```
CORE 16 SUBROUTINES 3/15/73 JBL/BPC (SUBRS,63)

/P POINTERS
BITS=Ø
BILLTT=6
SUTD=7
PROGNM=11
DDTSEG=16
RESTPC=32
DDTBKP=2Ø
HH16PC=41
HH16IO=42
USERAC=44
USERPC=45
USERIO=46
USERFG=47
C16TEM=55
DDTS6=65
IOPMAX=74
TISMAX=75
TTNO=76
TRAPPC=77
ERCODE=1Ø2
OWNWD=1Ø4
TTTSU=1Ø5
IOPTSU=1Ø6

/TEM STORAGE IN CORE Ø
FSA=1Ø7
STS=11Ø
JMODE=111
ATEM=112
FAC=123
FOV=125
BTEM=127  /ALSO REFERENCED AS CBTM3 (BTEM+3)
DTEM=15Ø
ETEM=373
```

```
SZS=640000
DELAY=IOT 1600
EOTHNG=IOT 101
TYI=IOT 201
TYO=IOT 301
TIS=IOT 401
TOS=IOT 501
GTY=IOT 600
CTTS=IOT 1001
ADDLP=IOT 1300
WPP=IOT 3200
GTD=IOT 3400
PEEK=IOT 3700
C16RET=IOT 4000
RPA=IOT 4200
PPA=IOT 4500
RCK=IOT 4700
RPPU=IOT 5501
RPPB=IOT 5541
EXPP=IOT 5741
BEDTR=IOT 5701
RLSQTK=IOT 6041
RAI=IOT 6101
RAB=IOT 6141
EXAI=IOT 6201
EXAB=IOT 6241
WNBF=IOT 6341
WNI=IOT 6301
WNIH=IOT 6321
WAI=IOT 6401
WAIH=IOT 6401
WAIA=IOT 6411
IVNR=IOT 6500
WNIA=IOT 6701
HALT.=IOT 12401

/TEM STORAGE IN CORE 16
A=0
B=1
C=2
D=3
E=4
F=5
G=6
H=7
```

```
/EXECUTIVE FUNCTIONS

160010/   C16RET+1        /ILLEGAL

PSTART,   LIO C6
          JMP START1
STARTUP,  LIO C14         /HALT IF OVF SET
          SZO I
          LIO (15)        /STARTUP
START1,   DZM I (IOPMAX)
          LAW 36
          IVNR 23
          JMP .
          C16RET          /DEBREAK VIA C16PC IN ITEM READ

HHORR,    LAW DDTSEG
          WPP             /SAVE TRPBUF
          GTD+1
          LAW DDTBKP+7
          WPP             /SAVE TIME AND DATE OF CRASH IN ...
          GTD+1           /... LOWER CORE
          LIA
          LAW DDTBKP+6
          WPP
          CLA
          WNBF+2          /WRITE OUT FIRST 50. WORDS AS A BLOCK
           JMP .
          LAW 37          /PUT DRA OF BLOCK IN REWRITE NUMBER ...
          WPP             /... OF ITEM TO BE WRITTEN
          LAW 36
          LIO (I-36)      /LENGTH
          WPP
          WNI+2
          JMP .
          LAW DDTBKP
          WPP
          LAW DDTSEG
          PEEK
          C16RET+3        /SOROBAN TYPEOUT
          LAC (HHORRT)
          TOS
          HLT
          HALT.

HHORRT:   TEXT /
PROGRAM CANNOT CONTINUE.#/
```

/START UP A NEW USER PROGRAM.

```
SUPGO,    LIO I (USERAC)
          LAW RESTPC
          WPP
          LIO I (USERIO)
          LAW 36
          DZM I (IOPMAX)
          RAI+2
          JMP .
          DIO I (USERIO)
          LAW RESTPC
          PEEK
          DAC I (USERAC)
          DZM I (DDTS6+1)
          SPA
          LAW DDTS6+1
          DAC I (USERPC)
          LAW BILLTT
          PEEK
          DAC I (TTNO)
          C16RET
```

/GET A CLEAN FILE PACKAGE.

```
FILPKG,   DZM I (IOPMAX)
          LIO FILIVN
          LAW FILBEG          / user low core (127-637)
          IUNR+23
          JMP .
          LAC FILCAL
          DAC I C101          / user CAL instruction
          JMP R1
FILIVN,   251
FILCAL,   JMP FILBEG+15
```

```
/CORE 16 BREAKPOINT LOGIC

BKPNT,    LAC (177777)
          AND I (HH16PC)
          SAS (DDTS6+1)
          SAD (DDTS6+2)
          JMP GETDDT      /DDT IS EXECUTING AN INSTRUCTION
          RAL 2S
          SPA             /FROM CORE 16?
          JMP GETDDT      /YES, GET DDT
          LAW 7777
          AND I (HH16PC
          DAC A           /ADDRESS OF INSTRUCTION
          LAW DDTBKP
          DAC B           /ADDRESS OF BREAKPOINT(BPT)
16BPTL,   PEEK            /READ FIRST BPT INTO AC
          DAC C           /CONTENTS OF BPT REGISTER
          AND (7777)      /?
CHARLH,   SAD A           /IS THIS THE BPT?
          JMP 16BPTF      /YES
          IDX B           /NO, LOOP
          IDX B
          SAS (DDTBKP+1Ø  /NOT A BPT?
          JMP 16BPTL      /NO
          JMP GETDDT

16BPTF,   IDX B
          PEEK            /GET PROCEED COUNTER INTO AC
          LIO B
          ADD B17         /IDX PROCEED COUNTER
          SWP
          WPP             /WRITE OUT PROCEED COUNTER
          SPI I
          JMP GETDDT
          LAC I A
          XOR C           /PUT ADDRESS INTO REG.
          AND (17777)
          XOR C
          DAC I (DDTS6+3)
          LAW 1
BØ,       ADD A           /GET PC
          ADD CJ          /?
          DAC I (DDTS6+4)
          IDA
          DAC I (DDTS6+5)
          LAC BØ
          AND I (HH16PC) /OVF BIT
CIORIN,   IOR (DDTS6+3)
          DAC I (HH16PC)
          C16RET+2        /DEBREAK THRU HH16
```

```
/DDT-GO LOGIC.  DDT'S IOPMAX IS ZERO.  IOT'S OVF AND LINK CLEAR.
/REWRITES ON HELD DONE WITH "DON'T CHECK OWNWD OR REWRITE NUMBER".

DDTGO,     LAW DDTSEG      /OVF, LINK CLEAR
           PEEK+20
           LAW 36
           WAIH+22         /WRITE DDT OUT
            JMP DDTWER
           SZF 1
           JMP DDTUF       /U"F", C"F", OR L"F"
           LIO (DDTSEG+1   /READ CORE IMAGE
           PEEK+60
           RAI+2
            JMP DDTERR
           SZF 4
           JMP DDTZ        /"Z"
           SZF 5
           JMP DDTSF       /"S"F"
           SZF I 6
           C16RET+2        /REGULAR "GO"
           LIO A           /P"F".  LENGTH OF ITEM APPEARED IN A.
           WPP
           LIO I (HH16IO   /USER PUT DRA IN REG 42 OF CORE IMAGE

/SAVE NEXT 2 INSTRUCTIONS WHEN WAIH+22 WILL REWRITE ITEM ON FREE.
/AS LONG AS REWRITE NUMBER AND OWNWD REALLY DO MATCH UP.

           WAI+2           /REWRITE ITEM (PRESUMABLY ON LIBRARY)
            JMP DDTERR
           JMP DDTPFX

DDTUF,     LIO I (USERIO   /DDT'S I, O, HAS DRA OF FILE OR...
           RAI+2           /...OF STARTUP PGM.
            JMP DDTERR
           SZF 2
           JMP DDTCAL      /C"F" OR L"F"
GETDDT,    CLO             /BPT LOGIC MAY HAVE SET OVFL
BKKEY,     LAW 36          /BK KEY ENTERS HERE WITH OVF SET
           LIO (I-36
           WPP
           LIO (DDTSEG+1
           PEEK+60
           WAIH+22         /WRITE OUT CORE IMAGE
            JMP DDTERR
DDTPFX,
DDTSFX,    LAW 36          /ENTRY FROM DDTERR, DDTUF, DDTSF
           LIO (DDTSEG
           PEEK+60
           DZM I (IOPMAX
           RAI+2           /READ DDT IN
            JMP .          /CAN'T FIND DDT
           IDX I (USERPC
           SZL
           C16RET          /ERROR RETURN
           SZO
           JMP R1          /BK KEY
           JMP R2          /ILLEGAL INSTR OR BPT OR NORMAL RETURN
```

```
DDTCAL,    LAW 1ØØ          /SA OF STARTUP PGM SEG
           DAC I (HH16PC
           LIF
           DIO I C1Ø1       /C"F" VS L"F" DISTINCTION FOR SUP SEG
           C16RET+2

DDTZ,      LAW RESTPC
           PEEK+2Ø          /GET G# INTO I. O.
CLACB,
DDTZL,     LAC B            /UPPER LIMIT APPEARED IN B
           SUB A            /LOWER LIMIT IN A
           SPA
           JMP GETDDT
           DIO I A
           IDX A
           JMP DDTZL

DDTSF,     WNIH+2           /WRITE COPY OF CORE
            JMP DDTERR
           LAW DDTSEG+1
           WPP              /RETURN DRA OF SAVED FILE IN DDTSEG+1
           JMP DDTSFX

DDTERR,    LAW RESTPC       /IOP ERROR
           LIO I (ERCODE
           WPP              /SAVE ERCODE FOR DDT IN RESTPC
           CLL"U"CML
           JMP DDTSFX       /READ DDT AND GIVE R1

DDTWER,    LAW RESTPC       /ERROR TRYING TO WRITE DDT OUT
           LIO I (ERCODE
           WPP              /SAVE ERCODE WHERE DDT WILL BE LOOKING FOR IT
           JMP R1           /GIVE ERROR RETURN.  DDT STILL IN CORE.


/FASTRAND SWAPPING
C16FIN,    C16RET+4
C16FOU,    C16RET+5
C16FSW,    C16RET+6
           C16RET+1Ø
C16SWF,    C16RET+7

LISSUP:    ADDLP 2
           C16RET

16LSSP:    ADDLP 2
           LAC B
ML2,       JMP A
```

```
/USER FUNCTIONS, IOT'S

/EDIT IOT AND SUBR
EDIT,     LAW IR1          /SETUP RETURN TO GIVE RETURN TO USER
.EDIT,    LIO I (USERAC) /SUBROUTINE ENTRY, USED ONLY IN BCJBH
.EDITJ,   DAP H            /JOB HUNTER ENTRY. SET RETURN
          DIO G            /TEXT PTR FOR EDITING
          DIO F
          DIO E            /BACKSPACE STOPPER
          LCH I E          /JUST TO IDC IT
EDITL,    LCH I F
          SAD ML4          /EOM?
          JMP EDITE
          SAS ML6
          JMP EDITS        /NOT A WARNING CHAR.
          LCH I F          /GET 2ND HALF
          SAD CHARLD       /(440000) BS?
          JMP EDITB
          SAD ML4          /RO?
          JMP EDITR
          RAR 6S
          IOR ML6          /SETUP TO DCH TWICE
          DCH I G
EDITS,    DCH I G          /COPY BACK INFO
          JMP EDITL

/BACKSLASH
CLACG,
EDITB,    LAC G
          IDC
          SAD E
          JMP EDITL        /CAN'T BACK UP
          LAC G
          SMA
          SUB B1
          SUB B1           /UNSTEP
          DAC G
          LCH G
          SAS ML6
          JMP EDITL
          JMP EDITB        /PREV. CHAR, WAS A WARNING CHAR.

/EOM
EDITE,    IDX H            /SECOND EXIT
EDITR,    LAC ML4
          DCH I G          /STORE EOM
          LIO G            /RETURN PTR TO JBH IN IO
          SZF I 1          /AND TO USER IN AC EXCEPT ON EDIT+40
          DIO I (USERAC)
          JMP HEXIT        /GO EXIT THRU H
```

```
/RPB SUBROUTINE AND IOT
RPB,        LAW R1
.RPB,       DAC H           /EXIT
            LAW I 3
            DAC G           /CTR FOR RPA'S
RPB1,       RPA
            RIR 8S
            SPI I           /CHANNEL 8 BIT?
            JMP RPB1        /NO, IGNORE CHAR.
            RIL 2S
            LAC I (USERIO)
            RCL 6S
            DAC I (USERIO) /SHIFT INTO USER'S IO
            ISP G
            JMP RPB1        /3 TIMES
HEXIT,      LAC CJ
CDIPH,      DIP H
CJMPH,      JMP H           /EXIT THRU H

/ENTER READIN MODE (ERIM) IOT
ERIM,       DZM F           /USERCORE PTR. FOR DIO'ING
ERIM1,      JSP .RPB        /USER MUST HAVE READER
            LAC I (USERIO)
            SPA
            JMP GO          /GO TO LOC. SPECIFIED IN AC
            DAP F
            JSP .RPB
            LAC I (USERIO)
            DAC I F
            JMP ERIM1       /LOOP UNTIL TERMINATOR

/PPB IOT
PPB,        LAW I 3         /PUNCHES 3 LINES OF TAPE
            DAC G
            LIO I (USERIO)
PPB1,       LAW 2           /8TH HOLE
            RCL 6S
            SWP
            PPA
            SWP
            ISP G
            JMP PPB1
            JMP R1
```

```
/TTON IOT
TTON,     GTY               /IO POSITIVE
          DIO I (USERIO)
          SPI
          JMP R1            /CAN'T GET TT
          LAC (TTONT)
          TOS"U"1           /TURN ON TT AND SEND WRU'S
           JMP TTONR2
          RCK
          DAC H

TTONL,    CLA"U"CLI
          DELAY+40
          RCK
          SUB H
          SPA
          ADD C60,K
          SUB (17777)       /8 SECONDS
          SMA
          JMP R1            /WON'T TURN ON
          TYI"U"3           /TEST FOR "HERE IS"
           JMP TTONR2
          SPI
          JMP TTONL

TTONR2,   LAC (760002)      /ACTS LIKE LAW 3 WITH DELAY+40
          DELAY+40
          TYO"U"1
           NOP
          JMP R2

TTONT:    770077
          007700
          774177
          417741
          770077
          007700
          774177
          417741
          774177
          417705
          770574
```

```
/HALT IOT
HALT,     CTTS
          NOP
          DZM I (JMODE)
          LAC (760000)
          TYO
           JMP HALT1      /TRY ONCE MORE
          GTD             /OLD TIME
          DAC A
          DIO B
          GTD+1           /CURRENT TIME
          SAD A
          STF 3           /SUPPRESS DATE IF SAME
          DAC I (USERAC)
          DIO I (USERIO)
          LAC CHLT
          DAC I (TTTSU)
          JSP .STD
HALT1,    LAW BITS
          PEEK
          SPA             /RUNNING UNDER IDDT?
          JMP I (66)      /YES, SILENT RETURN
          LAC (HALTT)
          TOS
CHLT,      HLT            /GIVE UP
          LAW 37          /SEE IF SHOULD TYPE EOT
          SUB I (TTNO)
          SPA
          JMP HALT2
          SUB (37-30)
          SPQ
           HLT
HALT2,    LAC (HLTEOT)
          TOS
           JMP SICKTT
          HLT

HALTT:    771476
          767676
          767674
HLTEOT:   770474
```

```
/DECODE NUMBER IOT AND SUBROUTINE
DNM,        LAW IR1          /USER ENTRY
.DNM,       DAP H            /SUBR ENTRY
.DNM+1,     CLO
            LAC I (FSA)      /TEXT PTR.
            DAC G
            LAC I (USERAC) /SAVE UP MAX IN ATEM
            DAC I (ATEM)
            DZM I (USERAC) /INIT. FOR NUMBER
            LCH I G
            SAD CHLPLS       /CHECK 1ST CHAR FOR SIGN
            JMP DNMPL
            SAD (CHARAC L-)
            JMP DNMM
DNMDO,      SZF 5            /OCTAL?
            JMP DNMOCT
DNMDI,      RAL 6S
            XOR B13          /CONVERT INTERNAL TO OCTAL
            DAC I (ATEM+1)
            SUB C1Ø,
            SMA
            JMP DNMDE        /NON-NUMERIC
            STF 1            /SET NUMERIC FLAG
            LAW 1Ø,          /RADIX
            MUL I (USERAC)
            DIV B17
            JMP DNMMO        /OVERFLOW
            ADD I (ATEM+1)
            DAC I (USERAC)
DNMDL,      LCH I G          /NEXT CHAR.
            JMP DNMDI

/MINUS SIGN
DNMM,       SZF 4
            JMP DNMILL       /- WAS ILLEGAL
            STF 2            /REMEMBER SIGN
            JMP DNMSS        /SKIP SPACES

/PLUS SIGN
DNMPL,      SZF 3
            JMP DNMILL       /+ WAS ILLEGAL
DNMSS,      LCH I G          /SKIP SPACES
            SZA I
            JMP DNMSS
            JMP DNMDO        /TEST RADIX

/OVERFLOW
DNMMO,      LAC BØ
            ADD BØ           /SET OVF FF
            JMP DNMDL

/ILLEGAL FORMAT
DNMILL,     DZM I (USERAC)
            JMP DNMUX1       /DON'T CHANGE FSA, R1
```

```
/NON-NUMERIC CHARACTER ON DECIMAL DECODE
DNMDE,     SZF I 1
           JMP DNMILL      /NO NUMBER, ILLEGAL
           LIO I (USERAC)
           SZF I 6
           JMP DNMDSM      /SKIP MAX. TEST
           LAI             /GET ABS. VAL OF NUMBER
           SUB I (ATEM)    /SUB MAX
           SUB BØ          /SET OVF FF IF AC IS Ø OR MINUS
DNMDSM,    SZF 2
ML6,       CMI             /COMP. IF NEGATIVE
           DIO I (USERAC)
DNMOSM,    SZO
           JMP DNMOVF
           IDX H           /OK R2
DNMUX,     LAC G           /UNSTEP
           SMA
           SUB B1
           SUB B1
           DAC I (FSA)
DNMUX1,    CLF 1
           CLF 2
           JMP HEXIT       /EXIT THRU H

/OCTAL DECODE LOOP
DNMOCT,    CLI"U"SWP       /CHAR. INTO IO, CLA
           RCL 3S
           SAS B16         /IS CHAR, "2X" IN INTERNAL
           JMP DNMOE       /NO, TERMINATOR
           STF 1
           LAC I (USERAC)
           RCL 3S          /SHIFT IN NUMBER
           DAC I (USERAC)
           LAI             /OVERFLOW TO AC
           SUB BØ          /SET OVF FF IF POS. QUANT.
           LCH I G
           JMP DNMOCT

/NON-NUMERIC CHAR. IN OCTAL DECODE
DNMOE,     SZF I 1
           JMP DNMILL
           LAC I (USERAC)
           SZF 2
           CMA             /COMP. IF NEGATIVE
           DAC I (USERAC)
           SZF 6
           SZA I
           JMP DNMOSM      /SKIP MAX. TEST
           SCM"U"IDA       /2'S COMP, CMA
           TAD I (ATEM)    /LEAVE LINK FF=Ø IF OVERFLOW
           SZL
           JMP DNMOSM      /OK

/OVERFLOW ERROR
DNMOVF,    CLC             /AC GETS -Ø
           DAC I (USERAC)
           JMP DNMUX       /R1, FSA UPDATED
```

```
/COUNT DIGITS (COUNT IN F) SUBROUTINE
CDG,      DAP H
          LAC I (FSA)
          DAC G
CDGL,     LCH I G
          RAL 6S
          XOR B13
          SUB C1Ø,
          SMA
          JMP .DNM+1      /NON-NUMERIC, OK GO DNM
          ISP F           /COUNT NUMERIC CHAR.S
          JMP CDGL        /LOOP
          JMP HEXIT       /R1, TOO MANY CHAR'S
```

```
/SUBR TO TYPE OR SETUP FROM IO, USES PF6, ROTATES IO
TYOIF,     DAP H
           LAI
           SZF 6
           JMP TYOIF1      /TYPE
           DCH I G
           JMP TYOIF2
TYOIF1,    TYO
           JMP SICKTT
           RAL 6S
TYOIF2,    LIA
           JMP HEXIT

/SETUP NUMBER SUBR AND IOT
SNM,       JSP STYOIF      /SKIP TYPING FOR JBH MAYBE
.SNM,      LIO I (USERAC) /GENERAL-PURPOSE SUBR ENTRY
                          /ABOVE USED ONLY BY BC JBH?
.SNMJ,                    /JOB HUNTER ENTRY
           DIO I (ATEM+5) /OUTER LOOP TERMINATOR
           DAP I (ATEM+2) /RETURN
           LAC I (STS)
           SZF I 6        /TYPE BIT
           DAC G
           LAC I (USERIO) /RIGHT ADJUST COUNT
           IDA"U"LIF
           DIO I (ATEM+6)
           CMA"U"CLI
           DAC I (ATEM)
           LAW 10.
           SZF 5
           LAW 10          /OCTAL
           DAC I (ATEM+1) /RADIX
           LAC I (ATEM+5) /GET NO. BACK
           SZF I 5        /NO SIGN FOR OCTAL NUMBERS
           SMA"U"SZF 3    /INHIBIT POSITIVE SIGN BIT
           JMP SNMNS       /SKIP SIGN TYPEOUT
           SZF 4
           LIO CHLPLS
           SPA
           LIO (CHARAC L-)
           SPA
           CMA             /ABS. VAL
           DAC I (ATEM+5) /RE-DAC OUTER LOOP TERMINATOR
           IDX I (ATEM)
           JSP TYOIF       /TYPE SIGN
```

```
SNMNS,      DZM I (ATEM+4) /INNER LOOP TERMINATOR
SNMOL,      LAC I (ATEM+5)
SNML,       DAC I (ATEM+3) /INNER LOOP VALUE
            CLI"U"SWP
            RCL 1S
            DIV I (ATEM+1)
C60.K,      60000.
            SAS I (ATEM+4)
            JMP SNML       /KEEP DIVIDING (INNER LOOP
            LAW 20
            AAI            /CHAR. MADE IN AC
            RCR 6S
            IDX I (ATEM)   /CHAR COUNT
            SZF I 1        /RIGHT ADJUST BIT
            JSP TYOIF      /TYPE CHAR. IF NOT RIGHT ADJUST
            LAC I (ATEM+3)
            DAC I (ATEM+4) /RE-INIT INNER LOOP
            SAS I (ATEM+5)
            JMP SNMOL      /GET NEXT CHAR. (OUTER LOOP)
            SZF 1
            JMP SNMRA      /RIGHT ADJUST
            LAC G
            SZF I 6        /TYPE BIT
            DAC I (STS)
            LAC ML4        /(CHARAC L#)
            SZF I 6
            DCH I G        /DCH EOM AFTER PTR.
            LIO I (ATEM+6)
            LFI
            LAC I (ATEM+2)
            DAP H
            JMP H

SNMRAL,     SZF 2          /LEADING ZEROES BIT
            LIO B1         /CHARAC L0
            JSP TYOIF
SNMRA,      CLI"U"CLF 1    /RIGHT ADJUSTER
            ISP I (ATEM)
            JMP SNMRAL
            JMP SNMNS      /DO WHOLE THING OVER AGAIN, LESS SIGN

/ROUTINE TO GIVE IMMEDIATE RETURN 1 IF SETUP ROUTINE
/IS GOING TO TYPE AND JMODE IS NEGATIVE, PRECEDES ALL SETUP ROUTINES.
STYOIF,     DAP H
            LAC I (JMODE)
            SPA"U"SZF 6
CJR1,       JMP R1
            LAC CJR1       /RETURN WITH R1 IN ADDRESS OF AC
            DIP H
            JMP H
```

```
/SUBR TO TYOIF, MAKING USE OF DISCREPANCY BETWEEN G AND I (STS)
TYOIFR,    DAP H
           LAI
           SZF 6
           JMP TYOIF1
           DCH G
           LAC G
           DAC I (STS)
           JMP HEXIT
```

```
/TIME AND DATE ROUTINES
/SUBR TO SETUP DATE FROM AC INTO D-E-F
.DNUM,    DAP H
          LAC I (USERAC) /DAYS SINCE 1/1/1849
          SUB (18686.)   /DAY # OF 3/1/1900
          SMA
          IDA            /DISPLACE DAYS AFTER 3/1/1900 UP 1
          ADD (18686.)
          CLI"U"SWP
          RCL 1S
          DIV C1461.     /# OF DAYS IN A 4-YEAR CYCLE
C1461.,   1461.
          SAL 2S
          DAC F          /YEARS IN INTEGRAL 4-YEAR CYCLES
          LAI
          SUB C1154.     /DAY # OF 2/29 OF FOURTH YR OF CYCLE
          SZA I
          JMP SUTD29     /HANDLE 2/29 SEPARATELY
          SMA
          SUB B17        /DISPLACE DAYS AFTER 2/29 DOWN 1
          ADD C1154.     /NOW PRETEND ALL YEARS HAVE 365. DAYS
          CLI"U"SWP
          RCL 1S
          DIV C365.      /GET # OF YEAR WITHIN CYCLE
C1154.,   1154.
          ADD F          /ADD YEARS IN 4-YEAR CYCLES
          ADD C1849.     /AD IN ORIGIN OF YEARS
          DAC F          /EXACT YR #
          DIO D          /# OF DAY WITHIN YEAR
          LAC (SUB SUTDTB-1)       /INIT TO FIND MONTH
          DAC B
          LAI
SUTDML,   DAC E          /STORE IN LOOP, ANSWER ON IXIT
          IDX B          /LOOK AT NEXT MONTH
          LAC D
          XCT B          /SUBTRACT FROM TABLE
          SMA
          JMP SUTDML     /CONTINUE UNTIL NEGATIVE
          IDX E          /EXACT DAY OF MONTH
          LAC B
          SUB (SUB SUTDTB-1)       /CALCULATE MONTH
SUTDM,    DAC D          /STORE MONTH
          JMP HEXIT

/SUBR TO SETUP TIME FROM IO INTO B-C
.TNUM,    DAP H
          CLA
          LIO I (USERIO) /MINUTES SINCE MIDNIGHT
          RCL 1S
          DIV (60.)
C2020,    2020
          DAC B          /HOURS
          DIO C          /MINUTES
          JMP HEXIT
```

```
/FEB 29 CASE
SUTD29,   LAW 1852.       /ADD YEAR ORIGIN + 3
          ADD F           /TO # OF YEARS FROM 1 YEAR CYCLES
          DAC F
          LAW 29.         /DAY = 29.
          DAC E
          LAW 2
          JMP SUTDM

/TABLE OF # OF DAYS IN A YEAR PREVIOUS TO FIRST OF ENTRY MONTH
CZ,       Ø
SUTDTB,   31.
          59.
CBTM3,    90.
CJQBP,    120.
          151.
          181.
          212.
          243.
          273.
          304.
          334.
C365.,    365.

/ARGUMENT INDIRECT TRACING ROUTINES
TRACEI,   DAP H
CHLCOL,   DIO A
          JMP TRACEL

TRACE,    DAP H           /ENTRY TO TRACE WORD AFTER IOT
          IDX I (USERPC)
          DAC A
TRACEL,   LAC I A
          LIA
          AND (607777)
          DAC A
          RIL 5S
          SPI
          JMP TRACEL      /LOOP UNTIL I BIT SET
          JMP HEXIT       /PTR IN A

/IOT TO CONVERT FROM 2 WD TO 5 WD FORMAT
TDNUM,    JSP TRACE
          JSP .DNUM
          JSP .TNUM
          LAC CLACB       /INIT TO COPY INTO USER CORE
          DAC G
TDNUML,   XCT G
          DAC I A
          IDX A
          IDX G
          SAS CLACG
          JMP TDNUML
          JMP R1
```

```
/SETUP TIME AND DATE, IOT AND SUBR
STD,        JSP STYOIF
.STD,       DAP A           /EXIT THRU A
            LAC I (USERAC
            DAC I (ATEM+7   /SAVE UP AC
            LAC I (USERIO
            DAC I (ATEM+1Ø)          /SAVE UP IO
            LIF
            DIO D           /SAVE UP FLAGS
            DIO C
            SZF 2
            JMP SUTDD       /SUPPRESS TIME BIT
            STF 3           /INHIBIT SIGN BIT IN .SNM
            CLF 5           /OCTAL BIT CLEARED IN .SNM
            JSP .TNUM       /PUT TIME IN B_C
            LAW I 12.
            ADD B
            DAC B           /SIGN BIT OF B IS AM-PM FLAG
            SPA
            ADD C14
            SZA I
            LAW 12.         /GET NUMBER FROM 1 TO 12. FOR HOUR
            DAC I (USERAC
            LAW 2           /DIGIT COUNT IF COLLUMNATING
            DAC I  (USERIO
            JSP .SNM
            SZF 4           /CONDENSED TIME BIT
            JMP SUTDCT
            LIO CHLCOL      /:
            JSP TYOIFR
            LAC C
            DAC I (USERAC   /# OF MINUTES
            STF 2           /LEADING ZEROS
            STF 1           /RIGHT ADJUST, 2 IS IN IO
            JSP .SNM
            LAC I (STS
            DAC G           /UNSTEP G
            LIO (FLEXO  PM
            LAC B
            SPA
            LIO (FLEXO  AM
            JSP TYOIF
            JSP TYOIF
SUTDTI,     JSP TYOIF
            LIO G
            LAC ML4
            SZF I 6
            DCH I G
            DIO G
            LIO D           /RESTORE FLAGS
            DIO C
            LFI"U"SCI
            SZF I 3         /SUPPRESS DATE BIT
            JSP TYOIF       /TYPE SPACE IF BOTH TIME AND DATE SETUP
            LAC G           /UPDATE PTR.
            SZF I 6         /TYPE BIT
            DAC I (STS      /ONLY IF NOT TYPE+ING
            SZF 3           /SUPRESS DATE BIT
```

```
JMP SUTDX
LAC I (ATEM+7  /RESTORE DATE TO AC
DAC I (USERAC
```

```
SUTDD,     JSP .DNUM       /SETUP DAY IN D-E-F
           LAW 2           /DIGIT COUNT
           DAC I (USERIO
           CLF 2           /LEADING SPACES
           CLF 5           /DECIMAL
           STF 3           /INHIBIT SIGN
           LAC D           /DAY WITHIN MONTH
           DAC I (USERAC
           JSP .SNM
           LIO (CHARAC L/
           JSP TYOIFR
           LAC E           /MONTH
           DAC I (USERAC
           STF 2           /LEADING ZEROS
           JSP .SNM
           LAW 2           /TEST FLAG 5
           AND C
           SZA
           JMP SUTDX       /COMPRESS DATE, NO YEAR
           LIO (CHARAC L/
           JSP TYOIFR
           LAC F           /YEAR
           DAC I (USERAC
           JSP .SNM        /OVERRIDE 2 DIGIT COLUMNATION WITH 4 DIGIT YEAR
SUTDX,     LAC I (ATEM+7)
           DAC I (USERAC   /RESTORE AC AND IO
           LAC I (ATEM+10)
           DAC I (USERIO
AEXIT,     LAC CJ
           DIP A
CJ,        JMP A

/CONDENSED TIME HANDLER
SUTDCT,    LAC I (STS
           DAC G
           LIO CJ          /(CHARAC LP
           LAC B
           SPA
           LIO CHARLA      /(CHARAC LA
           JMP SUTDTI      /CONTINUE IN MAIN CODING
```

```
/DECODE TIME IOT, FLAGS 3,4,6
DTM,      LAC I (FSA
          DAC A           /SAVE FOR ERRORS
          DAC G
          LCH I G
          SAD (CHARAC LT)
          JMP DTMT        /RIGHT NOW
          LAW 12.         /MAX. FOR HOUR
          DAC I (USERAC
          LAW I 3
          DAC F
          JSP CDG
           JMP DUNILL
          LAC I (USERAC
          SZA I
           JMP DUNILL     /Ø IS ILLEGAL HOUR
          SAD C14
          CLA
          MUL (6Ø.)
          RCR 1
          DIO B
          LCH G
          SAS CHLCOL      /CHARAC L:
          JMP DTMB        /NO COLON, MUST BE IN BRIEF FORMAT
          LAC G
          DAC I (FSA      /UPDATE PTR,
          LAW I 3
          DAC F
          LAW 59.         /MAX FOR MINUTE
          DAC I (USERAC)
          JSP CDG
           JMP DUNILL
          LAW I 1
          SAS F           /REQUIRE EXACTLY 2 DIGITS
          JMP DUNILL
          LAC B
          ADD I (USERAC   /GET MINUTES SINCE MIDNIGHT
          DAC B
          LCH G
DTMB,     SZA I
          LCH I G         /PERMIT 1 SPACE
          LIO C72Ø.
          SAD CJ          /(CHARAC LP
          JMP DTMPM       /PM, GO ADD IN 72Ø. FROM IO
          SAS CHARLA      /(CHARAC LA
          JMP DUNILL
          CLI             /AM, ADD IN Ø FROM IO
DTMPM,    LAC B
          AAI
          DAC I (USERAC   /RETURN TIME TO AC
          LIO G           /PTR PAST A OR P
          LCH I G
          SAD (CHARAC LM
DTM2,     LIO G           /IF AM OR PM, UPDATE PTR, IN IO
          DIO I (FSA
          JMP R2
```

```
/RIGHT NOW (LOWER CORE
DTMT,       LAW SUTD+1
            PEEK
            DAC I (USERAC)
            JMP DTM2
```

```
/DECODE DATE IOT, FLAGS 3,4,6
DDT,        LAC I (FSA
            DAC A           /SAVE FOR ERRORS
            DAC G
            LCH I G         /FIRST CHARAC
            SAD (CHARAC LT)
            JMP DDTT        /N DAYS HENCE
            SAD CHARLY      /(CHARAC LY
            JMP DDTY        /N DAYS AGO
            LAW 12.
            DAC I (USERAC   /MAX FOR MONTH
            LAW I 3
            DAC F           /2 DIGITS
            JSP CDG
             JMP DUNILL
            LAC I (USERAC   /MONTH
            DAC I (ATEM+2   /SAVE FOR LATER
            LCH G
            SAS (CHARAC L/  /REQUIRE /
            JMP DUNILL
            LAC G
            DAC I (FSA      /STEP PTR.
            LAW 31.
            DAC I (USERAC   /MAX FOR DAY
            LAW I 3
            DAC F           /2 DIGITS
            JSP CDG
             JMP DUNILL
            LAC I (USERAC
            DAC I (ATEM+3   /SAVE DAY FOR LATER
            LCH G
            SAS (CHARAC L/)
            JMP DDTCY       /NO YEAR, PRESUME CURRENT
            LAC G
            DAC I (FSA)     /STEP PTR.
            LAW 2099.
            DAC I (USERAC
            LAW I 5
            DAC F           /4 DIGIT MAX
            JSP CDG
             JMP DUNILL
            IDX F
            SZA I
            JMP DDTCS       /4 DIGITS, CENTURY SPECIFIED
            SAS (-2
            JMP DUNILL
            LAW 1900.       /2DIGITS, ADD IN CURRENT CENTURY
DDTCS,      ADD I (USERAC
DDTGY,      DAC I (ATEM+4) /SAVE YEAR
DCDTDI,     CLL"U"CML"U"SCI          /SET LINK FOR LATER
            LAC I (ATEM+2  /MONTH
            ADD (ADD SUTDTB-2        /TABLE LOOK-UP
            DAC B
```

```
          SAD (ADD SUTDTB
          CLL               /CLEAR LINK IF FEBRUARY
          LAW I 1
          ADD I (ATEM+3     /GET DAY #
          XCT B             /ADD IN DAYS FROM PREVIOUS MONTHS
          DAC B             /DAY # (0-364.)
          LAW I 1849.       / - ORIGIN OF YEARS
          ADD I (ATEM+4     /GET YEARS SINCE ORIGIN
          SCR 2S            /GET # OF 4 YEAR CYCLES
          RIL 2S
          DIO C             /GET # OF LEFTOVER YEARS IN IO
          MUL C1461.        /GET # OF DAYS DUE TO 4 YEAR CYCLES
          RCR 1
          DIO D
          LAW 365.
          MUL C             /GET # OF REMAINING DAYS FROM LEFTOVER YEARS
          RCR 1
          LAI
          ADD B             /ADD POSITION OF DAT WITHIN YEAR
          SUB C1154.        /DAY 3 OF 2/28/4TH YEAR
          SMA               /DON'T ADD IF BEFORE 2/28/4TH YEAR
          TAD CZ            /ADD 1 IF NOT FEB (LINK SET)
          ADD D
          SUB (18686.-1154.)        /ADD 1154, SUB DAY#(3/1/1900)
          SMA
          SUB B17           /SUB 1 IF 3/1/1900 OR AFTER
          ADD (18686.       /RESTORE DAT #
          DAC I (USERAC)
          JSP .DNUM         /NOW CONVERT BACK TO MONTH/DAT/YEAR
          LAC I (ATEM+2
          SAS D             /ALL 3 MUST COMPARE
          JMP DUNILL
          LAC I (ATEM+3
          SAS E
          JMP DUNILL
          LAC I (ATEM+4
          SAD F
          JMP R2

/ILLEGAL SYNTAX DUN,DTM,DDT
DUNILL,   LAC A
          DAC I (FSA)       /RESTORE FSA
          JMP R1
```

```
/CURRENT YEAR
DDTCY,      LAW SUTD          /GET DATE
            PEEK
            DAC I (USERAC
            JSP .DNUM         /BREAK DOWN FIELDS TO GET YEAR
            LAC F
            JMP DDTGY

/N DAYS AGO
DDTY,       STF 2             /FLAG 2 MAKES .DNM COMP. #
            CLL"U"CML         /LINK SET MEANS "Y" AFTER .DNM
DDTT,       LAC G             /N DAYS HENCE
            DAC I (FSA        /STEP PAST Y OR T
            CLF 6             /NO MAX
            JSP .DNM
            JMP DDTO
DDTOI,      LAW SUTD
            PEEK              /GET DATE
            ADD I (USERAC     /ADD N
            DAC I (USERAC
            JMP R2
```

```
/NO NUMBER AFTER Y OR T
DDTO,       LAC I (USERAC  /FIND OUT WHY .DNM FAILED
            SZA I          /IF OVERFLOW, FAIL
            SZL
            JMP DUNILL
            JMP DDTOI      /AC=Ø, TODAY

DCDTD,      LIO (USERAC)   /PTR. TO BLOCK IN AC
            JSP TRACEI
            LAW 24.        /CHECK THAT HOUR IS LEGAL
            SUB I B
            LIO I B
            DIO A          /SAVE UP HOUR
            IAI
            SPA
            JMP R1         / "I B" >23. OR <Ø
            IDX B
            LAW 6Ø.        /CHECK THAT MINUTE IS LEGAL
            SUB I B
            IOR I B
            SPA
            JMP R1         / "I B" >59. OR <Ø
            LAW 6Ø.
CHARLL,     MUL A
            SCL 9S
            SCL 8S
            ADD I B
            DAC I (USERIO  /COMPLETE TIME
            IDX B
            LAC I B
            DAC I (ATEM+2
            IDX B
            LAC I B
            DAC I (ATEM+3)
            IDX B
            LAC I B
            DAC I (ATEM+4
            LAC I (FSA
            DAC A          /PUT FSA IN A SO DUNILL DOES'NT KILL FSA
            JMP DCDTDI

/TT TRAP CONDITION ON TYPEOUT SETUP ROUTINES
SICKTT,     LAC I (USERPC
            DAC I (TRAPPC)
            LAW TTTSU

/RETURN ENTRIES TO USER CORE
GO,         DAP I (USERPC) /GO THRU AC
IRØ,        C16RET         /INDEXABLE RETURN Ø
IR1,        JMP R1         /INDEXABLE RETURN 1
R2,         IDX I (USERPC) /RETURN 2
R1,         IDX I (USERPC) /RETURN 1
            C16RET         /TO USER CORE

START
```