```
RUP     IDENT   3/7/68
* STRING AND FLOATING POINT SYSPOPS

SWCI    OPD     15700000B,1,1   SYSPOPS WITH BIT 0 REMOVED ...
SCIO    OPD     16100000B,1,1   ... FOR USE BY SYSTEM MODE ROUTINES
SBRS    OPD     17300000B,1,1
SSKSE   OPD     16300000B,1,1
UGCI    MACRO   A
        EAX     A(1)
        BRM     GCU
        ENDM


*BRS 3:  A KLUGE TO INTERFACE TWIXT THE STRING PACKAGE AND
*THE FILE HANDLING.  TAKES A STRING POINTER IN A-B, AND RETURNS
*THE FIRST 9 CHARACTERS OF THE STRING IN A-B-X, LEFT JUSTIFIED
*AND FILLED OUT WITH BLANKS IF THERE ARE LESS THAN 9 CHARACTERS

$SPFDC STA SPFDC1; STB SPFDC2; ADD =9; SKG SPFDC2; STA SPFDC2; LDX =-3
SPFDC3 CLA; STA SSR,2; STX SPFDC4; LDX =-3
SPFDC5 STX SPFDC6; UGCI SPFDC1; LDA =200B; LDB =200B
       XMA SPFDC2; SKG SPFDC1; CLB; XMA SPFDC2
       COPY AX,BX,XA; COPY AX,XA,B; EOR =200B; LDX SPFDC4
       XMA SSR,2; LSH 8; ADM SSR,2; LDX SPFDC6; BRX SPFDC5
       LDX SPFDC4; BRX SPFDC3; LDA SS01; LDB SS02
       LDX SS03; BRU EPOPX
SPFDC1 ZRO
SPFDC2 ZRO
SPFDC4 ZRO
SPFDC6 ZRO



* UTILITY BRS'S (33,34,35,37)


*       BRS 33

$GETSTR STB     GSIN1
        STX     GSIN3
        ETR     =40037777B
        CAX
        LDB     1,6
        SKA     =40000000B
        STB     0,6
        EAX     0,6
        STX     GSIN2
        SCIO    GSIN3
GSIN4   SWCI*   GSIN2
        SCIO    GSIN3
        SKE     GSIN1
        BRU     GSIN4
```

```
        LDA     Ø,6
        LDB     1,6
        LDX     GSIN3
        BRU     EPOPX


*       BRS 34

$OUTMSG STB     STRO3
        MUL     =3
        LSH     23
        SUB     =1
        SKN     STRO3
        BRU     TYM2                    NOT SPECIAL MODE
        LDB     =137777B
        BRU     STRO6
TYM2    CAB
        ADD     STRO3
        XAB
        BRU     STRO6


*       BRS 35

$OUTSTR ADD     =Ø                      CLEAR X\Ø
        STX     STRO3
STRO6   ADD     =Ø                      CLEAR X\Ø
        STX     STRO1
        ETR     =177777B
        STA     STRO2
        CBA
        ETR     =177777B
        STA     STRO21
STRO4   UGCI    STRO2
        BRU     STRO8
        SKN     STRO3
        BRU     STRO7
        SKE     =17B
        BRU     STRO5
STRO8   LDX     STRO1
        BRU     EPOPX
STRO5   SKE     =4
        BRU     STRO7
        LDA     =155B
        SCIO    STRO1
        LDA     =152B
STRO7   SCIO    STRO1
        BRU     STRO4



*       BRS 37
```

```
        IF -1
$GSLOOK STX     WR13
        ABC
        STA     WR7
        LCY     12
        SKG     =Ø
        LDA     UCOUT
        STA     WR1
        CLA
        LCY     12
        STA     WR2
        LDA     SSØ2
        ETR     ADMSK
        MRG     X4
        STA     WR3
        CAX
        LDA     1,6
        SUB     Ø,6
        STA     WR5

        LDX     WR13            PICK UP HASH TABLE BOUNDS
        LDA     Ø,6
        LDB     1,6
        STB     WR6
        CAX
        BRU     *+3

Z37B19  EAX     3,2
        CXA
        SKE     WR6                                          ND OF
        BRU     Z37B18           NO

        SKN     WR7                                           
        BRU     Z37B14           NO

        LDX     WR1Ø                                          
        LDA     1,6
        ETR     =177777B
        CAB
        LDA     WR9
        STA     WR11
        STB     WR12
Z37B2Ø  UGCI    WR11
        BRU     Z37B15
        SCIO    WR1
        BRU     Z37B2Ø
Z37B15  LDX     WR1Ø                                         XIT 
        LDB     2,6
        CXA
        MIN     SBRSRT                                        
        BRU     Z37B22
Z37B14  LDX     WR3             EXIT VIA HERE IF NO MATCH FOUND
        LDA     Ø,6
```

```
        LDB     1,6
Z37B22  LDX     WR13
        BRU     EPOPX

Z37B16  CLA
        STA     WR7
Z37B18  LDA     0,6
        SKG     =0              HASH TABLE ENTRY EMPTY ?
        SKG     =-2
        BRU     *+2
        BRU     Z37B19          YES, TRY NEXT ONE

        LDA     1,6             NO, IS HASHT STR. SHORTER THAN GIVEN STRING
        SUB     0,6
        ETR     =177777B
        ADD     =1
        SKG     WR5
        BRU     Z37B19          YES, MATCH IMPOSS

        LDA     0,6             YES
        ETR     =177777B
        CAB
        ADD     WR5
        XAB
        SSKSE*  WR3             COMPARE PROPER INITIAL"PARTS
        BRU     Z37B19          NO MATCH

        SKN     WR7
        BRU     Z37B17          YES
        SCIO    WR2             NO, APPEND CHARS UNTIL ONE ELIMINATED
        SWCI*   WR3
        MIN     WR9
        MIN     WR5
        LDA     WR8
        LDB     WR9
        SSKSE*  WR3
        BRU     Z37B16          PREVIOUS MATCH ELIMINATED
        BRU     Z37B18

Z37B17  STA     WR8
        STB     WR9
        STX     WR10
        LDA     =40000000B      RESET 'FIRST MATCH'
        STA     WR7
        BRU     Z37B19

        ENDF
```

```
*WRITE CHARACTER AND INCREMENT. THE POP ADDRESSES A STRING POINTER, AND
*THE CHARACTER IN A IS WRITTEN ONTO THE END OF THE SPECIFIED STRING.
*THE SECOND POINTER IS INCREASED BY ONE, SO THAT THE POINTER NOW
*POINTS TO THE NEW STRING. X AND A ARE PRESERVED, B DESTROYED.
WCI     POPD    15700000B,1,1,0,1
        ETR     =377B
        STA     WCH1            SAVE CHARACTER TO BE WRITTEN
        STB     WCIB
        STX     WCH2            SAVE X REGISTER
        EAX*    0               GET EFFECTIVE ADDRESS OF POP
        MIN     1,6             INCREMENT SECOND WORD OF STRING POINTER
        LDA     1,6             AND PICK IT UP.
        BRU     WCH5            GO TO WCH CODE, WHICH IS IDENTICAL FROM
*WRITE CHARACTER AND DECREMENT: DECR. 1ST PTR.
*AND WRITE CHAR. ON BEGINNING OF STRING
WCD     POPD    13500000B,1,1,0,1
        ETR     =377B
        STA     WCH1
        STB     WCIB
        STX     WCH2
        EAX*    0
        LDA 0,6; SKR 0,6; NOP
        BRU     WCH5
*                                THIS POINT ON.
*STORE POINTER. THE A AND B REGISTERS ARE STORED IN THE WORD ADDRESSED
*BY THE POP AND THE NEXT WORD
STP     POPD    16700000B,1,1,0,1
        STX     STP1            SAVE INDEX REGISTER
        EAX*    0               GET EFFECTIVE ADDRESS
        STA     0,6             DO THE
        STB     1,6             STORE
        LDX     STP1            RESTORE X
        BRR     0               RETURN
*LOAD POINTER. THE A AND B REGISTERS ARE LOADED FROM THE WORD ADDRESSED BE
*ANF THE NEXT WORD.
LDP     POPD    16600000B,1,1,0,1
        STX     STP1
        EAX*    0
        LDA     0,6
        LDB     1,6
        LDX     STP1
        BRR     0


*       GET CHARACTER AND DECREMENT
*   TRANSFER CHAR. FROM OPERAND STRING TO A REGISTER. DECREMENT END-STR. P

GCD     POPD    13700000B,1,1,0,1
        STB     WCIB
        STX     GCI1
        EAX*    0
        LDA     1,6
        SKG     0,6
```

```
        BRU     GCI3
        SKR     1,6
        BRU     GCI8

*GET CHARACTER AND INCREMENT. THE POP ADDRESSES A STRING POINTER. IF THE
*FIRST WORD OF THE POINTER IS+=OR= TO THE SECOND, IT RETURNS WITHOUT
*       SKIPPING. OTHERWISE, IT SKIPS AND RETURNS WITH THE FIRST
*       CHARACTER OF THE STRING IN A AFTER INCRIMENTING THE FIRST POINTER
*       BY 1. X IS PRESERVED, B DESTROYED.
GCI     POPD    16500000B,1,1,0,1
        STB     WCIB
        STX     GCI1                    SAVE X
        EAX*    0                       YES. DON'T HAVE TO WORRY ABOUT WHETHER
*                                       ADDRESS IS RELABLED. GET IT.
GCI9    MIN     0,6                     INCREMENT POINTER
        LDA     0,6                     AND GET IT
        SKG     1,6                     IS STRING NULL (1ST POINTER +== END)
        BRU     GCI8                    NO
        SKR     0,6                     YES. RESTORE POINTER
        NOP     0                       ALLOW FOR POSSIBLE SKIP
GCI3    LDX     GCI1                    RESTORE X
        BRR     0                       AND RETURN WITHOUT SKIPPING
GCI8    MIN     0                       INCREMENT RETURN ADDRESS FOR SKIP ON RETURN
        MUL     =12525253B              MULTIPLY CHARACTER ADDRESS BY 1/3. THIS
*                                       LEAVES WORD ADDRESS IN A, CHARACTER
*                                       ADDRESS IN TOP TWO BITS OF B.
        RCH     412B                    CAX + BAC. WORD ADDRESS TO X
        LCY     2                       MOVE CHARACTER ADDRESS TO BOTTOM OF B
        LDA     0,6                     GET WORD
        RCH     24B                     CBX + CAB
        EXU     GCI2,2                  SHIFT TO GET CHARACTER IN BOTTOM OF A
        ETR     =377B                   REMOVE SUPERFLUOUS BITS
        LDB     WCIB
        LDX     GCI1                    RESTORE X
        BRR     0                       AND RETURN
GCI2    LCY     8                       SHIFT TABLE FOR EXTRACTING A CHARACTER
        LCY     16                      FROM A WORD
        CBA
        BRU     TRAP                    ILLEGAL STRING POINTER
* GCI WITH POINTER IN T.S. BLOCK
GCU     ZRO
        LDA     GCU
        STA     0
        MIN     0,2
        LDA     0,2
        SKG     1,2
        BRU     GCI8
        SKR     0,2
        NOP
        BRR     GCU
*WRITE CHARACTER ON THE END OF STRING STORAGE. THIS IS THE ONLY POP WHICH
*       CAN CAUSE A GARBAGE COLLECTION. IT EXPECTS THE ADDRESS OF THE
*       PARAMETER TABLE IN X AND THE CHARACTER TO BE WRITTEN IN A AND
*       RESTORES BOTH. B IS DESTROYED.
```

```
WCH     POPD    16400000B,1,1,0,1
        ETR     =377B
        STA     WCH1            SAVE CHARACTER
        STB     WCIB
        STX     WCH2            AND INDEX (ADDRESS OF PARAMETER TABLE)
        EAX*    0
        MIN     0,6             INCREMENT CURRENT LAST CHARACTER TO GET
*                                CHARACTER ADDRESS FOR THE NEXT CHARACTER
        LDA     0,6             GET THE ADDRESS TO A
        SKE     1,6             IS IT EQUAL TO LAST ADDRESS IN STRING
*                               STORAGE$$
        BRU     WCH5
        LDA     WCH1
        LDB     0
        BRU     2,6
WCH5    MUL     =12525253B      COMPUTE WORD AND CHARACTER ADDRESSES
*                                  SEE GCI CODE
        CAX     SAVE            WORD ADDRESS
        LCY     5               CHARACTER ADDRESS SHIFTED 3 IN AC
        ETR     =30B            REMOVE EXTRANEOUS BITS
        LDB     0,6             PICK UP WORD
        XXA     SHIFT           COUNT TO X
        LCY     8,2             SHIFT IT TO PUT CHARACTER TO BE ALTERED
*                                  AT BOTTOM
        ETR     =77777400B      MASK OUT THIS CHARACTER
        MRG     WCH1            AND INSERT THE NEW ONE
        RCY     8,2             SHIFT WORD BACK
        CAX     RESTORE         WORD ADDRESS
        STB     0,6             STORE WORD
        LDA     WCH1            RESTORE A AND B
        LDB     WCIB
        LDX     WCH2            AND X
        BRR     0               AND RETURN
*SKIP ON STRING EQUAL. THIS POP COMPARES THE STRINGS WITH POINTERS IN
*       AB AND IN THE TWO WORDS ADDRESSED BY THE POP. IT IS
*       EXACTLY IDENTICAL TO THE MACHINE COMMAND SKE. ALL REGISTERS ARE
*       PRESERVED. THIS POP CHECKS FOR EQUAL LENGTH STRINGS FIRST
*       RETURNING WITHOUT A SKIP IF THE STRINGS ARE NOT EQUAL. IF THEY
*       ARE, IT CALLS SKCO TO FIND OUT ABOUT EQUALITY OF THIER CONTENTS
SKSE    POPD    16300000B,1,1,0,1
        STA     SKS1
        STB     SKS1+1
        STA     SKS5            SAVE TWO COPIES
        STB     SKS5+1          OF FIRST STRING POINTER
        STX     SKS2            SAVE X
        SUB     SKS1+1          COMPUTE
        STA     SKS6            AND SAVE LENGTH
        EAX*    0               GET ADDRESS OF SECOND STRING POINTER
        LDA     0,6             COMPUTE
        SUB     1,6             ITS LENGTH
        SKE     SKS6            AND COMPARE WITH FIRST
        BRU     *+2             NOT EQUAL
        BRU     SKSE2           EQUAL
        LDX     SKS2            NOT EQUAL. RESTORE X,
```

```
        LDA     SKS5            A AND
        LDB     SKS5+1          B
        BRR     0               AND RETURN WITHOUT SKIPPING
SKSE2   LDB     1,6             LENGTH EQUAL. PICK UP SECOND
        LDA     0,6             STRING POINTER
        LDX     SKS2            AND ORIGINAL X
        BRM     SKCO            COMPARE STRING IN AB WITH THE ONE IN SKS1
        BRU     *+2             AB LESS
        MIN     0               EQUAL. INCREMENT RETURN ADDRESS FOR SKIP
SKSE1   LDA     SKS5            GREATER. RESTORE A
        LDB     SKS5+1          AND B
        BRR     0               AND RETURN
*SKIP ON STRING GREATER. THIS POP IS TO SKG AS SKSE IS TO SKE. IT CALLS
*       SKCO IMMEDIATELY WITHOUT COMPUTING LENGTHS
SKSG    POPD    16200000B,1,1,0,1
        STA     SKS1            SAVE
        STB     SKS1+1          TWO COPIES
        STA     SKS5            OF FIRST STRING.
        STB     SKS5+1
        STX     SKS6            SAVE X
        EAX*    0               GET
        LDA     0,6             SECOND STRING
        LDB     1,6             POINTER
        LDX     SKS6            RESTORE X
        BRM     SKCO            DO COMPARISON
        MIN     0               SECOND STRING LESS, I.E. FIRST GREATER($.)
*                                  INCREMENT RETURN ADDRESS FOR SKIP
        NOP     0               EQUAL
        LDA     SKS5            FIRST LESS. RESTORE A
        LDB     SKS5+1          AND B
        BRR     0               AND RETURN
*THIS ROUTINE COMPARES THE STRINGS IN AB (FIRST STRING) AND IN SKS1
*       (SECOND STRING). IT RETURNS WITHOUT SKIPPING IF THE FIRST IS
*       LESS, SKIPS ONCE IF THEY ARE EQUAL, AND SKIPS TWICE IF THE FIRST
*       IS GREATER. NOTE THAT SEVERAL OF THE NECESSARY GCI'S ARE WRITTEN
*       OUT INSTEAD OF BEING CALLS ON THE POP.
SKCO    ZRO     0
        STA     SKS4            SAVE FIRST
        STB     SKS4+1          STRING POINTER
        STX     SKS7            AND X
SKCO1   MIN     SKS1            BEGIN LOOP. GET A CHARACTER
        LDA     SKS1            FROM SECOND STRING
        SKG     SKS1+1          (IS IT NULL$$
        BRU     SKCO4           NO
SKCO2   MIN     SKCO            YES.) FIRST STRING CANNOT BE LESS, SO
*                                  INCREMENT RETURN ADDRESS
        MIN     SKS4            IS FIRST STRING NULL
        LDA     SKS4
        SKG     SKS4+1
        MIN     SKCO            NO. THEREFORE IT IS GREATER
        BRU     SKCO5           RETURN
SKCO4   MUL     =12525253B      SECOND STRING NOT NULL. GET CHARACTER
        RCH     401B            AS
        LRSH    22              IN
```

```
        LDA     0,6             GCI
        RCH     24B             CODE
        EXU     GCI2,2          ABOVE
        ETR     =377B
        STA     SKS3            GOT IT
        MIN     SKS4            TRY TO GET CHARACTER
        LDA     SKS4            FROM FIRST STRING
        SKG     SKS4+1          IS IT NULL$$
        BRU     *+2             NO
        BRU     SKCO5           YES SO IT MUST BE LESS. RETURN WITHOUT
*                                  SKIPPING
        MUL     =12525253B      NOT NULL. GET
        RCH     401B            CHARACTER
        LRSH    22              AS
        LDA     0,6             IN
        RCH     24B             GCI
        EXU     GCI2,2          CODE
        ETR     =377B           ABOVE
        SKE     SKS3            AND COMPARE WITH CHARACTER OF 2ND STRING
        BRU     SKCO3           NOT UAL
        BRU     SKCO1           EQUAL. LOOP
SKCO3   SKG     SKS3            COMPARE FURTHER
        BRU     SKCO5           LESS. RETURN WITOUT SKIPPING
        MIN     SKCO            GREATER SKIP
        MIN     SKCO            TWICE
SKCO5   LDX     SKS7            RETURN POINT. RESTORE X
        BRR     SKCO            AND RETUR
* THIS ROUTINE, CALLED WITH BRS 5, SEARCHES A HASH TABLE FOR AN
*       APPROPRIATE ENTRY FOR THE STRING IN AB. IT SKIPS IF THE STRING
*       IS ALREADY IN THE TABLE. OTHERWISE IT SAVES THE INDEX OF THE
*       PARAMETER TABLE AND DOES NOT SKIP. X IS PRESERVED. A AND B ARE
*       PRESERVED IF THERE IS NO SKIP, BUT B IS THE INDEX IN THE HASH
*       TABLE AND A THE VALUE (THIRD WORD) IF A MATCHING ENTRY IS FOUND
        IF -1
$SSCH   STA     SCH1            PUT STRING
        STB     SCH1+1          POINTER AWAY
        STA     SCH5            (TWO COPIES)
        STB     SCH5+1
        CBA
        SUB     SCH5            LENGTH OF STRING
        STA     SCH9            AND SAVE IT
*                                  COMPUTE HASH TABLE INDEX.
        RCH     402B            CAX + CLB
        SKG     =3              FIGURE OUT MASKS FOR FIRST 3 AND LAST 3
*                                  CHARACTERS
        BRU     SCH12           IF STRING IS LESS THAN 3 CHARACTERS LONG
        LDA     =77777777B      NOT LESS. THE MASK
        STA     SCH10           IS ALL
        STA     SCH11           ONES
        BRU     SCH13
SCH12   LDA     SCH14,2         LESS. GET THE FIRST
        STA     SCH10           CHARACTERS MASK
        LDA     SCH15,2         AND THE LAST
        STA     SCH11           CHARACTER'S MASK
```

```
SCH13   CXA     RECOVER         LENGTH
        LSH     3               MULTIPLY BY 8
        STA     SCH4            AND USE IT TO START THE HASH CODE
        LDA     SCH5
        ADD     =1
        MUL     =12525253B      COMPUTE WORD ADDRESS FOR FIRST CHARACTER
*                                  OF STRING
        RCH     401B            CAX + CLA. WORD ADDRESS TO X
        LRSH    22              CHARACTER ADDRESS TO B
        LDA     0,6             GET FIRST WORD OF STRING
        LDX     1,6
        XXB     GET             SECOND WORD
        EXU     SCH16,2         SHIFT SO THAT FIRST 3 CHARACTERS ARE IN A
        ETR     SCH10           GOT FIRST 3 CHARACTERS
        LRSH    12              FOLD THIS
        ADM     SCH4            WORD IN HALF, ADDING THE
        CLA     HALVES
        LSH     12              TO THE HASH
        ADM     SCH4            CODE BEING ACCUMULATED
        LDA     SCH5+1          REPEAT
        MUL     =12525253B      THE
        RCH     401B            ABOVE
        LRSH    22              FOR
        LDA     0,6             THE
        LDX     -1,6            LAST
        XXB     THREE
        EXU     SCH17,2         CHARACTERS
        ETR     SCH11           OF
        LRSH    12              THE
        ADM     SCH4            STRING
        CLA
        LSH     12
        ADM     SCH4
        LDX     SS03            RECOVER ADDRESS OF CONTROL TABLE
        LDA     1,6             COMPUTE LENGTH
        SUB     0,6             OF HASH TABLE
        XMA     SCH4            AND GET
        LRSH    23              HASH CODE
        DIV     SCH4            MOD
        CLA
        LSH     1               THIS LENGTH,
        DIV     =3              AND AN
        MUL     =3              EVEN MULTIPLE
        LSH     23              OF 3
        ADD     0,6             TURN IT INTO A HASH TABLE ADDRESS
        CAX
        STA     SCH20
        LDA     =-1             SET THE 'DID WE PASS OVER A -1 (DELETED)
*                                  ENTRY' FLAG
        STA     SCH52           TO 'NO'9
SCH50   LDA     0,6             BEGIN LOOP TO SCAN HASH TABLE9 NOTE
*                                  THAT THE SCAN GOES BACKWARDS AND IS
*                                  CYCLIC
        CXB     SAVE            INDEX
```

```
        LDX     SS03
        SKE     =-1             IS THIS A -1 (DELEATED) ENTRY
        BRU     *+2             NO
        BRU     SCH51           YES
        SKE     =0              NO. IS IT EMPTY
        BRU     SCH8            NO.
SCH18   STB     2,6
        LDA     SCH52           DID WE PASS A -1 ENTRY
        SKN     SCH52           BEFORE FINDING THIS 0 ENTRY
        STA     2,6             YES. LEAVE THE INDEX OF THE
*                                       -1 ENTRY AS THE EMPTY SLOT, SINCE
*                                       THE NEAREST AVAILABLE SLOT TO THE
*                                       ED ADDRESS FOR THE STRING
        LDA     SCH1            RESTORE
        LDB     SCH1+1          A AND B
        BRU     XPOP            AND EXIT THROUGH A SYSTEM ROUTINE WHICH
*                                       CHECKS FOR TIME OVERFLOW
SCH8    ETR     =177777B        ENTRY NOT EMPTY. MASK OUT TOP 8 BITS
        STA     SKS1            AND STORE FIRST WORD OF ITS STRING POINTER
        CBX
        LDA     1,6             DO THE SAME
         ETR     =177777B        FOR THE
        STA     SKS1+1          SECOND WORD
        SUB     SKS1            COMPUTE LENGTH
        SKE     SCH9            DOES IT MATCH LENGTH OF STRING BEING
*                                       SEARCHED FOR
        BRU     SCH7            NO. GO ON TO NEXT ENTRY
        LDA     SCH1            YES
        LDB     SCH1+1          CALL SKCO
        BRM     SKCO            FOR COMPARISON OF STRINGS
        BRU     SCH7            NO MATCH
        BRU     *+2             MATCH
        BRU     SCH7            NO MATCH
        MIN     0               AND INCREMENT FOR SKIP
        CXA     PUT             INDEX
        ETR     =77777B         INTO
        CAB     B
        LDA     2,6             AND VALUE INTO A
        LDX     SS03            RESTORE X
        BRU     XPOP            AND EXIT
SCH7A   CBX
SCH7    CXA     ENTRY           DOES NOT MATCH. GO ON TO NEXT ONE.
*                                       INDEX OF CURRENT ENTRY TO A
        LDX     SS03            GET ADDRESS OF CONTROL TABLE
        ETR     =777777B
        SKE     0,6             HAVE WE REACHED THE BEGINNING OF THE
*                                       TABLE IN OUR BACKWARDS SCAN
        BRU     *+2             NO
        LDA     1,6             YES. CYCLE TO END
        SUB     =3              MOVE DOWN TO THE NEXT ENTRY (EACH
*                                       ENTRY IS 3 WORDS)
        CAX
        SKE     SCH20           HAVE WE MADE A FULL CIRCLE
        BRU     SCH50           AND LOOP
```

```
        LDX     SSØ3
        LDB     =-1
        BRU     SCH18           GIVE UP
SCH51   SKN     SCH52           FOUND A -1 ENTRY. IS THE FIRST ONE
        BRU     SCH7A           NO. DO NOTHING
        STB     SCH52           YES. SET FLAG POSITIVE ADD TO ADDRESS OF
*                                       ENTRY FOUND
        BRU     SCH7A
SCH14   DATA    Ø,77600000B,77777400B MASKS FOR FIRST 3 CHARACTERS OF STRIN
*                                       NOT 3 CHARACTERS LONG
SCH15   DATA    Ø,377B,177777B  MASKS FOR LAST 3 CHARACTERS IF STRING IS
*                                       NOT 3 CHARACTERS LONG
SCH16   NOP     Ø               SHIFT INSTRUCTIONS TO GET FIRST 3
*                                       CHARACTERS OF STRING IN A REGARDE
*                                       CHARACTER ADDRESS
        LCY     8               OF THE FIRST CHARACTER
        LCY     16
SCH17   RCY     16              SAME FOR LAST 3 CHARACTERS
        RCY     8
        NOP     Ø
* BRS 6 INSERTS THE MISSING STRING
$SSIN   LDX     SSØ3
        LDA     2,6
        SKA     =40000000B      WAS TABLE FULL ?
        BRU     TRAP            YES, CAN'T FIT NEW ENTRY
        CAX     NO
        LDA     SSØ1
        STA     Ø,6
        STB     1,6
        CXB
        LDA     2,6
        LDX     SSØ3
        BRU     XPOP

        ENDF
STP1    ZRO     Ø
GCI1    ZRO     Ø
GCI7    ZRO     Ø
WCH1    ZRO     Ø
WCH2    ZRO     Ø
WCIB    ZRO     Ø
SKS1    BSS     2
SKS2    ZRO     Ø
SKS3    ZRO     Ø
SKS4    BSS     2
SKS5    BSS     2
SKS6    ZRO     Ø
SKS7    ZRO     Ø
SCH1    BSS     2
SCH5    BSS     2
SCH4    ZRO     Ø
SCH9    ZRO     Ø
SCH1Ø   ZRO     Ø               MASK FOR FIRST 3 CHARACTERS OF STRING
SCH11   ZRO     Ø               MASK FOR LAST 3 CHARACTERS OF STRING
```

```
SCH20   ZRO     Ø               STARTING LOCATION FOR SCAN OF HAST TABLE
SCH52   ZRO     Ø               FLAG IS -1 IF NO -1 ENTRY HAS BEEN
*                                  ENCOUNTERED DURING SCH SCAN, THE INDEX
*                                  OF THE FIRST SUCH ENTRY ENCOUNTERED
*                                  OTHEWISE



* FLOATING POINT ARITHMETIC


* FAD   FLOATING ADD
FAD     POPD    15600000B,1,1,Ø,1
$FADE   STA     SSØ1
        CLA
FASE    STX     SSØ3
        STA     FLAG
        STB     ZM
        STE
        STX     ZE
        LDX     SSØ3
        EAX*    Ø
        CLA
        SKE     Ø,6
        BRU     FAN
        CLAB
        BRU     FLAD
FAN     SKE     SSØ1
        BRU     FAX
        CLAB
        BRU     FLAC
FAX     CXA
        LDB     1,6
        STE
        XXA
        SUB     ZE
        SKG     =-1
        BRU     FLAGM
        SKA     =-100B
        LDA     =39
        XMA     SSØ1
        LDB     ZM
        RSH*    SSØ1
        XAB
FLAC    SKN     FLAG
        BRU     FAS
        MRG     =777B
        SUB     1,6
        EOR     =777B
        XAB
        SUC     Ø,6
        BRU     FLAF
```

```
FAS     COPY    A,E
        ADD     1,6
        XAB
        ADC     0,6
FLAF    STE
FLAOT   OVT
        BRU     FLAOS
        NOD     38
        SKA     =-1
        BRU     FLANZ
        CLX
FLANZ   XXA
        SKG     =-257
FLAOF   BRR     FLAOA           SET OVERFLOW DEVIOUSLY
FLAOK   XXA
FLAX    OVT
        BRU     FLOF
FLAX1   LDE
        LDX     SS03
        BRU     XPOP
FLAOA   DATA    10000000B+FLAOK-1
FLAGM   CNA
        SKA     =-100B
        LDA     =39
        LDX     0,6
        XXA
        RSH     0,2
        COPY    B,E
FLAD    SKN     FLAG
        BRU     FLSS
        XMA     SS01
        XAB
        XMA     ZM
        SUB     ZM
        XAB
        SUC     SS01
        BRU     FLAF
FLSS    XAB
        ADD     ZM
        XAB
        ADC     SS01
        BRU     FLAF
FLAOS   RSH     1
        EOR     =40000000B
        BRX     FLAX
        XXA
        SKG     =255
        BRU     FLAOK
        BRR     FLAOA           SET OVERFLOW DEVIOUSLY
FLOF    BRX     *+2
        BRU     FLOJ
        CLEAR
        LDX     SS03
        BRU     XPOP
```

```
FLOJ    EOR     =40000000B
        RSH     39
        EOR     =40000000B
        EAX     255
        BRR     FLOJ1
FLOJ1   ZRO     FLAX1-1,1
* FSB  FLOATING SUBTRACT
FSB     POPD    15500000B,1,1,0,1
$FSBE   STA     SS01
        LDA     =-1
        BRU     FASE
* FNA FLOATING CNA (BRS 21)
$FNA    LDX     SS03
        BRM     FNAS
        BRU     XPOP
FNAS    ZRO
        STX     FNASX
        SKB     =-1000B
        BRU     FLNA
        CNA
        SKE     =0
        SKA     =17777777B
        BRR     FNAS
        STE
        SKE     =40000000B
        BRU     FLCOM
        RCY     1
        BRX     FLCA
FLCOM   NOD     4
FLCA    XXA
        SKG     =255
        SKG     =-257
        BRR     FNAOFL
        BRU     FLNB1
FNAOFL  ZRO     FNAOF2-1,1
FNAOF2  XXA
        EOR     =40000000B
        RSH     39
        EOR     =40000000B
        BRX     *+3
        EAX     255
        BRU     FLNB
        CLEAR
        BRU     FLNB
FLNB1   XXA
        ROV
FLNB    LDE
FLMX    LDX     FNASX
        BRR     FNAS
FLNA    STE
        XAB
        CNA
        XAB
        EOR     =-1
```

```
        BRU     FLNB
* ISC   INTERNAL TO STRING CONVERSION
ISC2    NOP     1000B
ISC     POPD    14000000B,1,1,0,1
$ISC1   STA     SS01
        LDA     ISC2
        BRU     SC1
* SIC   STRING TO INTERNAL CONVERSION
SIC2    NOP     1001B
SIC     POPD    14100000B,1,1,0,1
$SIC1   STA     SS01
        LDA     SIC2
SC1     STA     BSX
        STB     SS02
        STX     SS03
        EAX*    0
        CXA
        ETR     ADMSK
        STA     UBE
        BRU     BSE
* FMP   FLOATING MULTIPLY
FMP     POPD    15400000B,1,1,0,1
$FMPE   STX     SS03
        STA     SS01
        STE
        STX     ZE
        BAC
        LDX     SS03
        EAX*    0
        LRSH    2
        MUL     0,6
        STA     ZM
        LDA     1,6
        CXB
        COPY    AX,A,E
        XXA
        ADM     ZE
        COPY    XA,BX,B
        LRSH    2
        MUL     SS01
        ADD     ZM
        MUL     =2
        STB     ZM
        XMA     SS01
        MUL     0,6
        XAB
        ADD     ZM
        XAB
        ADC     SS01
        LDX     ZE
        SKA     =37777777B
        BRU     FLEND
        SKB     =-1000B
        BRU     FLEND
```

```
        SKE     =40000000B
        BRU     FLND2
        RCY     1
        BRX     FLEND
FLEND   NOD     4
        ROV
        XXA
        SKG     =255
        SKG     =-257
        BRR     FLA0A
        XXA
        LDE
FLND2   LDX     SS03
        BRU     XPOP

* FDV   FLOATING DIVIDE
FDV     POPD    15300000B,1,1,0,1
$FDVE   STX     SS03
        STA     SS01
        STE
        STX     ZE
        LDX     SS03
        EAX*    0
        RSH     2
        DIV     0,6
        OVT
        BRU     FDV0
        STA     ZM
        BAC
        RSH     1
        STA     SS01
        LDB     1,6
        CXA
        STE
        XXA
        CNA
        ADD     =2
        ADM     ZE
        BAC
        RCY     2
        CNA
        MUL     ZM
        ADD     SS01
        DIV     0,6
        MUL     =2
        ADD     ZM
        LDX     ZE
        SKA     =-1
        BRU     FLEND
        BRU     FLND2
FDV0    LDA     SS01
        EOR     0,6
        BRU     FLOJ
* STORAGE
```

```
ZE      ZRO
ZM      ZRO
FLAG    ZRO
FNASX   ZRO

* PRINT UNSIGNED INTEGER
*
$OUTNUM STA     OUTNO2
        STB     OUTNO3
        STX     OUTNO5
        CBA
        SKG     =1
        BRU     TRAP
        CLA
OUTNO6  STA     OUTNO4
        LDA     OUTNO2
OUTNO7  STA     OUTNO1
        LRSH    23
        DIV     OUTNO3
        SKE     OUTNO4
        BRU     OUTNO7
        CBA
        ADD     =20B
        SCIO    OUTNO5
        LDA     OUTNO1
        SKE     OUTNO2
        BRU     OUTNO6
        LDB     OUTNO3
        LDX     OUTNO5
        BRU     EPOPX
*
* DEMAND SIGNED INTEGER
*
$GETNUM STX     GETNO1
        BAC
        STA     GETNO2
        STB     GETNO3
        STB     GETNO7
        LDA     =-1
        STA     GETNO8
        SCIO    GETNO1
        SKE     =13B
        BRU     *+2
        BRU     GETNO6
        SKE     =15B
        BRU     GETNO6+1
        LDA     =-1
        STA     GETNO7
GETNO6  SCIO    GETNO1
        SKN     GETNO8
        BRU     *+4
        SKG     =0
        BRU     GETNO6
        STA     GETNO8
```

```
        SUB     =20B
        SKG     =-1
        BRU     GETNO4
        SKG     GETNO2
        BRU     GETNO5
GETNO4  ADD     =20B
        CAB
        LDA     GETNO3
        SKN     GETNO7
        BRU     EPOPX
        CNA
        BRU     EPOPX
GETNO5  SKE     GETNO2
        BRU     *+2
        BRU     GETNO4
        XMA     GETNO3
        MUL     GETNO2
        LSH     23
        ADM     GETNO3
        BRU     GETNO6
*
* FIX AND FLOAT
*
$FFIX   SKA     X4
        BRM     FNAS
        SKD     =23
        BRU     FIXBIG
        COPY    B,E
        RSH     0,2
FIXEND  SKN     SS01
        BRU     FEND1
        CNA
FEND1   LDX     SS03
        BRU     XPOP
FIXBIG  COPY    B,E
        LSH     0,2
        ETR     XX
        BRU     FIXEND
*
$FFLT   CLB
        SKE     =0
        BRU     *+2
        BRU *+4    CHANGED 4/8/69  T.ENGLAND
        LDX     =23
        NOD     48
        LDE
        LDX     SS03
        BRU     XPOP
        END
```