
          DIO VDRA
          DAC VDRA+1
          JMP NILL

COLON BOTH
          JSP GNS
          SAD (500000 CAND)
          JSP GNS
          JSP BOOLE
          JMP BOTHL       /FIRST ARG FALSE SO RT FALSE
BOTHA,    SJSP CALC
          JSP BOOLE
          JMP BOTHL+2     /RETURN FALSE
          LAW TRUE
          JMP CEMPTY+1
```

```
COLON OR
          JSP GNS
          SAD (500000 CAND)
          JSP GNS
          JSP BOOLE
          JMP BOTHA       /FIRST ARG FALSE, RETURN SECOND ARG
          SJSP CALC       /FIRST ARG TRUE, RETURN TRUE
          JSP BOOLE       /JUST MAKE SURE ITS TRUE OR FALSE
          NOP
          LAW TRUE
          JMP CEMPTY+1

BOTHL,    SJSP CALC       /JUST RETURN FALSE
          LAW FALSE
          JMP CEMPTY+1

BOOLE,    DAP BOOLEX      /RTN 1=TRUE, RTN 2=FALSE
          JSP VGET
          JSP LOOK
           TFTABLE
          CAL BOLERR      /NEITHER TRUE OR FALSE
          SPI I
          IDX BOOLEX
BOOLEX,   JMP .

&L
```

```
TRALL,    JSP GNS          /BURY, TRACE AND DIGUP ALL
          SAS (500000 PRCDS)
          CAL ERERR5       /ALL WHAT?
          LODE PDRA
          JDA SGWORD
TRALLA,   JSP RGWORD       /MAIN LOOP
          UNLOAD TEMD
          JSP GWORD
          DAC TEME
          SZA I
          JMP TRALLC       /ALL DONE
          SUB (2)
          JDA NTHWD
          SZA
          JMP TRALLB       /A DEFINED PRCD, TRACE IT
          JSP GWORD        /CONTINUE TO NEXT
          JMP TRALLA

TRALLB,   LOAD TEMD        /MARK AS TRACED
          JDA SGWORD
          LAC TEME
MAGIC,    IOR BITZ         /GETS CHANGED ACCORDING TO FUNCTION
          JDA GWORDS
          SUB (1)
          JDA NTHWD
          JMP TRALLA

TRALLC,   JSP UGWORD
          JSP RESETT
          JMP COMRTN

BITZ,     400000
ANBITZ,   AND BITZ
IBITZ,    IOR BITZ

BURALL,   LAC (200000)     /FAKE TRALL INTO BURYING ALL
          DAC BITZ
          JMP TRALL

DIGALL,   LAC (-200000)
          DAC BITZ
          LAC ANBITZ
          DAC MAGIC
          JMP TRALL

RESETT,   DAP RESETX
          LAC (400000)
          DAC BITZ
          LAC IBITZ
          DAC MAGIC        /VOILA!
RESETX,   JMP .
&L
```

```
COLON BURY
ABURY,      JSP GNS
            SAD (500000 ALL)
            JMP BURALL
            SAS (400000)
            CAL TOERR1      /WHAT PROCEDURE
            JSP FPROD
            SZA I
            CAL EVER3       /UNDEFINED PROCEDURE
            LOAD TEMD
            JDA SGWORD
            LAC (200000)
            IOR TEME        /SET BIT 1
            JDA GWORDS
            JSP UGWORD
            JSP GNS
            SZA I
            JMP NILL
            SAD (500000 CAND)
            JMP ABURY
            JMP ABURY+1

COLON TRACE                 ATRACE,     JSP GNS
            SAD (500000 ALL)
            JMP TRALL
            SAS (400000)
            CAL TRCER1      /ONLY TRACE PROCEDURES
            JSP FPROD
            SZA I
            CAL EVER3       /X NEEDS A MEANING
            LOAD TEMD
            JDA SGWORD
            LAC TEME
            LIO PROD
            SNI I           /IF COMMAND STORED DON'T MAKE ERROR
            JMP .+3
            SPA
            CAL TRCER3      /ALREADY TRACED
            IOR (400000)
            JDA GWORDS
            JSP UGWORD
            JSP GNS
            SZA I
            JMP NILL
            SAD (500000 CAND)
            JMP ATRACE
            JMP ATRACE+1
&L
```

```
COLON DIGUP
ADIGUP,    JSP GNS
           SAD (500000 ALL)
           JMP DIGALL
           SAS (400000)
           CAL TOERR1
           JSP FPROD
           SZA I
           CAL EVER3
           LAC TEME
           RAL 1S
           SMA
           CAL DIGER1
           LOAD TEMD
           JDA SGWORD
           LAC TEME
           AND (-200000)
           JDA GWORDS      /CLEAR BIT 1
           JSP UGWORD
           JSP GNS
           SZA I
           JMP NILL
           SAD (500000 CAND)
           JMP ADIGUP
           JMP ADIGUP+1
&L
```

```
COLON TTLINE            LAW TTLINA+1          /SCANNER LINE NUMBER
          DAC STS
          LAC 76
          SNM+2
          LAW TTLINA
          JMP CEMPTY+1

TTLINA,   300000          000000      000000

COLON PSWORD
          SJSP PSWRDG
COLON LFILE
LFILK,    LIO FILDRA      /GET SENTENCE OF FILE NAMES
          LAC (JMP)
          JDA SGWORD
          JSP GWORD
          SZA
          JMP LFILKX      /COMPACTER RUNNING
LFILJ,    CLF 7
          LAW GWORDA
          DAP TGETF
          JSP GWORD
LFILL,    SZA I           /NONE AT ALL?
          JMP LFILF       /YES. RETURN /EMPTY/
          SZF 2           /ENTRIES?
          JDA ENTRB       /YES. CHECK ERASED OR LOCKED
LFILA,    JSP TGET
          SAD (CHARACTER L#)
          JMP LFILB
LFILH,    JDA TSTORE
          JMP LFILA

LFILB,    STF 3           /NOT EMPTY
          LAC (JMP)
          IOR GWORDP
          DAC GWORDP
          SZF I 2
          JSP GWORD       /SKIP USER NUMBER
          JSP GWORD       /SKIP DRA
          JSP GWORD
LFILN,    SZA I
          JMP LFILG       /ALL DONE
          SZF 2           /ENTRIES?
          JDA ENTRB       /YES, CHECK THINGS
LFILM,    CLA
          JMP LFILH
```

```
LFILG,     LAC (400000)
           DAC POINT        /MARK AS SENTENCE
           JSP TDONE
           JSP UGWORD
           JMP EVALO

LFILKX,    JDA CHKBRK
           LAW 2
           DELAY
           JMP LFILK
&L
```

```
LFILF,    JSP UGWORD
          JMP CEMPTY
COLON ENTRIES             /ENTRIES OF ..
          CLF 7
          JSP VGET        /SET UP FILE NAME
          LAW ENTRA
          DAP DLOOKX
ENTRF,    LIO FILDRA
          LAC (JMP)
          JDA SGWORD
          JSP GWORD
          SZA             /COMPACTER
          JMP ENTRE       /YES
          JMP DLOOKH
ENTRA,    JMP CEMPTY      /NO SUCH FILE
          LIO USER
          LAW 7777
          AND USER
          SPI I
          SAD POINT+1
          STF 1           /OK FOR PRIVATES
          STF 2           /SAY TO CHECK ERASED AND PRIVATES
          LODE POINT+2
          JDA SGWORD
          LAW TSYM
          DAC FWORDP
          DZM FWORDP+4
          JMP LFILJ+1

ENTRE,    LAW 2
          DELAY
          JSP CHKBRK
          JMP ENTRF

ENTRB,    Ø
          DAP ENTRD
          LAC ENTRB
          SPA
          JMP ENTRC       /ERASED
          RAL 1S
          SZF I 1
          SMA
ENTRD,    JMP .           /OK, RETURN IT
          RAR 1S
ENTRC,    JDA NTHWD       /SKIP IT
          SZF I 3
          JMP LFILL
          JMP LFILN
&L
```

```
COLON ACNTNT
          LODE PDRA       //CONTENTS/
          JDA SGWORD
          LAW GWORDA
          DAP TGETF
          CLF 7
ACNTNA,   JSP RGWORD
          UNLOAD SDRA
          JSP GWORD       /WRDCNT
          SZA I
          JMP ACNTNG      /ALL DONE
          RAL 1S
          SPA
          JMP ACNTNB      /HIDDEN, SO SKIP IT
          RAR 1S
          SUB (2)
          JDA NTHWD
          SZA             /DEFINED?
          JMP ACNTNC      /YES, ADD IT TO LIST
ACNTNF,   JSP GWORD       /NO, SKIP # OF ARGS
          JMP ACNTNA      /GET NEXT ONE

ACNTNC,   JSP UGWORD      /A REAL ONE
          LOAD SDRA
          JDA SGWORD
          JSP GWORD       /SKIP WRDCNT
          CLA
          SZF 1           /FIRST WORD?
          JDA TSTORE      /NO, STORE A SPACE
          STF 1           /MARK AS SOMETHING STORED
ACNTND,   JSP TGET
          SAD (CHARACTER L#)
          JMP ACNTNE
          JDA TSTORE
          JMP ACNTND

ACNTNB,   RAR 1S          /HIDDEN, SO SKIP
          SUB (1)
          JDA NTHWD
          JMP ACNTNA

ACNTNE,   LAC (JMP)       /DONE WITH THIS NAME
          IOR GWORDP
          DAC GWORDP
          JSP GWORD
          JMP ACNTNF

ACNTNG,   JSP UGWORD      /ALL DONE
          SZF I 1         /WERE THERE ANY?
          JMP CEMPTY      /NO
          LAC (400000)    /MARK AS SENTENCE
          DAC POINT
          JSP TDONE
          JMP EVALO
```

```
COLON HORN                  SJSP EVAL
          LAW TURTH
          DAP TURTD
          JMP TURTE
COLON LEFT                  SJSP EVAL
          LAW 2677
          JMP .+4
COLON RIGHT                 SJSP EVAL
          LAW 1077
          DAP TURN
          LAW TURN
          DAP TURTD
TURTE,    SJSP EVAL
          LAC POINT         /GET NUMBER
          AND (700000)
          SAS (300000)      /6 CHARACTERS?
          CAL TURTER        /NO. TOO LONG
          LAW POINT+1
          DAC FSA
          DNM
          CAL TURTER
          SPA
          CAL TURTER
          IDA
          CMA
          DAC TURNMA
          LCH I FSA         /ANYTHING ELSE?
          SAS (CHARACTER L#)
          CAL TURTER
TURTA,    ISP TURNMA
          JMP .+1
          JMP COMRTN
TURTD,    LAW .             /TEXT
          TOS
          JMP TURTA

COLON FRONT                 SJSP EVAL
          LAW 0177
          JMP .+4
COLON BACK                  SJSP EVAL
          LAW 1677
          DAP TURTB
          LAW TURTB
          DAP TURTD
          JMP TURTE

TURNMA,   0                 /-# OF TIMES TO TURN
TURN,     TURTB,            770000      757775
          777577            757775      777574
TURTH,    773577            757775      773577      757400
WORD JMP T8                 NEWSEG COMMANDS 2
```

```
COLON LIST              JSP GNS     /LIST WHAT?
          CLF 7
          SAD (500000 ALL)
          JMP LTALL
          SAD (500000 CONTENTS)
          JMP LTCNT
          SAD (500000 FILE)
          JMP LSTF        /LIST FILE
          SAD (500000 LINE)
          JMP LISTA       /LIST LINE
          SAD (500000 ENTRY)
          JMP LSTENT
          SAD (700000 TITLE)          /LIST TITLE
          JMP LSTTTL
          SAD (500000 PRCDS)          /LIST PROCEDURES ON FILE
          JMP LISTPR
          SAD (500000 NAMES)          /LIST NAMES ON FILE
          JMP LISTNM
          SAD (500000 ABBRS)          /LIST ABBRS ON FILE
          JMP LISTAB
          SAD (500000 COMMENT)
          JMP LISTCM
          AND (700000)
          SAS (600000)
          SAD (500000)
          JMP LISTB       /LIST MACHINE PROCEDURE
          SAS (400000)
          CAL TOERR1      /LIST WHAT
          JSP FPROD
          LIO TEME        /IS THIS PROCEDURE HIDDEN
          RIL 1S
          SPI
          STF 2           /YES, SO ONLY PRINT FIRST LINE
          SZA I
          CAL EVER3
          JSP CLINE       /IS IT END OF LINE
          JSP SLINE
          JSP PPROD       /PRINT IT
LISTE,    JSP SLINE
          JMP NILL        /LEAVE

&L
```

```
LISTA,    SJSP EVAL       /EVALUATE NUMBER
          JSP GNS
          SZA I
          JMP .+4         /JUST THAT LINE
          SAS (500000 ON)
          CAL LSTER9      /ONLY ON AFTER LIST
          STF 5           /LIST FROM N ON
          LAC (700000)
          AND POINT
          SAS (300000)
          CAL LSTER4
          LAW POINT+1     /NOW KNOW THAT WORD SUFFICIENTLY SHORT
          DAC FSA
          DNM
           CAL LSTER3
          SPQ
           CAL LSTER3
          DAC SNUM
          LCH I FSA
          SAS (CHARACTER L#)
          CAL LSTER3
LSTTTA,   LIO DDRA        /MUST BE DEFINING A PROCEDURE
          SNI                                    DIO PPDRA
           CAL LSTER2
          JSP FSTEP       /GO SEARCH FOR STEP
           JMP .+2        /FOUND
          JMP LSTAB       /NO SUCH LINE FOUND
LISTAC,   JDA SGWORD
          SZF I 5
          STF 1           /NO, JUST ONE LINE
          LAW NILL
          DAP PPRODX
          JMP PPRODF      /LIST

LSTAB,    SZF I 5
          CAL ERERR2      /NO, NO SUCH STEP
          JMP LISTAC      /YES, SO CONTINUE ON

LSTTTL,   LIO DDRA        /LIST TITLE
          SNI
           CAL LSTER2     /AREN'T DEFINING A PROCEDURE
          LOAD DNAME      /LOOK UP TO SEE IF TRACED
          JDA SGWORD
          JSP GWORD
          DAC TEME        /NEG MEANS TRACED
          JSP UGWORD
          DZM SNUM
          JMP LSTTTA      /NOW LIST LINE 0
```

```
LISTB,    LAW FTBL        /MACHINE PROCEDURE
          DAP LISTC       /SEARCH TABLE FOR IT
LISTC,    LAC .
          SZA I
          CAL EVER2       /"- ISN'T A PROCEDURE."
          SAD I TEXTP
          JMP LISTD       /FOUND IT
          IDX LISTC       /NO FIND
          IDX LISTC
          JMP LISTC
&L
```

```
LISTD,    JSP SLINE       /FOUND PROCEDURE
          LAW LTTXTA
          TOS             /TYPE "TO"
          IDX LISTC
          LAW 7777
          AND I LISTC
          TOS             /TYPE NAME
          LIO I LISTC     /FOR 1 OR 2 ARGS?
          SPI
          JMP LISTDA      /NO ARGS
          RIL 1S
          LAW LTTXTB      /TYPE "/INPUT/"
          SPI
          LAW LTTXTD      /TYPE "/FIRST INPUT/"
          TOS
          LAW LTTXTC
          SPI
          TOS             /TYPE "AND /SECOND INPUT/"
LISTDA,   JSP SLINE
          JSP SLINE
          JMP COMRTN

LTTXTA,   TEXT .TO #.
LTTXTB,   TEXT . /INPUT/#.
LTTXTC,   TEXT . AND /SECOND INPUT/#.
LTTXTD,   TEXT . /FIRST INPUT/#.

DEFINE FF NAME,ARG,ADDR/A
REMOTE [A,                TEXT /NAME#/          REPEAT 1IF P,EXPUNGE A]
          600000 ADDR
          REPEAT 1IF VZ ARG-1,A
          REPEAT 1IF VZ ARG-2,200000 A
TERMINATE FF

DEFINE FG NAME,ADDR/A
REMOTE [A,                TEXT /NAME#/          REPEAT 1IF P,EXPUNGE A]
          500000 ADDR
          400000 A
TERMINATE FG
```

```
FTBL,       FF BOTH,2,BOTH
            FF INITIALS,1,FINIT
            FF DATE-GOTTEN,1,DATEGT
            FF SIZE,1,SIZE
            FF OWNER,1,OWNER
            FF DATE-SAVED,1,DATESV
            FF ASK,1,ASK
            FF BEFOREP,2,BEFOREP
            FF BUTFIRST,1,BUTF
            FF BUTLAST,1,BUTL
            FF COUNT,1,COUNT
            FF DIFFERENCE,2,DIFF
            FF ENTRIES,1,ENTRIES
            FF FIRST,1,FIRST
            FF IS,2,IS
            FF LAST,1,LAST
            FF MAXIMUM,2,MAXIMUM
            FF MINIMUM,2,MINIMUM
            FF EITHER,2,OR
            FF SENTENCE,2,SENT
            FF SUM,2,SUM
            FF THING,1,THING
            FF WORD,2,WRD
            FF WORDP,1,WORDQ
            FF SENTENCEP,1,SENTQ
            FF ZEROP,1,ZEROQ
            FF EMPTYP,1,EMPTQ
            FF NUMBERP,1,NUMQ
            FF GREATERP,2,GREATQ
            FG RANDOM,RANDOM
            FG CLOCK,CLOCK
            FG TIME,TIME
            FG DATE,DATE
            FG REQUEST,REQUEST
            Ø               HERE
&L
```

```
PLINE,    DAP PLINEX      /PRINT A LINE:ASSUME GWORD IS SET
          DZM TEMC
          JSP GWORD       /SKIP RELATIVE COUNT
          SZA I
          JMP PLINEX
          IDX PLINEX
          JSP TLINE       /MAKE SURE WE ARE A LEFT OF PAGE
          LAW PBUFF       /TYPE OUT LINE NUMBER
          DAC STS
          DAC TEMA
          JSP GWORD
          SZA I
          JMP PLINET      /FOR "TO" LINE
          SNM+10          /SET UP LINE NUMBER IN PBUFF
PLINEK,   LCH I TEMA
          SAD (CHARACTER L#)
          JMP PLINEJ
          TYO
          IDX CHARNO
          JMP PLINEK
&L
```

```
PLINET,   LAC TEME
          SMA             /IS IT TRACED?
          JMP PLINEU      /NO
          LAW PPTXTB
          TOS             /"(TRACED)"
          LAW 10
          DAC CHARNO

PLINEU,   LAW 3
          JMP PLINEA-1

PLINEJ,   CLA
          TYO
          IDX CHARNO
          DAC TEMA        /SAVE POSITION FOR CONTINUATION
PLINEA,   JSP GWORD       /GET NEXT SYMBOL
          SZA I
          JMP PLINEB      /DONE WITH LINE
          SAD (CTHING)
          JMP PLINEC      /THING FOR CALL
          SAD (CNAME)
          JMP PLINED      /NAME FOR CALL
          AND (700000)
          RAL 3S
          ADD .+1
          DAP .+1
          JMP .
           JMP PLINEE     /1: COMMENT
           JMP PLINEF     /2: CONSTANT
           JMP PLINEG     /3: VARIABLE
           JMP PLINEH     /4: PROCEDURE NAME
           NOP            /5: MACHINGE PROCEDURE
           NOP            /6: MACHINE PROCEDURE
          LODE LTBL       /7: VERB
          JDA SFWORD      /LOOK UP IN TABLE
PLINEI,   JSP FWORD
          SAD I GWORDP
          JMP PLINEL      /FOUND IT
          JSP FWORD       /COUNT PAST IT
          AND (7777)
          SUB (1)
PLINEV,   ADD FWORDP
          DAC FWORDP
          SUB FWORDP+3
          SPQ
          JMP PLINEI
          DAC TEMD
          JSP FWORDA
          LAC TEMD
          JMP PLINEV
&L
```

```
PLINEL,   JSP FWORD       /WORD COUNT
          LIA
          AND (7777)
          SAL 1S
          ADD I FWORDP
          RIL 1S
          SPI
          IDA
          RIL 1S
          SPI
          IDA
          AND (7777)
          ADD CHARNO
          SUB (75.)
          SMA
          JSP PLINEM
          JMP PLINER

PLINEM,   DAP PLINMX      /CONTINUTATION SUBROUTINE
          JSP SLINE
          DZM TEMC
PLINMA,   CLA
          TYO
          IDX CHARNO
          IDX TEMC
          SAS TEMA
          JMP PLINMA
PLINMX,   JMP .

PLINES,   JSP UFWORD
PLINEN,   CLA             /TYPE SPACE AFTER EACH SYMBOL
          TYO
          IDX CHARNO
          SAD (72.)
          JSP PLINEM
          JMP PLINEA
&L
```

```
PLINEH,   JSP GWORD
          DAC TEMD
          JSP GWORD
          LIO TEMD
          JDA SFWORD
          JSP FWORD
          AND (7777)
          DAC TEMD
          SAL 1S
          ADD TEMD
          ADD CHARNO
          SUB (81.)       /72. PLUS 3 WORDS OF OVER TIMES 3
          SMA
          JSP PLINEM      /CONTINUE TO NEXT LINE
PLINER,   JSP FWORD       /TYPE OUT NAME
          LIA
PLINEP,   CLA
          SCL 6S
          RAR 6S
          SZA I
          JMP PLINER
          SAD (CHARACTER L#)
          JMP PLINES
          TYO
          SAS (770000)
          IDX CHARNO
          JMP PLINEP

PLINEB,   JSP TLINE       /DONE
PLINEX,   JMP .

PLINEC,   LAW I 10        /SET CONTINUATION
          ADD TEMA        /FOR TYPING THING
          DAC TEMA
          JSP PLINEM
          LAW 11          /CHANGE CONTINUATION
          ADD CHARNO
          DAC TEMA
          LAW PLTHING
          TOS
          LAW 7
          JMP PLINEQ

PLTHING,  TEXT . THING: #.
PLNAME,   TEXT . NAME: #.
&L
```

```
PLINED,   JSP PLINEM      /NAME:
          LAW 10
          ADD CHARNO
          DAC TEMA
          DAC CHARNO
          LAW PLNAME
          TOS
          LAW 10
PLINEQ,   ADD CHARNO
          DAC CHARNO
          JMP PLINEN

PLINEG,   LAC (FLEXO // )             /VARIABLE NAME
          JDA PTEXT
          JMP PLINEN

PLINEF,   LAC (FLEXO "" )
          JMP PLINEG+1

PLINEE,   LAC (FLEXO () )
          JMP PLINEG+1
&L
```

```
PTEXT,    Ø                 /TOP SIX BITS FIRST CHAR, NEXT 6 END CHAR
          DAP PTEXTX
          LAC (200000)
          DAC POINT
          STORE GWORDP,POINT+1
          LAW POINT
          JDA TSET
          LAW PBUFF
          DAC FSA
          LAW I 7777
          AND PTEXT
          SZA
          DCH I FSA
          LAC CHARNO
          DAC TEMB
          JMP PTEXTE-1
PTEXTA,   LAC FSA           /MAIN LOOP
          SAS (LAC PBUFF+26.)
          SAD (JMP PBUFF+25.)
          JMP PTEXTF
          JSP TGET
          SAD (CHARACTER L#)
          JMP PTEXTB
          SZA I
          JMP PTEXTC
          SAD (770000)
          JMP PTEXTD
          DCH I FSA
          IDX CHARNO
PTEXTE,   SAS (72.)
          JMP PTEXTA
PTEXTE+2, JSP PLINEM
          LAC TEMB
          SAS TEMA
          JMP PTEXTI
PTEXTF,   CLC               /WORD TOO LONG
          DAC TEMC
PTEXTC,   LAC (CHARACTER L#)
          DCH I FSA
PTEXTH,   LAW PBUFF
          DAC FSA
          TOS
          CLA
          LIO TEMC
          SPI
          JMP PTEXTA
          TYO
          IDX CHARNO
PTEXTN,   DAC TEMB
          JMP PTEXTE
&L
```

```
PTEXTI,   LAW 72.
          SUB TEMB
          ADD CHARNO
          DAC CHARNO
          LAC TEMA
          DAC TEMB
          JMP PTEXTA

PTEXTB,   LCH (ADD PTEXT)
          SZA
          DCH I FSA
          LAC (CHARACTER L#)
          DCH I FSA
          LAW PBUFF
          TOS
          IDX CHARNO
          SAD (72.)
          JSP PLINEM
          LAC (600000)
          IOR GWORDP      /RESET GWORDP FOR OTHER ROUTINES
          DAC GWORDP
PTEXTX,   JMP .

PTEXTD,   DCH I FSA       /WARNING CHARACTERS
          JSP TGET
          SAD (020000)
          JMP PTEXTG
          SAD (110000)    /TAB
          JMP PTEXTJ
          DCH I FSA
          SAS (46)
          SAD (47)
          JMP PTEXTE-1
          JMP PTEXTA

PTEXTG,   LAC (CHARACTER L#)
          DCH FSA
          JMP PTEXTH
&L
```

```
PTEXTJ,   LAC (CHARACTER L#)          /TAB
          DCH FSA
          LAW PBUFF
          DAC FSA
          TOS
          LAW LTABS-1
          DAP PTEXTK      /SEARCH FOR NEXT TAB STOP
PTEXTL,   IDX PTEXTK
          SAD (LAC ELTABS+1)
          JMP PTEXTE+2    /OFF END OF LINE, IMPOSSIBLE TO GET HERE
PTEXTK,   LAC .
          SUB CHARNO
          SPQ
          JMP PTEXTL
PTEXTM,   CLA
          TYO
          IDX CHARNO
          SAS I PTEXTK
          JMP PTEXTM
          JMP PTEXTN

LTABS,    9.
          18.
          27.
          36.
          45.
          54.
          63.
ELTABS,   72.

PBUFF,    REPEAT 27.,0
&L
```

```
PPROD,    DAP PPRODX      /TYPE PROCEDURE
          LIO I GWORDP
          DIO PPDRA       /FOR LATER COMPARISON
          JSP UGWORD
          LODE PPDRA
          JDA SGWORD
          JSP GWORD       /ARGLIST REL COUNT
          SUB (1)
          JDA NTHWD       /SKIP OVER ARGLIST
PPRODF,   JSP PLINE       /TYPE OUT A LINE
           JMP PPRODA
          SZF 1
          JMP PPRODX-1    /IF JUST LISTING FIRST LINES
          SZF 2
          JMP PPRODX-2    /HIDDEN SO JUST FIRST LINE
          JMP PPRODF

PPRODA,   LAC DDRA
          SAD PPDRA
          JMP PPRODX-1    /IF DEFINING PROCEDURE LEAVE OFF END
          LAW PPTXTA
          TOS
          JSP SLINE
PPRODX-1, JSP UGWORD
PPRODX,   JMP .

PPTXTA,   TEXT .END#.
PPTXTB,   TEXT .(TRACED)#.
PPDRA,    Ø               /DRA OF PROCEDURE BEING LISTED


LTALL,    JSP GNS         /LIST ALL
          SZA I
          JMP LTALLA
          SAD (5ØØØØØ FILES)
          JMP LTALF
          SAD (5ØØØØØ PRCDS)
          JMP LTALP
          SAD (5ØØØØØ NAMES)
          JMP LTNAMJ-1
          SAS (5ØØØØØ ABBRS)
          CAL ERERR5      /ALL WHAT?
          JSP CLINE
&L
```

```
COLON LTALB6              JSP SLINE
          LODE ADRA
          DIO SDRA
          DAC SDRA+1
LTALA,    JDA SGWORD
          LAW GWORDA
          DAP TGETF
          LAW 10
          DAC TEMA
          DZM TEMC
          JSP GWORD       /REL PTR
          SZA I
          JMP LTALAD      /ALL DONE
          RAL 2S
          SPA
          JMP LTALAB      /HOARDED SO DON'T TYPE
          JSP GWORD
          SZA I
          JMP LTALAB      /THIS ONE ERASED
          LAW I 1
          ADD GWORDP
          DAC GWORDP
          LAC (CHARACTER M:)
          JDA PTEXT
          LAC (JMP)
          IOR GWORDP
          DAC GWORDP
          JSP LTALAC      /SPACE AFTER COLON
          JSP GWORD
          DAC POINT+1
          JSP GWORD
          IDC
          IDC
          SUB (1)
          DAC POINT+2     /THE POINTER NEEDS BACKING UP 1
          LAC (200000)
          DAC POINT
          JSP RGWORD
          DIO SDRA
          DAC SDRA+1
          JSP UGWORD
          LAW POINT
          JDA TSET
          CLA
          JDA PTEXT
          JSP UGWORD
          JSP TLINE
          LOAD SDRA
          JMP LTALA
&L
```

```
LTALAB,   LOAD SDRA
          JDA SGWORD
          JSP GWORD
          SUB (1)
          JDA NTHWD
          JSP RGWORD
          DIO SDRA
          DAC SDRA+1
          JMP LTALA+1

LTALP,    JSP CLINE       /LIST ALL PROCEDURES
          LODE PDRA
          JDA SGWORD
LTALPA,   JSP RGWORD
          DIO SDRA
          DAC SDRA+1
          JSP GWORD
          CLF 2
          DAC TEME        /FOR  HIDE AND TRACE
          AND (7777)
          SZA I
          JMP LTALPC      /ALL DONE
          SUB (2)
          JDA NTHWD
          SZA
          JMP LTALPB
LTALPD,   JSP GWORD       /GO TO NEXT
          JMP LTALPA

LTALPB,   LAC TEME
          RAL 1S
          SPA
          JMP LTALPD      /SKIP IT, HIDDEN
          SZF I 1
          JSP SLINE
          JSP PPROD
          LOAD SDRA
          JDA SGWORD
          JSP GWORD
          AND (7777)
          SUB (1)
          JDA NTHWD
          JMP LTALPA

LTALPC,   JSP SLINE
          JSP UGWORD
          JMP NILL
&L
```

```
LTALAC,   DAP LALACX
          CLA
          TYO
          IDX CHARNO
          SUB TEMA
          SPA
          JMP LTALAC+1
          SAD (64.)
          JSP PLINEM
LALACX,   JMP .

LTCNT,    STF 1           /LIST CONTENTS, FLG←ONLY FIRST LINES PRINTED
          JSP GNS         /FILE NAME?
          SZA I
          JMP LTCNTA
          SJMP GTCNT      /YES, SO FIND THE FILE

LTCNTA,   JSP SLINE
          JMP LTALP+1     /LIST ALL PROCEDURES,1ST LINE ONLY
LTALAD,   JSP UGWORD
          JMP LISTE

LTALLA,   JDA PUSH
           (LTALB6)
           (LTNAMS)
          JMP LTALP+1

LSTENT,   SJMP LISFLE     /LIST ENTRY .. ..
LTALF,    SJMP GLISTA     /LIST ALL FILES
LSTF,     SJMP GLISTF     /LIST FILE
LTNAMJ-1, JSP CLINE
LTNAMJ,   SJMP LTNAMS
LISTPR,   SJMP LISPRO
LISTNM,   SJMP LISNAM
LISTCM,   SJMP LISCOM
LISTAB,   SJMP LISABB     /LIST FILE STUFF

WORD JMP T8               NEWSEG LIST
&L
```

```
/FLG 1←COMMAND FILE; FLG 2←NO COMMENT

COLON SAVE
          JSP GNS          /MAKE SURE FILE AND ENTRY NAMES LEGAL
          SAS (400000)
          CAL SAVER1
          JSP TWORD
          DAC TEMD
          JSP TWORD
          DAC TEMD+1
          JSP GNS
          SAS (400000)
          CAL SAVER1
          JSP TWORD
          JDA ACPUSH
          JSP TWORD
          JDA ACPUSH
           TEMD            /SAVE THE POINTER FOR LATER
           TEMD+1
          LOAD TEMD        /NO. IS THIS HIS FILE
          JDA SGWORD       /SET UP FILENAME
          LAW .+3
          DAP SAVGTX
          JMP SAVGTA
SAVEP,    JSP FLPUTD       /GET FILE DIRECTORY
          SZA              /IS COMPACTER RUNNING HERE?
          JMP SAVWAT       /YES, WAIT A BIT
          LAW .+3
          DAP DLOOKX
          JMP DLOOKH       /FIND FILE
          JMP SAVEJ        /DOESN'T EXIST SO OK TO SAVE
          LAC POINT+1      /EXISTS. GET OWNER'S #
          SZA I            /IS THERE AN OWNER?
          JMP SAVEJ        /NO. SO OK TO SAVE
          LAC USER
          SPA
          JMP SAVEJ
          AND (7777)
          SAS POINT+1      /USER'S FILE?
          CAL USRERR       /NO
&L
```

```
SAVEJ,    STORE TEXTP,TEMD          /WILL WE HAVE TO EVAL?
          JSP GNS          /ANYTHING THERE?
          SZA
          JMP SAVTA        /YES, EVAL AND SAVE
          JSP UTWORD       /NO, SAVE EVERYTHING
          LOAD TEMD        /RESET TWORD
          JDA STWORD
          CLF 7
SAVTB,    JSP TWORD
          AND (700000)
          SZA I
          STF 2            /NO COMMENT
          SZA
          SAD (100000)     /SOMETHING, BETTER BE COMMENT
          JMP .+2
          CAL DISERR
          LAC I TEXTP
          JDA CHWRD        /GET WRDCNT OF COMMENT
          LAW 1
          DAC SVSIZE       /SIZE OF ENTRY
          LAW FLSGT
          SGIFL+10         /WRITE EMPTY ITEM FOR ENTRY
          DIO FLORG        /DRA OF ENTRY
          LAC (JMP-1)
          JDA SETUP
           400000 SSBASE
          DZM I BBPTR1
          LAW SAVFXA       /EXPUNGE ITEMS ON IOPERR
          DAP SAVFXX
          LAW 10           /NUMBER OF OVERHEAD WORDS
          JDA FLWDS
          GTD+1            /SAVE TIME AND DATE OF LAST GET
          JDA FLWDS
          SWP
          JDA FLWDS
          SWP              /AND OF SAVE
          JDA FLWDS
          SWP
          JDA FLWDS
          LAW 14           /GET INITIALS
          PEEK
          JDA FLWDS        /INITIALS
          LAW 7777
          AND USER
          JDA FLWDS
          CLA              /SAVE ROOM FOR SIZE
          JDA FLWDS
          DZM TEME
          SZF I 2
          JMP SAVFE        /STORE COMMENT
          JMP SAVFNA
&L
```

```
SAVTA,    SJSP CALC       /EVALUATE
          CLF 7
          STF 1           /SAVING TEXT, NOT EVERYTHING
          JMP SAVTB       /GO BACK AND DO COMMENT

SAVTC,    CLA             /NO NAMES
          JDA FLWDS
          JSP FLWDS+1     /AND NO ABBRS
          JSP FLWDS+1     /IDX FLWDP
          LAW POINT
          JDA TSET        /TEXT FOR SAVING
SAVTD,    JSP TGET
          SAD (CHARACTER L#)
          JMP SAVTE       /ALL SAVED
          SAD (760000)    /CRLF SERVES AS EOM
          LAC (CHARACTER L#)          /CONVERT IT
          JDA DCHFLE
          JMP SAVTD

SAVTE,    JDA DCHFLE      /SAVE THE EOM
          JSP UGWORD
          JMP SAVEU       /TERMINATE FILE

SAVFE,    JSP TWORD       /SAVE COMMENT
          JDA FLWDS
SAVFN,    IDX TEME
          SAS WRDCNT
          JMP SAVFE
          LAW I 1
          ADD SSBASE
          DAC SSBASE
          LCH I SSBASE
          SAS (CHARACTER L#)          /FIND THE ALT MODE
          JMP .-2
          CLA             /GET RID OF IT
          DCH SSBASE
&L
```

```
SAVFNA,   LAW GWORDP      /USE THAT AND BEGINNING OF FWORDP AS
          DAC STS         /...7 WORD BUFFER FOR TIME AND DATE
          GTD+1
          STD
          LAW GWORDP
          DAC STS
          LCH I STS
          JDA DCHFLE
          SAS (CHARACTER R#)
          JMP .-3
          JSP FLWDM       /IOR (JMP) AND SSBASE
          SZF 1
          JMP SAVTC       /SAVE TEXT INSTEAD OF EVERYTHING
          LAC (IDX GWORDP)
          DAC GWORDD      /SET UP GWORD TO GET VARIABLE NAMES
          LIO GVDRA
          LAC (JMP BLNG-20)          /GLOBAL NAMES
          JDA SGWORD
SAVFF,    JSP RGWORD
          DAC SDRA+1
          DIO SDRA
SAVEC,    JSP GWORD
          DIP POINT
          AND (7777)
          SZA I
          JMP SAVED
          SUB (2)
          JDA NTHWD
          DAC POINT+1
          JSP GWORD
          DAC POINT+2
          LAW POINT
          JDA EMPTYQ
          JMP SAVFF
          LAC (NOP)
          DAC GWORDD
          LAW POINT
          JDA TSET
&L
```

LAC POINT
AND (40000)
SZA
JMP SAVFF

```
SAVEB,    JSP TGET
          JDA DCHFLE
          SAS (CHARACTER R#)
          JMP SAVEB
          JSP UGWORD
          JSP FLWDM       /IOR (JMP) AND SSBASE
          LAC (IDX GWORDP)
          DAC GWORDD
          LOAD SDRA
          JDA SGWORD
          JSP GWORD
          AND (7777)
          SUB (2)
          LIO I GWORDP
          SPI
          IOR (400000)    /MARK SENTENCES (OTHERWISE CANT RECOG. 1 WORD S.
          JDA FLWDS
          JDA SAVEA
          JSP GWORD
          JSP GWORD
          JMP SAVFF
SAVED,    JDA FLWDS
          JSP UGWORD      /NOW SAVE THE ABBRS
          LAC (NOP)
          DAC GWORDD
          LAC (JMP-1)
          LIO ADRA
          JDA SGWORD
SAVEF,    JSP GWORD
&L
```

```
SAVEF+1,  DIP TEME        /BIT 2=1 MEANS DON'T SAVE
          AND (7777)
          DAC TEMD
          SZA I           /ANY MORE?
          JMP SAVEG       /NOPE
          JSP GWORD
          SZA I
          JMP SAVFH
          LAC TEME
          RAL 2S          /BIT 2 = FROTZ
          SPA
          JMP SAVFH       /PRETENT ERASED AND DON'T SAVE
          LAW I 2
          ADD TEMD
          JDA FLWDS
          SUB (1)
          LIA
          LAC I GWORDP
          JDA FLWDS
          LAI
          JDA SAVEA
          JSP GWORD
          DAC POINT+1
          JSP GWORD
          DAC POINT+2
          LAC (200000)
          DAC POINT
          JSP RGWORD
          UNLOAD SDRA
          JSP UGWORD
          LAW POINT
          JDA TSET
          LCH GWORDP
          JMP .+2
SAVEE,    JSP TGET
          JDA DCHFLE      /NOW SAVING THE VALUE OF THE ABBR
          SAS (CHARACTER R#)
          JMP SAVEE
          JSP UGWORD
          JSP FLWDM
          LOAD SDRA
          JMP SAVEF-1

SAVFH,    LAC TEMD
          SUB (1)
          JDA NTHWD
          JMP SAVEF+1
&L
```

```
SAVEG,    JDA FLWDS
          JSP UGWORD      /NOW FOR THE PROCEDURES
          LIO PDRA
          LAC (JMP-1)
SAVEH-1,  JDA SGWORD
SAVEH,    JSP GWORD
          DAC TEMC        /TO CHECK FOR HIDDEN
          AND (7777)      /IN CASE TRACE
          SZA I
          JMP SAVEU       /ALL DONE WITH PROCEDURES NOW
          SUB (3)
          JDA NTHWD
          JSP GWORD
          SZA             /IS PROCEDURE DEFINED
          JMP SAVEI       /YES, FILE IT
SAVEZ,    JSP GWORD       /NO, SKIP NUMBER OF ARGUMENTS
          JMP SAVEH       /AND GO TO NEXT

SAVEI,    DAC TEMA
          LAC TEMC        /CHECK IF HOARDED
          RAL 2S
          SPA
          JMP SAVEZ       /HOARDED SO SKIP IT
          JSP GWORD
          JSP RGWORD
          UNLOAD SVDRA
          JSP UGWORD
          LODE TEMA
          JDA SGWORD
          JSP GWORD
          JDA NTHWD
          JSP GWORD       /0 STEP
          JSP GWORD       /TO
          LAW SAVES
          DAP FLINEX
          LAC (772100)
          JDA DCHFLE      /SPECIAL TO FOR INPUT
          JDA DCHFLE
          LAC (CHARACTER LB)          /TO GO IN IF HIDDEN
          LIO TEMC
          RIL 1S
          SPI
          JDA DCHFLE      /HIDDEN SO FILE "S"BTO
          LAC (FLEXO TO#)
          JDA DCHFLE      /IF NOT HIDDEN "S"TO
          JDA DCHFLE
          JMP FLINEJ
&L
```

```
FLINE,    DAP FLINEX      /FILE A LINE
          JSP GWORD       /SKIP RELATIVE COUNT
          SZA I
FLINEX,   JMP .
          IDX FLINEX
          LAW POINT+1     /TWO WORD BUFFER
          DAC STS
          DAC TEMA
          JSP GWORD       /FOR STEP NUMBER
          SNM+10
FLINEK,   LCH I TEMA
          SAD (CHARACTER L#)
          JMP FLINEJ
          JDA DCHFLE
          JMP FLINEK

FLINES,   JSP UFWORD
FLINEJ,   CLA
          JDA DCHFLE
          JSP GWORD       /GET NEXT ON LINE
          SZA I
          JMP FLINEX      /NO MORE
          SAD (CTHING)    /HANDLE THREE LINE CALL PROPERLY
          JMP FLINET
          SAD (CNAME)
          JMP FLINEQ
          AND (700000)
          RAL 3S          /WHAT TYPE OF THING IS IT
          ADD .+1
          DAP .+1
          JMP .
           JMP FLINEE     /1: COMMENT
          JMP FLINEF      /2: CONSTANT
           JMP FLINEG     /3: VARIABLE
           JMP FLINEH     /4: PROCEDURE NAME
           NOP            /5: MACHINE PROCEDURE
          NOP             /6: MACHINE PROCEDURE
          LODE LTBL       /7: VERB
          JDA SFWORD      /GET LIST TABLE
FLINEI,   JSP FWORD
          SAD I GWORDP
          JMP FLINEL      /FOUND IT
          JSP FWORD       /COUNT PAST IT
          AND (7777)
          SUB (1)
FLINEV,   ADD FWORDP
          DAC FWORDP
          SUB FWORDP+3
          SPQ
          JMP FLINEI
          DAC TEMD
          JSP FWORDA
          LAC TEMD
          JMP FLINEV
&L
```

```
FLINEQ,   LAC (FLEXO #"Y")
          JMP .+2
FLINET,   LAC (FLEXO #"Z")
          JDA DCHFLE
          JDA DCHFLE
          JMP FLINEJ+1

FLINEL,   JSP FWORD       /WORD COUNT
          JMP FLINER      /AND WRITE IT OUT
&L
```

```
FLINEH,   JSP GWORD       /PROCEDURE NAME
          DAC TEMD
          JSP GWORD
          LIO TEMD
          JDA SFWORD
          JSP FWORD       /IGNORE RELATIVE COUNT
FLINER,   JSP FWORD
          LIA
FLINEP,   CLA
          SCL 6S
          RAR 6S
          SZA I
          JMP FLINER
          SAD (CHARACTER L#)
          JMP FLINES
          JDA DCHFLE
          JMP FLINEP
FLINED,   LAC (CHARACTER L#)
          JMP FLINEJ+1

FLINEG,   LAC (FLEXO // )
          JDA FTEXT
          JMP FLINEJ

FLINEF,   LAC (FLEXO "" )
          JMP FLINEG+1

FLINEE,   LAC (FLEXO () )
          JMP FLINEG+1

FTEXT,    Ø
          DAP FTEXTX
          LAC (2ØØØØØ)
          DAC POINT
          STORE GWORDP,POINT+1
          LAW POINT
          JDA TSET
          LAC FTEXT
          JDA DCHFLE
          DAC FTEXT
FTEXTA,   JSP TGET
          SAD (CHARACTER L#)
          JMP FTEXTB
          JDA DCHFLE
          JMP FTEXTA

FTEXTB,   LAC FTEXT
          JDA DCHFLE
          LAC (6ØØØØØ)
          IOR GWORDP
          DAC GWORDP
FTEXTX,   JMP .
&L
```

```
SAVET,     LAC (772100)    /SPECIAL END FOR INPUT
           JDA DCHFLE
           JDA DCHFLE
           LAC (FLEXO END)
           JDA DCHFLE
           JDA DCHFLE
           JDA DCHFLE
           LAC (CHARACTER L#)
           JDA DCHFLE
           JSP UGWORD
           LOAD SVDRA
           JMP SAVEH-1
SAVEU,     LAC (772174)    /DONE SIGNAL
           JDA DCHFLE
           JDA DCHFLE
           JDA DCHFLE
           JSP FLPUTB      /WRITE OUT ITEM REFERENCED BY FLWD
           JSP UGWORD
           LIO FLORG
           JSP GETIT       /GET FIRST ITEM OF ENTRY
           LAW 7
           ADD BBPTR2
           DAP .+2         /POINT TO SIZE WORD
           LAC SVSIZE
           DAC .
           LIO FLORG
           LAC BBPTR
           WAI+2

/NOW REWRITE THE DIRECTORIES. REWRITE ERRORS GENERALLY MEAN
/RETURN INTO HERE TO TRY AGAIN.

           JSP SAVGET      /LIKE VGET
           LAW SAVEY
           DAP SAVFXX      /SET UP TO RESTART HERE
SAVEY,     JSP FLPUTD      /GET THE DIRECTORY
           SZA             /THIS DIRECTORY COMPACTING?
           JMP SAVEW       /YES, WAIT THEN START AGAIN AT SAVEY
           LAW SAVFY
           DAP DLOOKX
           JMP DLOOKH      /FIND THIS FILE
SAVFY,     JMP FLPUT       /NO SUCH FILE, CREATE IT
SAVFY+1,   LIO POINT+2     /DRA OF ENTRY DIR FOR THIS FILE
           DIO TEMA
           JSP SAVGET      /SET UP ENTRY NAME
           LAW SAVFW
           DAP SAVFXX      /SET NEW RESTART ADDR
&L
```

```
SAVFW,    LIO TEMA
          STF 6           /ONLY ONE INFO WORD IN ENTRY DIR
          JSP DLOOK
          JMP ENPUT       /NO SUCH ENTRY. CREATE IT
          LOAD SDRA       /FOUND IT
          JDA SGWORD      /GET RELPTR
          JSP GWORD
          IOR (400000)    /TO MARK IT ERASED
          DAC I GWORDP
          LAW I 3         /AND REWRITE ITEM
          ADD GWORDP+5
          LIO I GWORDP+1
          WAIFL+1
          JMP SAVFX
          LAW I 1
          ADD DLOOKX      /RESET DLOOK RETURN
          DAP DLOOKX
          LAC (400000)
          IOR I GWORDP+4 /REFREEZE BUFFER
          DAC I GWORDP+4
          LAC I GWORDP
          STF 6
          JMP DLOOKB+1    /CONTINUE TO END OF DIRECTORY

/PUT IN NEW ENTRY
ENPUT,    JSP FLPUTC      /SET UP FLWDS
          LAW 2
          ADD WRDCNT
          SZF 1           /COMMAND FILE?
          IOR (40000)     /YES
          JDA FLWDS       /SAVE WRDCNT
          JSP FLPUTA      /SAVE WRDCNT NUMBER OF WORDS
          LAC FLORG       /SAVE DRA OF NEW ENTRY
          JDA FLWDS
          CLA
          JDA FLWDS       /MARKS END OF DIRECTORY
          JSP FLPUTB      /WRITE OUT ITEM
          SZF I 2
          JMP COMRTN
          JMP NILL
```

```
/ADD A NEW FILE TO THE DIRECTORY
/AND WRITE AN EMPTY ENTRY DIRECTORY
FLPUT,     JSP FLPUTC      /AIM FLWDS INTO END OF DIRECTORY
           LAW 3
           ADD WRDCNT
           JDA FLWDS       /SAVE WRDCNT
           JSP FLPUTA      /SAVE WRDCNT NUMBER OF WORDS
           LAW 7777
           AND USER
           JDA FLWDS       /SAVE USER'S NO
           LAW FLSGT
           SGIFL+10        /WRITE A ZERO ITEM FOR ENTRY DIRECTORY
           DIO POINT+2     /WHERE DLOOK WOULD HAVE PUT IT
           LAI
           JDA FLWDS
           CLA
           JDA FLWDS       /MARK END
           JSP FLPUTB      /WRITE IT OUT
           JMP SAVFY+1
&L
```

```
FLPUTA,   DAP FLPUTX      /SAVE WRDCNT NUMBER OF WORDS
          DZM TEMB
          JSP SGET
          JDA FLWDS
          IDX TEMB
          SAS WRDCNT
          JMP FLPUTA+2
FLPUTX,   JMP .

SAVGET,   DAP SAVGTX      /LIKE VGET EXCEPT IT WORKS
          JSP ACPULL
           TEMB
          LIO TEMB
          JDA SGWORD
SAVGTA,   LAW GWORDA
          DAP TGETF
          JSP FSET
          JSP GWORD       /SKIP REL PTR
          JSP TGET
          JDA FSTORE
          SAS (CHARACTER R#)
          JMP .-3
          JSP UFWORD
          JSP UGWORD
SAVGTX,   JMP .

FLPUTB,   DAP FLPUTY      /WRITE ITEM ADDR BY FLWD
          LAW I 3
          ADD SSBASE+5
          LIO I SSBASE+1
          WAIFL+1
          JMP SAVFX
          DZM I SSBASE+1
          DZM I SSBASE+4
FLPUTY,   JMP .

FLPUTC,   DAP FLPUTZ      /SETUP TO FIND ENTRY OR FILE IN DIRECTORY
          LOAD SDRA
          JDA SETUP
           400000 SSBASE
          LAW SYM-1
          DAP SGETP
FLPUTZ,   JMP .
```

```
FLPUTD,    DAP FLPUTW
           LIO FILDRA
           LAC (JMP)
           JDA SGWORD
           JSP GWORD
FLPUTW,    JMP .

SAVFX,     LAW 2000          /IOPERR. WAS IT REWRITE NUMBER?
           SAS ERCODE
           JMP SAVFXA        /NO. ERASE ENTRY, THEN QUIT
SAVFXX,    JMP .             /YES. DISPATCH TO PROPER RESTART

SAVFXA,    LIO FLORG         /EXPUNGE STUFF JUST WRITTEN
           JSP GETIT
           LAC BBPTR
SAVFXB,    RAIFL
           EAIFL
           LIO I BBPTR1
           SNI I
           JMP SAVFXB
           DZM I BCOUNT
           DZM I BDRA
           CAL IOPERR
&L
```

```
SAVEA,    Ø
          DAP SAVEAX      /FILE  SAVEA-1 WORDS
          LAW 7777
          AND SAVEA
          DAC SAVEA
          DZM TEMD
SAVEAB,   IDX TEMD
          SAD SAVEA
SAVEAX,   JMP .
          JSP GWORD
          JDA FLWDS
          JMP SAVEAB

FLWD,     DAP FLWDX       /GET A WORD FROM FILE, LIKE GWORD
          LAC SSBASE
          SAD SSBASE+3
          JSP FLWDA
          IDX SSBASE
          LAC I SSBASE
FLWDX,    JMP .

FLWDA,    DAP FLWDB
          DIO DRUMI       /SAVE THE IO
          LIO I SSBASE+2
          SNI I
          JMP FLWDF
          IDX SVSIZE      /SIZE OF ENTRY
          LAW FLSGT
          SGIFL+1Ø        /WRITE A ZERO ITEM
          LAW I 3
          ADD SSBASE+5
          DIO I SSBASE+2
          LIO I SSBASE+1
          WAIFL+1
          JMP SAVFX
          LIO I SSBASE+2
          DZM I SSBASE+2
          IDA
          DAP .+1
          DZM .           /RESET REWRITE NUMBER TO Ø
FLWDG,    DIO I SSBASE+1
          LAC (JMP-1)
          ADD SSBASE+5
          DAC SSBASE
          LIO DRUMI
FLWDB,    JMP .
```

```
FLWDF,      LAW I 3
            ADD SSBASE+5
            RAIFL
            JMP FLWDG

FLWDS,      Ø
            DAP FLWDSX
            JSP FLWD
            LAC FLWDS
            DAC I SSBASE
FLWDSX,     JMP .
&L
```

```
DCHFLE,   0
          DAP DCHFLX
          LAC SSBASE
          SAD SSBASE+3
          JSP FLWDA
          LAC DCHFLE
          DCH I SSBASE
DCHFLX,   JMP .

FLWDM,    DAP FLWDMX
          LAC (JMP)
          IOR SSBASE
          DAC SSBASE
FLWDMX,   JMP .

FLSGT,    040000+BASE     /SGTBL FOR WRITING EMPTY FILE ITEMS
          1
          050000
          97.
          -0

SAVES,    LAC (CHARACTER L#)
          JDA DCHFLE
          JSP FLINE
          JMP SAVET
          JMP SAVES
&L
```

```
SAVEW,      LAW 2           /DELAY 2 SECONDS FOR FILE COMPACTING
            DELAY
            JMP SAVEY       /TRY AGAIN

SAVWAT,     LAW 2           /WAIT A BIT
            DELAY
            JSP CHKBRK
            JMP SAVEP

FLORG,      Ø               /DRA OF WRITTEN FILE
SVSIZE,     Ø               /SIZE OF ENTRY
SVDRA,      REPEAT 2,Ø      /TEM STORAGE

WORD JMP T8                 NEWSEG SAVE
&L
```

```
/FLG 1←HOARD OR SHARE, NOT REAL GET
/FLG 2←GETTING THE INITIALIZING FILE
/FLG 3←GETTING A HOARDED FILE
/FLG 4←OK TO GET EVEN IF LOCKED


COLON GET CLF 7
AGET,     JSP GTFND       /SET UP FILE NAME
          LIO FILDRA
          LAC (JMP)       /SKIP ADDR OF INITIAL FILE
          JDA SGWORD
          JSP GWORD
          SZA             /NONZERO←COMPACTER RUNNING
          JMP GETO
          LAW GETA
          DAP DLOOKX
          JMP DLOOKH      /FIND THE FILE

GETO,     LAW 2
          DELAY
          JSP CHKBRK
          JMP AGET+1

GETA,     CAL GETER1      /NO SUCH FILE
          LIO POINT+2
          DIO TEMA        /SAVE DRA OF ENTRY DIRECTORY
          LAW 7777        /CHECK OWNERSHIP AND WHETHER LOCKED
          AND USER
          LIO USER
          SPI
          JMP GETC+1      /WHEEL USER. ANYTHING OK
GETC,     SAD POINT+1     /IS THIS THE OWNER
          STF 4           /YES, OK IF PRIVATE
          SZF I 4
          SZF I 1
          JMP .+2
          CAL USRERR      /HOARD,SHARE,LOCK,UNLOCK ON ANOTHER'S FILE
          LAC POINT+1
          SZA I
          SZF I 1
          JMP .+2
          CAL USRER2      /HOARD,SHARE,LOCK,UNLOCK A PUBLIC FILE
          JSP GTFND       /SETUP ENTRY NAME
GETRST,   LIO TEMA        /RETURN TO HERE IF REWRITE TROUBLE IN H.S
          STF 6
          JSP DLOOK       /LOOK UP ENTRY
          CAL GETER3      /NO SUCH ENTRY
          LAC POINT
          SMA             /ERASED?
          JMP GETJ        /NO, FOUND IT
          LAW I 1         /YES, LOOK FOR ANOTHER
          ADD DLOOKX      /RESET RETURN
          DAP DLOOKX
          STF 6
          JMP DLOOKH
```

```
GETJ,     RAL 1S
          SZF I 4          /OK TO GET LOCKED FILES?
          SMA              /NO. SKIP IF LOCKED
          JMP GETB         /GET THE FILE
          CAL GETER3       /PRIVATE. SAY IT ISN'T THERE

GETB,     RAL 1S
          SPA
          STF 3            /HOARDED
          SZF 1
          JMP HOARDA       /NOT REALLY GETTING
          LIO POINT+1
GETN,     LAC (JMP-1)
          JDA SETUP
           400000 SSBASE
          JSP GTWD         /COUNT OF OVERHEAD
          DAC TEME         /SAVE IT
          IDX SSBASE
          GTD+1
          DAC I SSBASE     /UP DATE INFO
          IDX SSBASE
          DIO I SSBASE
          LAW I 3
          ADD SSBASE+5
          LIO I SSBASE+1
          SZF I 2          /DON'T REWRITE INITIAL FILE
          WAIFL+1          /WRITE IT BACK OUT
          NOP              /SO SOMEONE ELSE IS READING IT TOO
          LAW I 3          /SKIP OVERHEAD
          ADD TEME
          ADD SSBASE
          DAC SSBASE
          LAW 3
          DAC TEMA         /SET CONTINUATION
          LAC FLPTR
          LIO PROD
          SZA I
          SNI I
          JMP GETDB        /SKIP TYPING COMMENT IF NOT A DIRECT GET
          SZF 2
          JMP GETDB        /SKIP COMMENT IF INITIALIZATION
          JSP GTEXT        /AND TYPE IT
```

```
GETD,     JSP TLINE
          JDA PUSH
           VDRA          /MAKE SURE NEW NAMES ARE GLOBAL
           VDRA+1
          LAC GVDRA      /RESET VDRA TO GLOBALS
          DAC VDRA
          LAC (JMP BLNG-20)
          DAC VDRA+1
GETDA,    CLF 1
          LAW TSYM       /NOW LOAD VARIABLES
          DAC FWORDP
          DZM POINT      /REMOVE SENTENCE MARK
          DZM FWORDP+4
GETH,     JSP GTWD
          SZA I
          JMP GETFA      /NO MORE VARIABLES
          JSP CHKBRK
          LAW I 1
          ADD SSBASE
          DAC SSBASE     /BACK TO BEGINING OF THING
GETE,     JSP LCHFLE
          SZA I
          STF 1
          SAD (CHARACTER L#)
          JMP .+3
          JDA TSTORE
          JMP GETE
          JSP TDONE
          LAC POINT+1
          DAC TPOINT+1
          LAC POINT+2
          DAC TPOINT+2
          JSP GTWDM      /IOR (JMP) AND SSBASE
          JSP FSET
          JSP GTWD
          LIA
          LAC POINT
          SZF I 1        /SENTENCE?
          SPI
          IOR (400000)   /YES
          DAC TPOINT
          CLF 1
&L
```

```
GETF,     JSP LCHFLE      /STORE THE NAME
          JDA FSTORE
          SAS (CHARACTER R#)
          JMP GETF
          JSP GTWDM
          LOAD POINT+1    /GUARD FROM DESTRUCTION BY DLOOK
          UNLOAD TEMD
          JSP DLOOKI
           STF 1          /WILL SAY DON'T PUT A Ø AFTER IT
GETG,     LOAD TEMD
          UNLOAD POINT+1
          LOAD SDRA
          JDA SGWORD
          LAW 3
          ADD WRDCNT
          IOR TPOINT
          JDA GTNGWD
          DZM TEMA
          LAW SYM-1
          DAP SGETP
GETGA,    JSP SGET
          JDA GTNGWD
          IDX TEMA
          SAS WRDCNT
          JMP GETGA
          LAC TPOINT+1
          JDA GTNGWD
          LAC TPOINT+2
          JDA GTNGWD
          CLA
          SZF 1
          JDA GTNGWD      /MARKS THE END OF THE TABLE
          JSP UFWORD
          JSP UGWORD
          JMP GETDA
GETFA,    JDA PULL        /SET VDRA BACK UP NOW
           VDRA+1
           VDRA
GETFB,    JSP FSET        /NOW GET THE ABBREVIATIONS
          JSP GTWD
          SZA I
          JMP GETM
          JSP CHKBRK
GETK,     JSP LCHFLE
          JDA FSTORE
          SAS (CHARACTER R#)
          JMP GETK
          JSP GTWDM
          JSP UFWORD
          LIO ADRA
          JSP DLOOK
           JMP GETKA
&L
```

SZF 3
IOR (40000)

hoa, dec! ✓

```
GETKB,     LOAD AVALUE
           JDA SFWORD
           DZM CHCNT
           DZM WRDCNT
GETKD,     JSP LCHFLE
           JDA FSTORE
           SAS (CHARACTER R#)
           JMP GETKD
           JSP GTWDM
           JSP UGWORD
           LOAD SDRA
           JDA SGWORD
           JSP GWORD
           AND (-100000)  /REMOVE HOARD BIT IF THERE
           SZF 3
           IOR (100000)   /AND SET IT IF NECESSARY
           DAC I GWORDP
           JSP USEDG
           LAC I GWORDP
           SUB (3)
           JDA NTHWD
           LAC AVALUE+1
           SAS (JMP BLNG-4)
           JMP GETL
           LIO AVALUE
           JSP GETIT
           LAC I BBPTR1
           DAC AVALUE
           CLA
GETL,      IDC
           DAC AVALUE+1
           LAC AVALUE
           JDA GWORDS
           LAC AVALUE+1
           JDA GWORDS
           JSP UGWORD
           STORE FWORDP,AVALUE
           JSP UFWORD
           JMP GETFB      /NEXT ABBR
&L
```

```
GETKA,    LAW SYM-1
          DAP SGETP
          LOAD SDRA
          JDA SGWORD
          LAW 3
          ADD WRDCNT
          JDA GWORDS
          DZM TEMA
GETKC,    JSP SGET
          JDA GWORDS
          IDX TEMA
          SAS WRDCNT
          JMP GETKC
          CLA
          JDA GWORDS
          JSP GWORDS+1
          JSP GWORDS+1
          JSP UGWORD
          JMP GETKB

GETM,     LAC FLPTR
          SZA I
          JMP GETMD       /NOT IN A FILE NOW
          LAC I FLPTR     /GET DRA OF PRESENT FILE
          DAC TEMA
          LAW I BTBL      /CALC REL TEXT PTR FROM PHYSICAL
          ADD FLPTR
          MUL (BLNG)
          SCL 9S
          SCL 8S
          ADD (BASE)      /NOW HAVE BEG OF BUFFER
          CMA
          ADD FLPTR+1
          DAC TEMB        /REL TXT PTR
          LAW BCHK-BTBL
          ADD FLPTR
          JDA RUNFRZ      /RELEASE CORE BUFFER
&L
```

```
GETME,    JSP UTWORD      /RELEESE COMMAND
          STORE TEXTP,TEMD          /SAVE COMMAND PTR
          JDA PUSH        /AND PUSH EVERYTHING
           HOARDB
           PROD
           PROD+1
           XDRA
           RNUM
           TEMD           /COMMEND
           TEMD+1
           CDRA           /DIRECT COMMAND
           CDRA+1
           TEMA           /DRA OF FILE IF ANY
           TEMB           /OLD REL PTR TO FILE IF ANY
          LOAD DCDRA      /RESET CDRA
          UNLOAD CDRA
          LAC (100000)
          IOR HOARDB
          SZF 3           /HOARDED FILE?
          DAC HOARDB      /YES. SET BIT
          LAC SSBASE
          DAC FLPTR+1
          LAW 7777
          AND SSBASE+1
          DAC FLPTR
          SJMP SINPUT     /AND READ FILE

GETMD,    DZM TEMA        /NOT FROM FILE NOW
          DZM TEMB        /SO WILL PUSH ZEROS
          LAC STBL+GETSEG-1          /RESET STBL
          DAC STBL+INSEG-1
          JMP GETME

&L
```

```
GETDB,    JSP LCHFLE      /SKIP TYPING COMMENT SINCE COMMANDS FROM FILE
          SAS (CHARACTER L#)
          JMP GETDB       /SKIP THROUGH IT
          JSP GTWDM
          JMP GETD+1

GTWD,     DAP GTWDX       /GET A WORD FROM FILE, LIKE GWORD
          DIO DRUMI
          LAC SSBASE
          SAD SSBASE+3
          JSP GTWDA
          IDX SSBASE
          LAC I SSBASE
          LIO DRUMI
GTWDX,    JMP .

GTWDA,    DAP GTWDB
          LIO I SSBASE+2
          LAW I 3
          ADD SSBASE+5
          RAIFL
          DIO I SSBASE+1
          LAC (JMP-1)
          ADD SSBASE+5
          DAC SSBASE
GTWDB,    JMP .

LCHFLE,   DAP LCHFLX
          LAC SSBASE
          SAD SSBASE+3
          JSP GTWDA
          LCH I SSBASE
LCHFLX,   JMP .

GTWDM,    DAP GTWDMX
          LAC (JMP)
          IOR SSBASE
          DAC SSBASE
GTWDMX,   JMP .
&L
```

```
GTNGWD,   Ø                 /COPIED NGWORD
          DAP GTNGX
          LAC GWORDP
          SAS GWORDP+3
          JMP GTNGA
          JSP UGWORD
          LIO I GWORDP+2
          SNI I
          JMP GTNGB
          LIO I GWORDP+1
          JSP NEWITM
          LAC I GWORDP+1
          DAC I BBPTR2
GTNGB,    LAC (JMP Ø)
          JDA SGWORD
GTNGA,    JSP USEDG
          IDX GWORDP
          LAC GTNGWD
          DAC I GWORDP
GTNGX,    JMP .

GTFND,    DAP GTFNDX        /SETUP FILE OR ENTRY NAME FOR LOOKUP
          JSP GNS
          SAS (4ØØØØØ)
          CAL SAVER1
          JSP TWORD
          DAC TEMB
          JSP TWORD
          LIO TEMB          /THIS SETS UP NAME LIKE VGET
          JDA SGWORD
          LAW GWORDA
          DAP TGETF
          JSP FSET
          JSP GWORD         /SKIP WRDCNT
          JSP TGET
          JDA FSTORE
          SAS (CHARACTER R#)
          JMP .-3
          JSP UFWORD
          JSP UGWORD
GTFNDX,   JMP .
&L
```

```
GTEXT,    DAP GTEXTX      /TYPE OUT COMMENT
          DZM TEMC
          LAW GBUFF
          DAC FSA
          LAC CHARNO
          SZF 3           /HOARDED?
          JMP GTEXTI      /YES
GTEXTK,   LAC POINT
          RAL 1S
          SPA
          JMP GTEXTL      /LOCKED
GTEXTJ,   DAC TEMB
          SZF 1           /COMMENT OR CONTENTS?
          JMP GTEXTA      /CONTENTS
          LAC (CHARACTER L()          /COMMENT
          DCH I FSA
          JMP GTEXTE-1

GTEXTA,   LAC FSA
          SAS (LAC GBUFF+26.)
          SAD (JMP GBUFF+25.)
          JMP GTEXTF
          JSP LCHFLE
          SAD (CHARACTER L#)
          JMP GTEXTB
          SZA I
          JMP GTEXTC
          SAD (770000)
          JMP GTEXTD
          DCH I FSA
          IDX CHARNO
GTEXTE,   SAS (72.)
          JMP GTEXTA
          JSP GLINEM
          LAC TEMB
          SAS TEMA
          JMP GTEXTM
GTEXTF,   CLC
          DAC TEMC
GTEXTC,   LAC (CHARACTER L#)
          DCH I FSA
GTEXTH,   LAW GBUFF
          DAC FSA
          TOS
          CLA
          LIO TEMC
          SPI
          JMP GTEXTA
          TYO
          IDX CHARNO
          DAC TEMB
          JMP GTEXTE

&L
```

```
GTEXTB,   LAC (CHARACTER L))
          SZF I 1          /COMMENT OR CONTENTS
          DCH I FSA
          LAC (CHARACTER L#)
          DCH I FSA
          LAW GBUFF
          TOS
          SZF 1
          JMP GTEXTX       /DON'T GO BACK IF IN CONTENTS
          JSP GTWDM
GTEXTX,   JMP .

GTEXTI,   LAC CHARNO       /HOARDED
          SUB (63.)
          SMA              /ROOM FOR "(HOARDED)"?
          JSP GLINEM       /NO
          LAW GTXTK
          TOS
          LAW 11
          ADD CHARNO
          DAC CHARNO
          JMP GTEXTK

GTEXTL,   LAW I 50.
          ADD CHARNO
          SMA
          JSP GLINEM
          LAW GTXTA
          TOS
          LAW 10
          ADD CHARNO
          DAC CHARNO
          JMP GTEXTJ

GTXTA,    TEXT /(LOCKED)#/
GTXTK,    TEXT /(HOARDED)#/
GTEXTD,   DCH I FSA
          JSP LCHFLE
          SAD (020000)
          JMP GTEXTG
          DCH I FSA
          SAS (46)
          SAD (47)
          JMP GTEXTE-1
          JMP GTEXTA
```

```
GLINEM,   DAP GLINMX
          JSP SLINE
          DZM TEMC
GLINMA,   CLA
          TYO
          IDX CHARNO
          IDX TEMC
          SAS TEMA
          JMP GLINMA
GLINMX,   JMP .

GTEXTG,   LAC (CHARACTER L#)
          DCH FSA
          JMP GTEXTH

GTEXTM,   LAW 72.
          SUB TEMB
          ADD CHARNO
          DAC CHARNO
          LAC TEMA
          DAC TEMB
          JMP GTEXTA

GBUFF,    REPEAT 27.,Ø
```

```
COLON GTFINI            JSP UTWORD  /RELEASE COMMAND AND LEAVE FILE
          LAW BCHK-BTBL
          ADD FLPTR
          JDA RUNFRZ    /RELEASE BUFFER
          JSP ACPULL    /REMOVE 2 RETURNS
           TEMB         /THROW AWAY THIS ONE TOO
           TEMB
           TEMA         /DRA IF ANY
           CDRA+1
           CDRA
           TEMD+1
           TEMD
           RNUM
           XDRA
           PROD+1
           PROD
           HOARDB
          LOAD TEMD
          JDA STWORD    /GET BACK COMMAND
          LIO TEMA
          SNI           /FROM A FILE?
          JMP GTFINC    /NO.
          JSP FITM      /YES, GET IT BACK IN
          JMP GTFINA                /NOT IN CORE, GO GET IT
          JSP GITM
GTFINB,   LAC (400000)
          DIP I BCOUNT  /FREEZE IT
          LAC BDRA
          DAP FLPTR
          LAC BBPTR
          ADD TEMB
          DAC FLPTR+1
GTFIND,   JMP COMRTN
```

```
GTFINA,   CLI
          JSP FITM
          JSP FBUF
          JSP GITM
          LAC BBPTR
          LIO TEMA
          RAIFL
          DIO I BDRA
          JMP GTFINB

GTFINC,   LAC PERMIN      /PUT BACK TT INPUT
          DAC STBL+INSEG-1
          DZM FLPTR
          DZM FLPTR+1
          JMP GTFIND

&L
```

```
COLON REINIT            LIO INITFL  /INITIALIZING FILE
          SNI
          JMP POP       /NO INIT FILE
          CLF 7
          STF 2         /SO NO COMMENT TYPED
          STF 3         /SO ACTS HOARDED
          JMP GETN

COLON UNLOCK
          LAC (600000)
          JMP HOARDE
COLON LOCK
          LAC (200000)
          JMP HOARDE
COLON SHARE
          LAC (500000)  /SIGN BIT MEANS REMOVE OTHER BIT
          JMP .+2
COLON HOARD
          LAC (100000)
HOARDE,   DAC HOARDD
          CLF 7
          STF 1
          JMP AGET

HOARDA,   LOAD SDRA
          JDA SGWORD
          JSP GWORD
          LIO HOARDD
          CMI
          NAI           /REMOVE THE BIT (AND SIGN BIT BUT THAT'S NOT THE
%%%%%)
          CMI
          SPI I
          IAI
          DAC I GWORDP
          LIO I BDRA
          LAC BBPTR
          WAIFL+1       /REWRITE DIRECTORY
          JMP HOARDC    /TRY AGAIN
          DZM I BDRA
          DZM I BCOUNT
          JMP COMRTN

HOARDC,   LAW 2000
          SAS ERCODE
          CAL IOPERR    /REAL ERROR
          DZM I BDRA
          DZM I BCOUNT
          JMP GETRST
HOARDD,   0

WORD JMP T8             NEWSEG GET FILE IN
&L
```