# DECUS PROGRAM LIBRARY

| | |
|---|---|
| DECUS NO. | 8-602A |
| TITLE | THE PDP-8 COOKBOOK, VOLUME 1 |
| AUTHOR | Floor Anthoni |
| COMPANY | Medical Biological Laboratory TNO<br>Rijswijk, The Netherlands |
| DATE | January 1973 |
| SOURCE LANGUAGE | PAL |

# THE PDP8 COOKBOOK

BY

FLOOR ANTHONI

Medical Biological Laboratory TNO, Rijswijk 2100, The Netherlands

SUBJECT: Subroutine standardisation

## INTRODUCTION:

By the beginning of 1972, nearly 14 000 computers of the PDP8 family have been produced and field-installed. All of them have to be programmed to fulfill the tasks, dedicated to them.

The small size of most PDP8 configurations has forced most programmers to program the machine in assembly language. Many programs have since then found their way to the DECUS PROGRAM LIBRARY. The typical application-oriented programs, however, were rarely submitted to the LIBRARY, because nobody would ever be likely to apply for them. The experience, accumulated elsewhere, was therefore not available to others.

In programming the PDP8 computer. I have experienced the usefulness of program modularity at the assembly level. The basic modules are, in effect, subroutines that perform a certain function, and that have been programmed in such a way, that they can be used as "recipies" in a cookbook. When these "recipies" are being sent to a central editor, and published regularly, they will accumulate experience into a common module library, THE PDP8 COOKBOOK, available to others.

This paper proposes a norm for modules, submitted to the library.

## THE SUBROUTINE AND ITS USE

The subroutine jump certainly is the most powerful instruction of any computer. It enables the programmer to avoid duplication of code, and to build hierarchical structures of software intelligence, increasing the semantic power of each free location in core.

Subroutines in hierarchical structures will in general do the task expected from them, with a minimum of "directions" given from "above". They can, themselves, set lower level subroutines to work for them, also with a minimum of directions. These directions are in general, information, that has to be transferred down to the subroutine. The subroutine can, on the other hand, send information back. Subroutines that can be directed to do many tasks, will, in general need more "instructions" from above. The programmer has to consider this aspect with great care. The following remarks on the ways, information can be sent to and from subroutines may assist him in this respect.

When only one parameter needs to be transferred, use the ACCUMULATOR. The LINK can be used as additional YES or NO information, although it is, in general not frequently used. The use of other registers, like the MULTIPLIER-QUOTIENT register, must be strongly dissuaded, because the module will then not be able to run on many machine configurations.

More information can be transferred as arguments, following the JMS instruction. This is especially useful for parameters that can be set at assembly time, or that need not to change very often. Use the AC for frequently changing information. A common information area in page Ø can also be used. This is especially useful when those parameters need to be accessed by many modules. (For example program- and buffer-limits, pointers, etc.). The main problem of the sharing of the same storage locations, by different subroutines, is that extreme care must be exerted when calling subroutines within those subroutines.

All subroutine modules in the COOKBOOK will be provided with the storage locations they need, in order to avoid conflicting use of these locations.

Another way to circumvent such problems is to employ the techniques of reentrant and recursive programming, in which push-down list structures are being used. This aspect will not be within the scope of this paper. The concept of creating an information "vector", that is a limited area in core with all the information, in order that only the pointer to this "vector" needs to be transferred, is, however, very useful for transfers, both in and out of the subroutine.

## HOW TO PREVENT UNWANTED INTERFERENCE

When using subroutines, that have been used before, the most likely assembly error is that illegal redefinitions will result from the duplicate use of symbols. Therefore care must be taken to label a location. The following conventions are proposed: use very few tags. Put all storage locations and other items in front of the subroutine entry, that needs to have more than 3 characters. All other tags need to share, at least the first 3 characters of the subroutine entry.

Those programmers that want to "pack" subroutines into the least possible space, will find it easy to modify the subroutines in this respect.

## DOCUMENTATION

Simple subroutines need less documentation than the more sophisticated ones. Comments should be inserted, wherever additional information is needed. Avoid trivial comments like CLA/CLEAR AC, but express the general concept and thoughts, as if it were a flow chart. The documentation must be adequate for the reader to easily understand how the subroutine works. For more sophisticated routines a flow chart is a must. Each subroutine must have a compact functional description of not more than one line (52 characters). Then follows a general description of the subroutine and an example of its use. All program lines and comment lines should not exceed 52 positions, as assembler output and cross-reference numbers must have room to be inserted.

The source tape should be submitted with the tabulations, not being converted to spaces.

The listing should preferably be made with a teletype printer (teletype type of character), printed with tabs converted to spaces. Use a clean typing head and a new black ribbon, as the listing will be offset-copied. Drawings and flow-charts should be drawn with black ink, or taped with special stickers.

For the use of symbols, the reader is referred to Appendix I.

## PROGRAM SUBMISSION

Submit your program subroutine to

The Editorial Board of
The PDP8 COOK BOOK
c/o Floor Anthoni,
Medical Biological Laboratory TNO,
139, Lange Kleiweg,
RIJSWIJK (ZH),2100,
The Netherlands.

NOTE! It is of vital importance that errors are reported back to the authors or the editorial board. Only by doing so one can achieve the highest reliability of the published subroutines.

COOKBOOK VOLUME 1 CATALOG LISTED BY NUMBER

029 Read or write DECtape in both directions

030 Subroutine to pack a fixed buffer in core (300 chars) into a fixed output buffer (200 chars) in TSS8 packed format

031 Pack characters into a buffer in TSS8 format, one by one

032 As 031, but with a fixed allocated buffer

033 Unpack TSS8 format packed buffer into an output buffer

034 Unpack TSS8 format packed buffer, one character at a time

035 Subroutine to read a 6 character name in core

036 Search a file name in DN blocks (Disk monitor)

037 Search for an unused block in SAM block, and reserve it for the current file

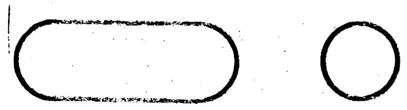
038 Search internal file number in SAM blocks (Disk Monitor)

039 Subroutine to read or write on disk (TSS8).

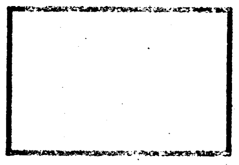
| LIST OF CONTRIBUTORS | Contributions |
|---|---|
| Floor Anthoni<br>Medisch Biologisch Laboratorium TNO,<br>Lange Kleiweg 139,<br>Rijswijk (ZH),<br>The Netherlands | 1, 2, 3, 4, 5, 6, 7, 9, 9, 10, 11, 12, 13, 14, 15, 17, 18 |
| Thierri den Dunnen<br>Dr.Neher Laboratorium,<br>St.Paulusstraat 4,<br>Leidschendam,<br>The Netherlands | 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39 |
| Hans Mees,<br>Prins Maurits Laboratoria, C.D.,<br>Lange Kleiweg 137,<br>Rijswijk (ZH),<br>The Netherlands | 8 |
| Paul Lohman,<br>Medisch Biologisch Laboratorium TNO,<br>Lange Kleiweg 139,<br>Rijswijk (ZH),<br>The Netherlands | 16 |

FLOW-CHART conventions

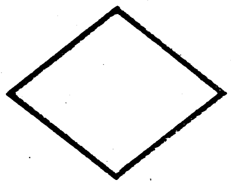
The flow-charts make use of relatively few symbols :



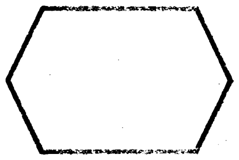
Entry, or exit of a program or sub-program, also used for the inter-connection of flow-charts on different pages.



A rectangle describes one or more program steps.



Decision, branching



Subroutine call.
A subroutine may have more than one return (branching).

/COMMENT

Comments appear behind a slash (/).

START,
LOOP,

Used to name program ties in agreement with the listings.

```
/001 TYPE THE CHARACTERS FOLLOWING THE JMS INSTR.
/TERMINATOR IS A ZERO.
/
/        JMS TYPTEX         /TYPE "ABC"
/        301                /"A"
/        302                /"B"
/        303                /"C"
/        0                  /TERMINATOR
/           RETURN          /AC=0

TYPTEX,  0
         TAD I TYPTEX       /GET CHAR.
         ISZ TYPTEX
         SNA                /ZERO?
         JMP I TYPTEX       /YES,JMP TO NEXT LOC.
         JMS TYPE           /NO,TYPE
         JMP TYPTEX+1
```

```
/[illegible]2 TELETYPE TYPE ROUTINE
/INITIALIZES  WHEN ENTERED FOR FIRST TIME.
/NOT RESTARTABLE !
/
/         TAD CHARACTER
/         JMS TYPE
/            RETURN          /AC=0

          NOP
TYPE,     0
          JMP .+3            /OVERLAID BY "NOP"
          TSF
          JMP .-1
          TLS
          CLA
          TAD TYPE-1
          DCA TYPE+1
          JMP I TYPE
```

```
/003 TYPE A CHARACTER CHAIN
/TYPE THE CHARACTERS IN THE LIST, POINTED TO
/BY THE FIRST ARGUMENT. LIST TERMINATOR =0
/
/       JMS TYPTEX       /TYPE "ABC"
/          LIST
/          RETURN        /AC=0
/
/
/
/LIST,  301
/       302
/       303
/       0

        0                /USED AS POINTER
TYPTEX, 0                /TYPE TEXTSTRING
        TAD I TYPTEX     /GET ARG
        DCA TYPTEX-1     /SAVE TO USE AS POINTER
        ISZ TYPTEX       /FOR CORRECT RETURN
        TAD I TYPTEX-1   /GET CHAR
        SNA              /ZERO?
        JMP I TYPTEX     /YES, RETURN
        JMS TYPE         /NO
        ISZ TYPTEX-1
        JMP TYPTEX+4     /LOOK FOR NEXT
```

```
/004 BINARY TO DECIMAL CONVERSION AND TYPE; NO SIGN
/ROUTINE TO CONVERT A BINARY WORD TO DECIMAL AND TYPE IT.
/VALID FOR NUMBERS 0-4095. NO SIGN.
/IF USED FOR 3 DIGITS: DELETE 6030;-4=-3 DIGIT COUNT.
/
/        TAD WORD
/        JMS PRINTD
/           RETURN          /AC=0

         6030               /-1000  CONVERSION CONSTANTS
         7634               /-100
         7766               /-10
         7777               /-1
         TAD .              /USED FOR CONV. CONSTANTS
         0                  /DIGIT BCD TO BE TYPED
         0                  /COUNTER
         260                /TO MAKE A CHAR.
         0                  /SAVE AREA
         -4                 /DIGITS TO BE TYPED (-4,-3,-2)
PRINTD,  0                  /ENTER WITH WORD IN AC
         DCA PRINTD-2
         TAD PRINTD-1       /SET UP COUNT
         DCA PRINTD-4
         DCA PRINTD-5       /CLEAR BCD
         TAD PRINTD-6       /FETCH CURR. CONV. CONST.
         TAD PRINTD-4       /BY ADDING COUNT TO TAD
         DCA .+1
         HLT
         CLL
         TAD PRINTD-2       /VALUE - CONSTANT
         SNL                /OVERFLOW?
         JMP .+4            /NO,TYPE IT
         ISZ PRINTD-5       /YES,NEXT TRY
         DCA PRINTD-2
         JMP PRINTD+5
         CLA
         TAD PRINTD-5       /BCD
         TAD PRINTD-3       /+260
         JMS TYPE
         ISZ PRINTD-4
         JMP PRINTD+4       /NEXT DIGIT
         JMP I PRINTD
```

```
/005 BINARY TO OCTAL CONVERSION AND PRINT
/ROUTINE PRINTS THE AC IN OCTAL, NO SIGN.
/
/        TAD WORD
/        JMS PRINT8
/           RETURN          /AC=Ø
/
         260
         7                  /MASK
         Ø                  /DIGIT COUNTER
         -4                 /# OF DIGITS
         Ø                  /TEMPORARY
PRINT8,  Ø
         RAL CLL
         DCA PRINT8-1
         TAD PRINT8-2
         DCA PRINT8-3       /SET UP COUNT
         TAD PRINT8-1
         RAL
         RTL
         DCA PRINT8-1
         TAD PRINT8-1
         AND PRINT8-4       /MASK
         TAD PRINT8-5       /MAKE ASCII
         JMS TYPE
         ISZ PRINT8-3       /4 DONE?
         JMP PRINT8+5       /NOT YET
         JMP I PRINT8
```

```
/ØØ6 HIGH SPEED READER SUBROUTINE
/ENTER WITH AC=Ø;ROUTINE INITIALIZES HSR.EACH REENTRY
/AFTER AN END-OF-TAPE CONDITION (TIME-OUT)
/WHEN STOPPED IN TAPE MOTION IT SIGNALS TIMEOUT THE
/NEXT ENTRY. THE ROUTINE HAS A BUILT-IN TIMING LOOP
/THAT TIMES OUT IF THE READER IS NOT SWITCHED ON,
/OR IF THE READER LOOSES ITS FLAG BY RUNNING OUT OF TAPE.
/
/       JMS HSREAD
/       OUT OF TAPE RETURN
/       NORMAL RETURN, CHAR. IN AC
/

        Ø                       /USED AS TIME-OUT COUNT
HSREAD, Ø                       /ENTRY
        DCA HSREAD-1            /SET UP COUNT
HSRFLG, 1                       /FLAG SIGNALS TO INIT READER
        TAD HSRFLG              /THESE INSTR.CONTRIBUTE TO LOOP
        SZA CLA
        JMP .+3                 /INIT READER
        RSF                     /SKIP?
        JMP .+5                 /NO, COUNT TIME-OUT
        DCA HSRFLG              /CLEAR FLAG
        6Ø16                    /READ
        ISZ HSREAD              /RETURN,CHAR IN AC
        JMP I HSREAD
        ISZ HSREAD-1
        JMP HSRFLG
        ISZ HSRFLG              /SET FLAG TO SIGNAL TIMEOUT
        JMP I HSREAD            /EOT RETURN
```

```
/007 TABULATOR ROUTINE
/THE USER HAS TO TAKE CARE OF:
/INCREMENTING TABCNT WITH EACH INCOMING CHARACTER, CLEARING
/IT WHEN CARRIAGE RETURN. TAB-INTERVAL IS VARIABLE.
/A JMS TO TAB WILL MOVE THE TYPING HEAD TO THE NEXT
/TABULATOR POSITION.
/
/         CLA
/         JMS TAB
/            RETURN          /AC=0

TABCNT,   0
          -10                /TAB INTERVAL
          240                /SPACE
TAB,      0                  /ENTER WITH AC =0
          TAD TABCNT         /SUBTRACT N TIMES TO GIVE REMAINDER
          TAD TAB-2
          SMA
          JMP .-2
          DCA TABCNT         /USE AS NEGATIVE COUNTER
          TAD TAB-1
          JMS TYPE
          ISZ TABCNT         /READY?
          JMP .-3
          JMP I TAB          /YES
```

```
/ØØ8 SUBROUTINE TO MOVE A BLOCK THROUGH CORE
/
/         CALLING SEQUENCE
/                 JMS MOVE
/                    BEGINADDRESS
/                    ENDADDRESS
/                    DESTINATION OF FIRST WORD
/                 RETURN   /AC=Ø
/
/         IF BEGINADDRESS AND ENDADDRESS ARE
/           THE SAME ADDRESS, OR BEGINADDRESS
/           IS GREATER THAN ENDADDRESS,
/           NO MOVE IS PERFORMED
/
/         IF BEGINADDRESS AND DESTINATION ARE
/           THE SAME ADDRESS, A COMPLETE MOVE
/           IS PERFORMED: YOU SHOULD BE LESS STUPID!
/
/         56 (OCTAL) CORE LOCATIONS ARE USED
          Ø
          Ø
          Ø
          Ø
/
MOVE,     Ø
          TAD I MOVE        /GET BEGINADDRESS
          DCA MOVE-1
          ISZ MOVE
          TAD I MOVE        /GET ENDADDRESS
          DCA MOVE-2
          ISZ MOVE
          TAD MOVE-2
          CMA
          TAD MOVE-1        /CALCULATE WORDCOUNT
          SNA               /IS IT POSITIV OR ZERO?
          JMP MOVRET        /YES, NO MOVE NEEDED
          DCA MOVE-4        /SAVE WORDCOUNT
          TAD MOVE-1
          CIA CLL
          TAD I MOVE        /CALCULATE MOVECOUNT
          DCA MOVE-3        /AND SAVE
          SZL               /LINK IS ON IF MOVE TO HIGHER CORE
          JMP .+3           /SKIP NEXT INSTRUCTIONS
          IAC CML           /LINK IS OFF
          TAD MOVE-4        /FIRST IN ADDRESS IS BEGINADDRESS
          TAD MOVE-2
          DCA MOVE-2        /SAVE INPUTPOINTER
          SZL               /SKIP IF MOVE TO LOWER CORE
          CLL CMA RAL       /TO HIGHER CORE, INC = -1
          IAC
          DCA MOVE-1        /SAVE INCREMENT
          TAD MOVE-2        /SET UP OUTPUTPOINTER
          TAD MOVE-3
          DCA MOVE-3        /AND SAVE
MVLOOP,   TAD I MOVE-2      /GET A WORD
          DCA I MOVE-3      /AND STORE IT IN DESTINATION BLOCK
          TAD MOVE-2
          TAD MOVE-1        /INCREMENT INPUTPOINTER
          DCA MOVE-2
          TAD MOVE-3
```

```
        TAD MOVE-1          /INCREMENT OUTPUTPOINTER
        DCA MOVE-3
        ISZ MOVE-4          /INCREMENT WORDCOUNT
        JMP MVLOOP          /AGAIN IF NOT ZERO
/
MOVRET, ISZ MOVE            /SET UP RETURNADDRESS
        JMP I MOVE          /RETURN
```

```
/009 BINARY PUNCH  WITH FIELD SETTING
/THREE SUBROUTINES TO PUNCH AN AREA OF CORE IN BINARY
/LOADER FORMAT. FIELD SETTINGS AND ORIGIN SETTINGS
/ARE BEING PUNCHED AT EACH ENTRY; CHECKSUM IS PUNCHED
/WHEN PUNCHK IS CALLED.
/THE ROUTINE CAN TAKE DATA FROM A DIFFERENT FIELD.
/IT CAN OPERATE IN ALL FIELDS. SEVERAL USES APPLY:
/1) NORMAL USE. THE DATA IS LOCATED IN THE SAME FIELD OR
/STRANGE FIELD. ENTER WITH FIELD IN AC; LINK=0
/2) THE CODE IS IN SAME FIELD AS BINPUN, ONLY A DIFFERENT
/FIELD SETTING NEEDS TO BE PUNCHED. ENTER BINPUN
/WITH FIELD IN AC AND LINK=1.
/3) THE CODE HAS BEEN MOVED IN CORE. THE FIRST LOCATION
/IS NOT NECESSARILY THE ORIGIN. NOW ENTER BINPUN WITH
/AC=FIELD FOR SETTING; LINK=1; SET ORIGIN UNEQUAL TO
/FIRST LOCATION IF THIS IS TRUE.
/EXAMPLE OF NORMAL USE:
/
/       JMS LEADER          /PUNCH LEADER, CLEAR CHECKSUM
/       CLL
/       TAD (0010           /FIELD 1
/       JMS BINPUN
/           ORIGIN          /IN NORMAL USE=FIRST LOC.
/           FIRST LOC.
/           LAST LOC.
/       JMS PUNCHK          /PUNCH CHKSM AND TRAILER

BINEND, 0                   /LAST LOC. TO PUNCH
        6201
        100
        300                 /FOR FIELD SETTING
        0                   /TEMP. STORAGE
BINPUN, 0
        DCA BINPUN-1
        SZL                 /SET UP DF IF LINK=0
        JMP BIN3
        TAD BINPUN-1
        TAD BINPUN-4        /MAKE CDF
BIN3,   DCA BIN2
        TAD BINPUN-1        /MAKE FIELD SETT. AND PUNCH
        TAD BINPUN-2        /NOTE!!FIELD SETT. NOT IN CHECKSUM!!
        JMS TYPE
        TAD I BINPUN        /GET ORIGIN
        ISZ BINPUN
        DCA BINPUN-1
        TAD BINPUN-1        /PUNCH ORIGIN
        JMS BINLH           /LEFT HALF
        TAD BINPUN-3        /+100 FOR ORIGIN
        JMS BINCHK
        TAD BINPUN-1        /RIGHT HALF AND PUNCH
        AND BINLH-1
        JMS BINCHK
        TAD I BINPUN        /SET UP POINTER
        DCA BINPUN-1
        ISZ BINPUN
        TAD I BINPUN        /GET END
        DCA BINEND
BIN2,   CDF 0               /OVERLAID BY CDF STRANGE FIELD
        TAD I BINPUN-1      /GET DATA
        JMS BINLH           /PUNCH LEFT HALF
        JMS BINCHK
        TAD I BINPUN-1   /PUNCH RIGHT HALF
        AND BINLH-1
        JMS BINCHK
        TAD BINEND       /END REACHED?
        CIA
        CLL
        TAD BINPUN-1
        ISZ BINPUN-1
        SNL CLA
        JMP BIN2+1       /NO, TAKE NEXT DATA
        RIF              /YES,RESTORE DF
        TAD BINPUN-4
        DCA .+1
        CDF Ø            /OVERLAID
        ISZ BINPUN
        JMP I BINPUN
        SPA CLA

/GET LEFT HALF OF THE AC
        77
BINLH,  Ø
        RTR
        RTR
        RTR
        AND BINLH-1
        JMP I BINLH

/UPDATE THE CHECKSUM, AND PUNCH FRAME
        Ø                /CHECKSUM
BINCHK, Ø                /ENTER WITH 6 BIT FRAME IN AC
        DCA BINLH        /USE BINLH TEMPORARILY
        TAD BINLH
        TAD BINCHK-1     /UPD. CHKSM
        DCA BINCHK-1
        TAD BINLH
        JMS TYPE
        JMP I BINCHK

/PUNCH 1ØØ(8) LEADER OR TRAILER HOLES; CLEAR CHECKSUM
        2ØØ
LEADER, Ø
        TAD BINPUN-3     /USE CHKSM AS NEG COUNT
        CIA
        DCA BINCHK-1
        TAD LEADER-1
        JMS TYPE
        ISZ BINCHK-1     /READY?
        JMP .-3          /NO
        JMP I LEADER     /YES

/PUNCH CHECKSUM;PUNCH TRAILER;CLEAR CHECKSUM

PUNCHK, Ø
        TAD BINCHK-1     /GET CHECKSUM
        JMS BINLH
        JMS TYPE
        TAD BINCHK-1
        AND BINLH-1
        JMS TYPE
        JMS LEADER       /CLEARS CHECKSUM
        JMP I PUNCHK
```

```
/010 PAL MESSAGE PRINTER
/PRINTS A MESSAGE CODED WITH THE PAL PSEUDO-OP
/'TEXT'. PAL3 AND PAL8 COMPATIBLE
/
/        JMS PRMESG
/           MESG
/        RETURN   /AC=0
/
/MESG,   TEXT 'ABC82' /CODED AS 0102;0370;6200

         77                  /MASK
         -40                 /TO TEST
         240                 /TO MAKE ASCII
         100                 /TO MAKE ASCII
         0                   /PACKSWITCH 0=LEFT;7777=R
         0                   /POINTER
PRMESG,  0
         CMA                 /SAVE POINTER(-1)
         TAD I PRMESG
         DCA PRMESG-1
         ISZ PRMESG          /FOR RETURN
PRM1,    CMA
         DCA PRMESG-2        /PACKSW=RIGHT
         ISZ PRMESG-1        /NEXT WORD
         TAD I PRMESG-1      /FETCH AND ROTATE 6
         RTR
         RTR
         RTR
PRM2,    AND PRMESG-6        /MASK 6 BITS
         SNA
         JMP I PRMESG        /ZERO ENDS THE LIST
         TAD PRMESG-5        /<40 ?
         SPA
         TAD PRMESG-3        /YES ASCII 301-337
         TAD PRMESG-4        /NO, ASCII 240-277
         JMS PRINT
         ISZ PRMESG-2        /LEFT OR RIGHT?
         JMP PRM1            /LEFT
         TAD I PRMESG-1      /RIGHT
         JMP PRM2
```

```
/011 GENERAL BRANCH ROUTINE
/BRANCH ROUTINE BRANCHES ACCORDING TO THE CONTENTS
/OF THE AC, COMPARED TO EACH ITEM OF A LIST.
/EXIT FROM BRANCH IS ALWAYS WITH AC=Ø
/
/       TAD AC
/       JMS BRANCH
/               LIST-1
/       RETURN IF NOT IN LIST (AC=Ø)
/
/LIST,  212
/       LF                /IF "CHAR"=212,PROGRAM JUMPS TO "LF"
/       215
/       CR;ETC;ETC........
/       Ø                 /Ø IS LIST TERMINATOR!!!!!!

        Ø                 /AC
        Ø                 /BRANCH POINTER
BRANCH, Ø                 /ENTER WITH ARGUMENT IN "CHAR"
        DCA BRANCH-2
        TAD I BRANCH
        ISZ BRANCH
        DCA BRANCH-1      /INIT POINTER
BRANC,  ISZ BRANCH-1
        TAD I BRANCH-1    /FETCH ELEMENT FRM LIST
        SNA               /END OF LIST?
        JMP I BRANCH      /YES
        CIA
        TAD BRANCH-2
        ISZ BRANCH-1
        SZA CLA
        JMP BRANC         /NO,TRY NEXT
        TAD I BRANCH-1    /YES, GO TO IT
        DCA BRANCH
        JMP I BRANCH
```

```
/Ø12 CHECK IF OCTAL
/ROUTINE CHECKS WHETHER THE AC IS AN OCTAL DIGIT.
/
/        TAD CHARACTER
/        JMS OCTCHK
/          NOT OCTAL RETURN        /AC=Ø
/          OCTAL RETURN            /AC=Ø

        1Ø
        -27Ø
OCTCHK, Ø
        TAD OCTCHK-1
        SMA
        JMP OCT2
        TAD OCTCHK-2
        SPA CLA
        JMP I OCTCHK
        ISZ OCTCHK
OCT2,   CLA
        JMP I OCTCHK
```

```
/Ø13 LOGICAL OPERATORS ON TWO NUMBERS
/THE RESULT OF LOGICAL OPERATIONS IS IN THE AC.

/AND (MASKING)              A 1Ø1Ø
/                           B 11ØØ
/                           = 1ØØØ
        TAD A
        AND B

/INCLUSIVE OR               A 1Ø1Ø
/SETS BITS B IN A           B 11ØØ
/                           = 111Ø
        TAD A
        CMA
        AND B
        TAD A

/CLEAR BITS B IN A          A 1Ø1Ø
/                           B 11ØØ
/                           = ØØ1Ø
        TAD B
        CMA
        AND A

/NOR                        A 1Ø1Ø
/                           B 11ØØ
/                           = ØØØ1
        TAD A
        CMA
        DCA TEM
        TAD B
        CMA
        AND TEM

/NAND                       A 1Ø1Ø
/                           B 11ØØ
/                           = Ø111
        TAD A
        AND B
        CMA

/EXCLUSIVE OR               A 1Ø1Ø
/                           B 11ØØ
/                           = Ø11Ø
        TAD A
        AND B
        CMA
        DCA TEM
        TAD A
        AND TEM

        TAD B
        AND TEM
```

```
/014 PS8-0S/8 OPTION DECODER
/CHECKS THE OPTION, SPECIFIED IN THE AC AND CAUSES
/A RETURN, DEPENDING ON WHETHER THE OPTION HAS BEEN
/SET
/OPTIONS IN OS8 RESIDE IN FIELD 1 LOC 7643-7645 :
/
/7643  A B C D E F G H I J K L   ASCII 301-314
/7644  M N O P Q R S T U V W X   ASCII 315-330
/7645  Y Z 0 1 2 3 4 5 6 7 8 9   ASCII 331,332,260-271
/
/       TAD (16             /CHECK OPTION 16 (N)
/       JMS OPTION
/           OPTION NOT SET RETURN/AC=0
/           OPTION SET RETURN    /AC=0
/

OPTM1,  7777
        -14                 /-12(10)
        7642                /POINTER
        0                   /TEMP. STORAGE
        7642                /COUNTER, ALSO POINTER
OPTION, 0                   /ENTER WITH POSITION IN AC
        DCA OPTION-2
        TAD OPTION-3        /RESTORE COUNTER
        DCA OPTION-1
        TAD OPTION-2        /SUBTRACT 12 TO FIND WORD
        TAD OPTION-4
        ISZ OPTION-1
        SMA SZA
        JMP .-3
        TAD OPTM1           /FOR L AND X
        DCA OPTION-2        /SAVE REMAINDER MODULO 12
        CLL CML             /AND ROTATE ONE BIT INTO POSITION
        RAL                 /ROTATE FURTHER
        ISZ OPTION-2
        JMP .-2
        CDF 10              /AND WITH OPT WORD FIELD 1
        AND I OPTION-1
        CDF 0
        SZA CLA
        ISZ OPTION          /IN CASE IT HAD BEEN SET
        JMP I OPTION
```

```
/015 PRINT TWO DIGITS IN DECIMAL
/THE VALUE OF THE AC IS PRINTED IN TWO DIGITS
/CORRECTLY IF < 99(DECIMAL).
/
/       TAD (VALUE
/       JMS PRNT2
/       RETURN              /AC=0

        260                 /TO MAKE ASCII
        -12                 /10 DECIMAL
        0                   /TEMP STORAGE
        0                   /COUNTER
PRNT2,  0
        DCA PRNT2-2
        TAD PRNT2-2         /TRY SUBTRACT 10 UNTIL OVFLO
        TAD PRNT2-3
        SPA
        JMP .+3
        ISZ PRNT2-1         /SUBTRACT FURTHER
        JMP PRNT2+1
        CLA
        TAD PRNT2-1         /PRINT HIGH ORDER DIGIT
        TAD PRNT2-4
        JMS PRINT
        TAD PRNT2-2
        TAD PRNT2-4
        JMS PRINT
        DCA PRNT2-1         /RESET COUNTER
        JMP I PRNT2
```

```
/Ø16 PRINT THE PS8-OS8 DATE
/THE DATE IS PRINTED AS: Ø7/17/72
/THE ROUTINE MAKES USE OF PRNT2, TO TYPE TWO
/DECIMALS. REQUIRES ROUTINES PRNT2 AND PRINT.
/DATE IN OS8 IS STORED IN LOC 7666, FIELD 1:
/
/7666  MMMMDDDDDYYY /M=MONTH,D=DAY,Y=YEAR
/
/       JMS DATE
/       RETURN              /AC=Ø

DATM,   7                   /MASKS
        17
        37
        257                 /SLASH
        1Ø6                 /7Ø YEARS
        Ø                   /STORAGE
        7666                /DATE LOC. IN OS8
DATE,   Ø
        CDF 1Ø              /PICK TH E DATE
        TAD I DATE-1
        CDF Ø
        DCA DATE-2
        TAD DATE-2
        CLL RTL             /SHIFT MONTH OUT
        RTL
        RAL
        AND DATM+1          /AND (17
        JMS PRNT2
        TAD DATE-4          /PRINT SLASH
        JMS PRINT
        TAD DATE-2
        RTR                 /SHIFT MONTH OUT AND PRINT
        RAR
        AND DATM+2
        JMS PRNT2
        TAD DATE-4          /SLASH
        JMS PRINT
        TAD DATE-2          /NOW THE YEAR
        AND DATM
        TAD DATE-3          /+7Ø
        JMS PRNT2
        JMP I DATE
```

```
/017 PRINT THE AC AS A FOCAL LINENUMBER
/THE VALUE OF THE AC IS PRINTED AS IN FOCAL:11.35
/XX.YY STORED AS FOLLOWS: XXXXXYYYYYYY IN 1 WORD.
/IF YYYYYYY>99 STRANGE DIGITS OCCUR AS IN FOCAL.
/REQUIRES ROUTINES PRNT2 AND PRINT.
/
/       TAD VALUE
/       JMS PRNTF
/       RETURN          /AC=0

PRNTFM, 37              /MASKS
        177
        256             /PERIOD.
        0               /STORAGE
PRNTF,  0
        DCA PRNTF-1
        TAD PRNTF-1     /ISOLATE AND PRINT HIGH ORDER
        CLL RTL
        RTL
        RTL
        AND PRNTFM      /AND (37
        JMS PRNT2
        TAD PRNTF-2
        JMS PRINT
        TAD PRNTF-1     /NOW LOW ORDER
        AND PRNTFM+1
        JMS PRNT2
        JMP I PRNTF
```

```
/018 PRINT 4 DECIMAL DIGITS USING  ROUTINE PRNT2
/THE CONTENT OF THE AC IS DIVIDED BY 100(10)
/GIVING TWO LOW ORDER DIGITS AND 2 HIGH ORDER.
/THESE ARE PRINTED BY PRNT2.
/
/        TAD VALUE
/        JMS PRNT4
/        RETURN             /AC=0

         7634               /-100(10)
         0                  /STORAGE AND LOW ORDER
         0                  /HIGH ORDER COUNTER
PRNT4,   0
         DCA PRNT4-2
         CLL
         TAD PRNT4-2        /TRY TO SUBTRACT 100 UNTIL OVERFLOW
         TAD PRNT4-3
         SNL
         JMP .+3
         ISZ PRNT4-1
         JMP PRNT4+1
         CLA
         TAD PRNT4-1        /PRINT HIGH ORDER DIGITS
         JMS PRNT2
         TAD PRNT4-2        /PRINT LOW ORDER DIGITS
         JMS PRNT2
         DCA PRNT4-1        /RESET COUNTER
         JMP I PRNT4
```

```
/019 SUBROUTINE READS A DECIMAL NUMBER FROM KEYBD
/RUBOUT REMOVES NUMBER COMPLETELY
/
/
/CALL    :JMS DECINP
/         RETURN WITH NUMBER BINARY IN AC
/
/
DECINP,0
         CLA
         DCA DECNUM          /CLEAR REGISTER
         JMS READ            /READ CHAR FROM KEYBOARD
         TAD CHAR
         JMS PRINT           /PRINT THAT CHAR
         TAD CHAR            /GET CHARACTER
         TAD M377            /IS IT RUBOUT?
         SNA CLA
         JMP DECINP+1        /YES READ ALL OVER AGAIN
         TAD CHAR            /NO
         TAD M260
         SPA                 /CHAR>=260?
         JMP DECOUT          /NO,CHARACTER IS DELIMETER
         TAD M12             /YES
         SMA CLA             /CHAR<272?
         JMP DECOUT          /NO,CHAR IS DELIMETER
         TAD DECNUM          /YES,CHAR IS FIGURE
         CLL RAL
         DCA DECTMP          /NUMB.*2
         TAD DECTMP
         RTL                 /NUMB*8
         TAD DECTMP          /NUMB*8+NUMB*2=NUMB*10
         TAD CHAR            /ADD LAST FIGURE
         TAD M260
         DCA DECNUM          /DECIMAL NUMBER
         JMP DECINP+3
/
DECOUT,CLA
         TAD DECNUM
         JMP I DECINP        /EXIT
/
/VARIABLES
/
DECNUM,0
DECTMP,0
/
/GENERAL CONSTANTS
M12,     -12
M260,    -260
M377,    -377
```

```
/020 DECIMAL PRINT ROUTINE,
/PRINTS AC DECIMAL IN 4 DIGITS
/MAX NUMBER = 4095 DECIMAL
/SKIPS LEADING ZERO'S
/
DPRT,   0
        DCA DPRREG          /SAVE AC IN PRINTREG.
        TAD DPRINS          /GET INSTRUCTION
        DCA DPRPTP          /PUT INSTR. ON POINTER
        TAD M4
        DCA DPRFAC          /4 DIGITS
        DCA DPRFL           /CLEAR PRINT 0 FLAG
        DCA DPRFIG          /CLEAR DIGIT
DPRSUB,CLL
        TAD DPRREG          /PICK UP SAVED AC
DPRPTP,TAD DPRTEN           /SUBTRACT POWER OF TEN
        SNL                 /REMAINDER POSITIVE?
        JMP .+4             /NO, PRINT DIGIT
        DCA DPRREG          /YES, SAVE REMAINDER
        ISZ DPRFIG          /DIGIT:=DIGIT+1
        JMP DPRSUB          /REPEAT SUBTRACTION
        CLA CLL
        TAD DPRFIG          /GET DIGIT
        SNA                 /A ZERO?
        JMP DPRZRO          /YES
DPRIN, TAD C260             /NO, CONVERT TO ASCII
        JMS PRINT
        ISZ DPRFL           /MAKE NOT EQUAL 0
DPRIN1,ISZ DPRPTP           /MODIFY INSTR ON DPRPTP
        ISZ DPRFAC          /PRINTED 4 DIGITS?
        JMP DPRSUB-1        /NO, PRINT NEXT DIGIT
        JMP I DPRT          /YES, RETURN
/
DPRZRO,TAD DPRFL
        SZA CLA
        JMP DPRIN
        JMP DPRIN1
/
DPRREG,0
DPRFL, 0
DPRINS,TAD DPRTEN
DPRFAC,0
DPRFIG,0
DPRTEN,6030       /-1000
        7634      /-100
        7766      /-10
        7777      /-1
/
/GENERAL CONSTANTS
M4,     -4
C260,   260
```

```
/021 SUBROUTINE TO PRINT DOUBLE LENGTH DECIMAL
/
/CALL: JMS DDECPR
/       MOST SIGNIFICANT PART
/       LEAST SIGNIFICANT PART
/       NUMBER OF DIGITS TO BE PRINTED ( <=8 )
/       RETURN
/
/
DDECPR,0
        TAD I DDECPR    /FETCH MOST SIGNIFICANT PART
        DCA DDX         /SAVE
        DCA DDPD        /CLR NUMB. OF PRINTED DIGITS
        ISZ DDECPR
        TAD I DDECPR    /FETCH LEAST SIGNIFICANT PART
        DCA DDX+1       /SAVE
        ISZ DDECPR
        TAD I DDECPR    /FETCH FORMAT
        DCA DDNDIG
        ISZ DDECPR      /CORRECT RETURN
        TAD DDATPL      /ADDRESS 10-POWER LOW
        DCA DDPTPL      /POINTER 10-POWER LOW
        TAD DDATPH      /ADDRESS 10-POWER HIGH
        DCA DDPTPH      /POINTER 10-POWER HIGH
        TAD M10
        DCA DDNFAC      /FACTORISE 8 DIGITS
        DCA DDIGIT      /CLEAR DIGIT
DDSUB,  CLL
        TAD DDX+1       /L SIGNIFIC PART OF NUMB.
        TAD I DDPTPL    /LOW PART FACTOR
        DCA DDX+1       /STORE
        RAL             /OVERFLOW IN AC
        TAD DDX         /M SIGNIFIC PART OF NUMB.
        TAD I DDPTPH    /HIGH FACTOR
        SNL             /RESULT NEGATIVE?
        JMP .+4         /YES
        DCA DDX         /STORE RESULT OF SUBTRACTION
        ISZ DDIGIT      /NO, STEP UP DIGIT
        JMP DDSUB       /SUBTRACT 2-LENGTH AGAIN
        CLA             /CLEAR BEFORE CORRECTION
        TAD I DDPTPL    /10-POWER LOW
        CIA             /MINUS
        TAD DDX+1       /CORRECT LAST SUBTRACTION
        DCA DDX+1       /STORE
        TAD DDIGIT      /GET DIGIT
        SZA             /=0?
        JMP DDPDIN      /NO
        TAD DDPD
        SZA CLA         /ALREADY PRINTED?
        JMP DDPDIN      /YES
        IAC
        TAD DDNFAC
        SMA CLA         /ALL DI#D0= 0?
        JMP DDPDIN      /YES
        TAD DDNFAC      /NEGATIVE VALUE
        TAD DDNDIG      /POSITIVE VALUE
        SPA CLA         /SPACE?
        JMP DDPTIN
        TAD C240        /YES
        JMP DDFPR
```

```
DDPDIN,ISZ DDPD
        TAD C260        /CONVERT TO ASCII
DDPPR,  JMS PRINT       /PRINT DIGIT
DDPTIN,ISZ DDPTPL       /STEP UP POINTER LOW
        ISZ DDPTPH      /STEP UP POINTER HIGH
        ISZ DDNFAC      /READY FACTORIZE?
        JMP DDSUB-1     /NO, NEXT DIGIT
        TAD DDPD
        CIA
        TAD DDNDIG
        SPA SNA CLA
        JMP .+3
        TAD DDNDIG
        DCA DDPD
        CLL
        JMP I DDECPR    /EXIT, END PUNCHOUT ROUTINE
/
/
/CONSTANTS PUNCH OUT ROUTINE
DDATPL,DDTPL
DDATPH,DDTPH
DDPTPL,Ø
DDPTPH,Ø
DDX,    Ø
        Ø
DDNFAC,Ø
DDIGIT,Ø
DDTPL,  46ØØ
        67ØØ
        454Ø
        436Ø
        6Ø3Ø
        7634
        7766
        7777
DDTPH,  3166
        7413
        7747
        7775
        7777
        7777
        7777
        7777
DDPD,   Ø
DDNDIG, Ø
/
/GENERAL CONSTANTS
M1Ø,    -1Ø
C24Ø,   24Ø
C26Ø,   26Ø
```

```
/Ø22 OCTAL PRINT ROUTINE
/NONSIGNIFICANT ZERO'S BECOME SPACES
/
/       CLA
/       DCA OCTFIG          /CLEAR FLAG FIGURE PRINTED
/       DCA OCTSPC          /CLEAR SPACE-COUNTER
/       TAD NUMBER
/CALL   :JMS OCTPRT         / WITH NUMBER IN AC
/       RETURN AC=Ø         /IF NUMBER=Ø,
/OCTSPC=4, = # OF SPACES TO PRINT
/       IF NUMBER IS ZERO,OCTSPC=4 IS #SPACES TO PRINT
/
OCTPRT, Ø
        RAL                 /ROTATE IN LINK
        DCA OCTTMP          /TEMP. STORAGE
        TAD M4              /4 OCTADES
        DCA OCTCNT
OCTPRØ, TAD OCTTMP
        RAL
        RTL
        DCA OCTTMP          /STORE RESULT
        TAD OCTTMP          /GET IT BACK
        AND C7              /MASK OCTADE
        SNA CLA             /ZERO ?
        JMP OCTZER          /YES
        TAD OCTSPC          /NO,SPACES TO PRINT?
        SNA
        JMP OCTNUM          /NO,GO PRINT FIGURE
        CIA                 /YES,SET COUNTER
        DCA OCTSPC
        TAD C24Ø
        JMS PRINT           /PRINT THE SPACES
        ISZ OCTSPC
        JMP .-3
OCTNUM, CLA IAC             /SET FLAG FIG. PRINTED
        DCA OCTFIG
        TAD OCTTMP
        AND C7
OCTØPR, TAD C26Ø            /MAKE THE FIGURE
        JMS PRINT
OCTPR1, ISZ OCTCNT          /READY?
        JMP OCTPRØ          /NO
        JMP I OCTPRT        /YES,EXIT
/
OCTZER, TAD OCTFIG
        SZA CLA             /FIGURES PRINTED ?
        JMP OCTØPR          /YES,PRINT THIS ZERO TOO
        ISZ OCTSPC          /NO COUNT AS SPACE
        JMP OCTPR1
/
OCTTMP,Ø
OCTCNT,Ø
OCTSPC,Ø
OCTFIG,Ø
M4,     -4
C7,     7
C24Ø,   24Ø
C26Ø,   26Ø
```

```
/023 DOUBLE WORD OCTAL PRINT ROUTINE
/USES ROUTINE OCTPR
/CALLING:JMS DOCTPR
/HIGH ORDER NUMBER
/LOW ORDER NUMBER
/RETURN AC=0
/
DOCTPR,0
        CLA
        DCA OCTFIG        /CLEAR FLAG FIGURE PRINTED
        DCA OCTSPC        /CLEAR SPACE-COUNTER
        TAD I DOCTPR      /HIGH ORDER PART
        ISZ DOCTPR
        JMS OCTPRT        /PRINT OCTAL
        TAD I DOCTPR      /LOW ORDER PART
        ISZ DOCTPR
        JMS OCTPRT        /PRINT OCTAL
        TAD OCTSPC
        CIA
        SNA               /SPACES TO PRIT?
        JMP I DOCTPR      /NO, EXIT
        IAC               /YES,NUMBER IS ZERO
        DCA OCTSPC        /PRINT SPACES
        TAD C240
        JMS PRINT
        ISZ OCTSPC
        JMP .-3
        TAD C260          /AND A "0"
        JMS PRINT
        JMP I DOCTPR      /EXIT
```

```
/024 SUBROUTINE TRANSLATES TELEX TO ASCII
/
/CALL    :JMS TLXAS WITH TELEX CHARACTER IN AC
/         RETURN CHARACTER IS SHIFT
/         RETURN WITH ASCII CHARACTER IN AC
/
/WHO IS TRANSLATED AS $
/? IS TRANSLATAD AS *
/BELL IS TRANSLATED AS ;
/
TLXAS,    Ø
          AND C37          /MASK 5 BITS
          DCA TLXTMP       /TEMP. STORAGE
          TAD TLXTMP
          SNA
          JMP TLXOUT       /BLANK
          TAD M2
          SNA
          JMP TLXCR        /CARRIAGE RETURN
          TAD M6
          SNA
          JMP TLXNL        /NEW LINE
          TAD M23
          SNA
          JMP TLXSW1       /FIGURESHIFT
          TAD M4
          SNA CLA
          JMP TLXSWØ       /LETTERSHIFT
          TAD TLXTMP       /GET CHARACTER AGAIN
          TAD TLXLA        /ADD LISTADDRESS
          DCA TLXTMP       /TEMP STORAGE
          TAD TLXSW        /WHICH SIDE?
          SZA CLA
          JMP TLXRGT       /RIGHT SIDE
          TAD I TLXTMP     /GET ASCII 6 BIT
          RTR
          RTR
          RTR
TLXMS,    AND C77          /MASK 6 BIT
          TAD M4Ø
          SPA
          TAD C1ØØ         /CHAR<40:3ØØ<=CHAR<=337
          TAD C24Ø         /CHAR>4Ø:24Ø<=CHAR<=277
TLXOUT,   ISZ TLXAS        /NORMAL RETURN
          JMP I TLXAS
/
TLXRGT,   TAD I TLXTMP
          JMP TLXMS
/
TLXSW1,   IAC
TLXSWØ,   DCA TLXSW        /REMEMBER WICH SHIFT
          JMP I TLXAS      /RETURN SHIFT
/
TLXCR,    TAD C215
          JMP TLXOUT
TLXNL,    TAD C212
          JMP TLXOUT
/
TLXLA,    TLXLST
TLXLST,   Ø
```

```
        2465    /T      5
C37,    37
        1771    /O      9
        4040    /SPACE
        1036    /H      ↑
        1654    /N      ,
        1556    /M      .
M40,    -40
        1451    /L      )
        2264    /R      4
        0735    /G      ]
        1170    /I      8
        2060    /P      0
        0372    /C      :
        2675    /V      =
        0563    /E      3
        3253    /Z      +
        0477    /D      WHO=$
        0252    /B      ?=*
        2347    /S      '
        3166    /Y      6
        0633    /F      [
        3057    /X      /
        0155    /A      -
        2762    /W      2
        1273    /J      BELL=;
C100,   100
        2567    /U      7
        2161    /Q      1
        1350    /K      (
/
/VARIABLES
/
TLXTMP,0
TLXSW, 0
/
/GENERAL CONSTANTS
M2,     -2
M4,     -4
M6,     -6
M23,    -23
C77,    77
C212,   212
C215,   215
C240,   240
```

```
/025 SUBROUTINE TO TRANSLATE TELEX CHAR TO ASCII
/
/CALL:   JMS TLXAS1
/        RETURN IF SHIFT CHARACTER
/        RETURN
/
TLXAS1,  0
         AND TLX37
         DCA TLXTMP         /STORE
         TAD TLXTMP
         TAD TLXM37
         SNA                /LETTERSHIFT?
         JMP TLXLSH         /YES, SET SHIFT
         TAD C4
         SNA CLA            /FIGURESHIFT?
         JMP TLXFSH         /YES, CLEAR SHIFT
         TAD TLXTMP
         TAD TLXSH
         TAD TLXLST
         DCA TLXTMP
         TAD I TLXTMP
         ISZ TLXAS1
         JMP I TLXAS1
/
TLXLSH,  TAD TLX40
TLXFSH,  DCA TLXSH
         JMP I TLXAS1
/
/
TLXLST,  .+1
         0000     /BLANK
         "5
         0215     /CR
         "9
         0240     /SPACE
         0000
         ",
         ".
         0212     /LF
         ")
         "4
         0000
         "8
         "0
         ":
         "=
         "3
         "+
         0205     /WRU
         "?
         "'
         "6
         0000
         "/
         "-
         "2
         0207     /BELL
TLXSH,   0
         "7
         "1
```

```
        "(
TLX40,   40
        0000
        "T
        0215     /CR
        "O
        "        /SPACE
        "H
        "N
        "M
        0212     /LF
        "L
        "R
        "G
        "I
        "P
        "C
        "V
        "E
        "Z
        "D
        "B
        "S
        "Y
        "F
        "X
        "A
        "W
        "J
TLX37,   37
        "U
        "Q
        "K
TLXM37, -37
/VARIABLES
/
TLXTMP,0
/
/GENERAL CONSTANTS
/
C4,     4
```

```
/026 ROUTINE TO TRANSLATE ASCII TO TELEX
/CALL    :JMS ASTLX
/         RETURN
/
/BEFORE FIRST CALL INITIALIZE ASTSFT:=4 AND
/PRINT A LETTERSHIFT
/
/NOT EXISTING CHARACTERS ARE PRINTED AS BLANK
/ALTMODE IS TRANSLATED AS FIGURESHIFT
/RUBOUT IS TRANSLATED AS LETTERSHIFT
/
ASTLX,    0
        DCA ASTTMP          /TEMP. STORAGE
        TAD ASTTMP
        AND C77             /MAKE 6 BIT
        SNA
        JMP ASTOUT+2        /BLANK=BLANK
        TAD ASTLA           /LISTADDRESS
        DCA ASTHLP          /LISTADDRESS + 6-BIT CHAR
        TAD ASTTMP
        TAD M300
        SMA CLA
        JMP ASTBIG          /CHAR>=300;RIGHT HALF OF LIST
        TAD I ASTHLP        /CHAR<300;LEFT HALF OF LIST
        RTR
        RTR
        RTR
        SKP
ASTBIG, TAD I ASTHLP
        DCA ASTTMP          /TEMP. STORAGE
        TAD ASTTMP
        AND C77
        SNA
        JMP ASTOUT+2        /NOT EXISTING IN TELEX:BLANK
        AND C40             /GET SHIFT BIT
        SZA CLA             /WHICH SHIFT
        JMP ASTSHF          /MUST BE FIGURES
        TAD ASTSFT          /MUST BE LETTERS
        SZA CLA             /IS IT LETTERS?
        JMP ASTOUT          /YES,PRINT CHAR
        CLA CLL IAC RTL     /+4;NO, MAKE AND PRINT
ASTPSH, DCA ASTSFT
        TAD ASTSFT
        TAD C33             /MAKE SHIFT
        JMS PRINT           /PRINT
ASTOUT, TAD ASTTMP
        AND C37             /MASK 5 BITS
        JMS PRINT           /PRINT
        JMP I ASTLX         /EXIT
/
ASTSHF, TAD ASTSFT          /MUST BE FIGURES
        SNA CLA             /IS IT FIGURES?
        JMP ASTOUT          /YES,PRINT CHAR
        JMP ASTPSH          /NO,MAKE AND PRINT
/
ASTLA,   ASTLST
ASTLST, 0000     /@
        0030     /A
        0023     /B
```

```
ØØ16    /C
ØØ22    /D
622Ø    /WHO,  E
ØØ26    /F
7213    /BELL,  G
ØØØ5    /H
ØØ14    /I
1Ø32    /NL,  J
ØØ36    /K
ØØ11    /L
Ø2Ø7    /CR,  M
ØØØ6    /N
ØØØ3    /O
ØØ15    /P
ØØ35    /Q
ØØ12    /R
ØØ24    /S
ØØØ1    /T
ØØ34    /U
ØØ17    /V
ØØ31    /W
ØØ27    /X
ØØ25    /Y
ØØ21    /Z
ØØØØ    /[
ØØØØ    / \
ØØØØ    /]
ØØØØ    /↑
ØØØØ    /←
Ø4ØØ    /SPACE
ØØØØ    /
ØØØØ    /"
ØØØØ    /#
ØØØØ    /$
ØØØØ    /%
ØØØØ    /&
64ØØ    /'
76ØØ    /(
51ØØ    /)
ØØØØ    /*
61ØØ    /+
46ØØ    /,
7ØØØ    /-
47ØØ    /.
67ØØ    / /
55ØØ    /Ø
75ØØ    /1
71ØØ    /2
6ØØØ    /3
52ØØ    /4
41ØØ    /5
65ØØ    /6
74ØØ    /7
54ØØ    /8
43ØØ    /9
56ØØ    /:
ØØØØ    /;
ØØØØ    /<
```

```
        5773      /=, ALTMOD=FIGSHIFT
        0000      />
        6337      /?, RUBOUT=LETTERSHIFT
/
/VARIABLES
/
ASTSFT,0
ASTTMP,0
ASTHLP,0
/
/GENERAL CONSTANTS
/
C33,    33
C37,    37
C40,    40
C77,    77
M300,   -300
```

```
/027 INTERRUPT OUTPUT HANDLER
/WITH HEAD-TAIL COUPLED BUFFER
/
/INITIALIZE ONCE BUFIPT:=BUFOPT:=BUFFER
/         BUFIBO:=0
/
/
/CHARACTER HANDLER
/
/CALL     :JMS BUFINP WITH CHAR IN AC
/         RETURN AC=0
/
/
BUFINP,  0
         DCA BUFTMP          /TEMP. STORAGE
         TAD BUFIBO          /INPTR BEHIND OUTPTR?
         SNA CLA
         JMP BUFPUT          /NO,STORE CHARACTER
         TAD BUFIPT          /YES
         CIA
         TAD BUFOPT          /INPTR = OUTPRT ?
         SNA CLA
         JMP BUFINP+2        /YES, WAIT FOR PLACE TO STORE
BUFPUT,  TAD BUFTMP          /NO,GET CHAR
         DCA I BUFIPT
         ISZ BUFIPT
         TAD BUFBUS          /PRINTER BUSY?
         SNA CLA
         6046                /NO,INIT WITH AC=0
         IAC                 /YES,SET PRINTER BUSY
         DCA BUFBUS
         TAD BUFIPT
         TAD BUFBND          /END OF BUFFER?
         SZA CLA
         JMP I BUFINP        /NO,EXIT
         TAD BUFADR          /YES,POINTER TO HEAD
         DCA BUFIPT
         IAC                 /AND SET INPTR BEHIND OUTPTR
         DCA BUFIBO
         JMP I BUFINP        /EXIT
```

```
/028 DEVICE INTERRUPT HANDLER
/
/CALL    :JMP BUFOUT       /DEVICE INTERRUPT DETECTED!
/        ROUTINE RETURNS TO INTERRUPT RESTORE "EXIT"
/
BUFOUT,  CLA
         6042              /CLEAR DEVICE FLAG
         TAD BUFIPT
         CIA
         TAD BUFOPT        /INPTR = OUTPTR ?
         SZA CLA
         JMP BUFGET        /NO,GET CHAR AND PRINT
         TAD BUFIBO        /YES,INPTR BEHIND OUTPTR?
         SZA CLA
         JMP BUFGET        /YES,GET AND PRINT
         DCA BUFBUS        /NO,PRINTER READY
         JMP EXIT
/
BUFGET,  TAD I BUFOPT      /GET CHAR
         ISZ BUFOPT
         6044      /PRINT CHAR
         CLA
         TAD BUFOPT
         TAD BUFBND        /END OF BUFFER?
         SZA CLA
         JMP EXIT          /NO,END OF ROUTINE
         TAD BUFADR        /YES,POINTER TO HEAD
         DCA BUFOPT
         DCA BUFIBO        /RESET INPTR BEHIND OUTPTR
         JMP EXIT          /END OF HANDLING
/
/GENERAL INTERRUPT RETURN ROUTINE
/
EXIT,    CLA CLL
/        TAD LINK
/        RAL               /RESTORE LINK
/        TAD ACCU          /RESTORE ACCU
/        ION               /INTERRUPT ON
/        JMP I 0
/VARIABLES
BUFTMP,0
BUFIBO,0
BUFIPT,0
BUFOPT,0
BUFBUS,0
BUFBND,-BUFEND
BUFADR,BUFFER
BUFFER,0
/
*BUFFER+200
BUFEND,0
```

```
/029 SUBROUTINE READS OR WRITES DECTAPE
/IN BOTH DIRECTIONS
/
/CALL    :JMS DCTAPE
/         DEFINING BITS
/         BLOCKNUMBER
/         -# WORDS (12 BITS)
/         BUFFERADDRESS-1
/         ERROR RETURN OR RETURNADDRESS
/         NORMAL RETURN OR RETURNADDRESS
/
/DEFINING BITS:BIT 0,1,2          UNIT NUMBER
/         3          0=FORWARD;1=REVERSE
/         4,5        0 (NOT USED)
/         6,7,8      MEMORY FIELD
/         9          0 (NOT USED)
/         10         0=DIRECT RETURN;1=INDIRECT
/         11         0=READ;1=WRITE
/
DTCA=     6762
DTXA=     6764
DTLB=     6774
DTRA=     6761
DTSF=     6771
DTRB=     6772
/
DCTAPE,   0
          CLA
          TAD I DCTAPE        /DEFINING BITS
          DCA DCTCOD          /SAVE
          ISZ DCTAPE
          TAD DCTCOD
          AND C7400           /UNIT# & DIRECTION BIT
          TAD C10             /SEARCH MODE
          DTCA DTXA           /I/O
          DTLB                /CLEAR FIELD REGISTER
          TAD DCTWC           /WORD COUNT ADDRESS
          DCA I DCTCA         /WORD COUNT:=BLKNR ADDRESS
          TAD C200            /GO BIT
DCTCNT,   JMS DCTTRN          /TURN DECT AND WAIT FOR FLAG
          TAD I DCTWC         /READ NUMBER
          CIA                 /NEG.
          TAD I DCTAPE        /NUMBER TO FIND
          SNA
          JMP DCTMAY          /FOUND, CHECK DIRECTION
DCTSET,   CLL RAL             /SAVE SIGN DIFFERENCE
          CLA
          DTRA
          AND C400            /DIRECTION BIT
          SNA CLA
          CML                 /IS FORWARD
          SNL                 /IS REVERSE
          TAD C400            /CHANGE DIRECTION
          JMP DCTCNT          /DIRECTION OK, NEXT NUMBER
/
DCTMAY,   TAD DCTCOD          /UNIT# & DIRECTION
          AND C400            /MASK DIRECTION
          SNA CLA
          JMP DCTRFW          /MUST BE FORWARD
          DTRA                /MUST BE REVERSE
          AND C400
          SZA CLA
          JMP DCTRDR          /IS REVERSE, GO READ OR WRITE
          JMP DCTCNT          /IS FORWARD, CONT SEARCHING
DCTRFW,   DTRA                /MUST BE FORWARD
          AND C400
          SNA CLA
          JMP DCTRDR          /IS FORWARD, GO READ OR WRITE
          JMP DCTCNT          /IS REVERSE, CONT SEARCHING
/
/
DCTRDR,   ISZ DCTAPE
          TAD I DCTAPE        /-# WORDS
          DCA I DCTWC         /SET WORD COUNT
          ISZ DCTAPE
          TAD I DCTAPE        /CORE ADDRESS-1
          DCA I DCTCA         /SET CURRENT ADDRESS
          TAD DCTCOD
          DTLB                /LOAD FIELD BITS
          TAD DCTCOD
          RAR
          SZL CLA             /READ OR WRITE?
          TAD C20             /WRITE
          TAD C130            /WRITE
          DTXA
          DTSF DTRB
          JMP .-1
          ISZ DCTAPE          /ADVANCE TO ERRORRETURN
          SMA CLA             /SKIP IF ERROR
          ISZ DCTAPE          /NORMAL RETURN
          TAD DCTCOD          /DIRECT OR INDIRECT?
          RTR
          SNL CLA
          JMP .+3             /DIRECT
          TAD I DCTAPE        /INDIRECT, PREPARE
          DCA DCTAPE
          DTRA
          AND C200            /GO BIT
          TAD C2              /PRESERVE ERROR FLAG
          DTXA                /STOP TAPE
          JMP I DCTAPE        /READY, EXIT
/
/
DCTTRN,   0
          DTXA
          DTSF DTRB
          JMP .-1
          SPA
          JMP DCTERR
          CLA
          JMP I DCTTRN
/
DCTERR,   RTL
          RAL
          CLA CML
          SNL
          TAD C400
          JMP DCTCNT-1
```

```
/
/
/VARIABLES
/
DCTCOD,0
DCTWC, 7754
DCTCA, 7755
/
/GENERAL CONSTANTS
/
C2,      2
C10,     10
C20,     20
C130,    130
C200,    200
C400,    400
C7400, 7400
```

```
/Ø3Ø SUBROUTINE TO PACK CHARACTERS (TSS8)
/THREE CHARACTERS IN TWO WORDS (TSS8 FORMAT)
/PACKED:111111112222
/       222233333333
/
/CALL   :JMS PACK
/       ADDRESS INPUTBUFFER
/       ADDRESS OUTPUTBUFFER
/       RETURN
/
/ROUTINE USES AUTO INDEX 1Ø AND 11
/
/FORMAT INPUTBUFFER= 1 CHAR/WRD
/LENGTH OUTPUTBUFFER= 2ØØ
/LENGTH INPUTBUFFER= 3ØØ
/
PACK,   Ø
        TAD PCKBFL          /-BUFFERLENGTH OUTPUTBUFFER
        STL RAR             /DEVIDE BY 2
        DCA PCKCNT
        CLA CMA             /-1
        TAD I PACK          /ADDRESS INPUTBUFFER
        DCA 1Ø
        ISZ PACK
        CMA                 /-1
        TAD I PACK          /ADDRESS OUPUTBUFFER
        DCA 11
        ISZ PACK
PCKLOP, TAD I 1Ø            /GET CHAR
        CLL RTL
        RTL
        DCA PCKTMP          /TEMP. STORAGE
        TAD I 1Ø            /NEXT CHAR
        RTR
        RTR
        DCA PCKTP1
        TAD PCKTP1
        AND C17
        TAD PCKTMP
        DCA I 11            /FIRST WORD
        TAD PCKTP1          /PICK UP AGAIN
        RAR
        AND C74ØØ
        TAD I 1Ø            /NEXT CHAR
        DCA I 11            /SECOND WORD
        ISZ PCKCNT          /BUFFER FULL ?
        JMP PCKLOP          /NO,PACK NEXT
        JMP I PACK          /YES, EXIT
/
/VARIABLES
/
PCKCNT,Ø
PCKTMP,Ø
PCKTP1,Ø
PCKBFL,-2ØØ
/
/GENERAL CONSTANTS
C17,    17
C74ØØ, 74ØØ
```

```
/Ø31 SUBROUTINE PACKS CHARACTERS ONE BY ONE (TSS8)
/THREE CHARACTERS IN TWO WORDS (TSS8 FORMAT)
/PACKED:111111112222
/       222233333333
/
/
/CALL   :JMS PCKSGL WITH CHAR IN AC
/        ADDRESS OF OUTPUTBUFFER
/        RETURN BUFFER FULL
/        RETURN NARMAL   AC=Ø
/
/INITIALIZE CE PCKSWT:=Ø
/
/
/
PCKSGL, Ø
        ISZ PCKSWT      /INITIALIZE?
        JMS PCKINI      /YES
        DCA I PCKRP     /NO PUT CHAR IN TEMP BUF
        ISZ PCKRP       /INCREMENT POINTER
        ISZ PCKRCT      /3 CHAR'S IN TEMP BUF?
        JMP PCKNRM      /NO,NORMAL EXIT
        JMS PCKRES      /YES,RESET POINTER TEMP. BUF
        TAD I PCKRP     /GET FIRST CHAR
        ISZ PCKRP
        CLL RTL
        RTL
        DCA I PCKPTR    /TEMP STORAGE
        TAD I PCKRP     /GET SECOND CHAR
        ISZ PCKRP
        RTR
        RTR
        DCA PCKSWT
        TAD PCKSWT      /TEMP. STORAGE
        AND C17         /MOST SIGN. 4 BITS
        TAD I PCKPTR
        DCA I PCKPTR    /FIRST WORD
        ISZ PCKPTR
        TAD PCKSWT
        RAR
        AND C74ØØ       /LEAST SIGNIFICANT 4 BITS
        TAD I PCKRP     /GET THIRD CHAR
        DCA I PCKPTR    /SECOND WORD
        ISZ PCKPTR
        JMS PCKRES      /RESET POINTER TEMP BUF
        ISZ PCKCNT      /BUFFER FULL?
        JMP PCKNRM      /NO
        DCA PCKSWT      /YES SET SWITCH
        JMP PCKEND
/
/
PCKNRM, CMA
        DCA PCKSWT      /SET SWITCH
        ISZ PCKSGL
PCKEND, ISZ PCKSGL
        JMP I PCKSGL
/
/
PCKINI, Ø
        DCA PCKSWT      /TEMP STORAGE
        JMS PCKRES          /SET POINTER TEMP BUF
        TAD I PCKSGL        /GET BUFFERADDRESS
        DCA PCKPTR
        TAD PCKBFL
        STL RAR             /BUFFERSIZE DEVIDED BY 2
        DCA PCKCNT
        TAD PCKSWT
        JMP I PCKINI
/
/
PCKRES, Ø
        TAD M3
        DCA PCKRCT          /TEMP BUF IS 3 WORDS
        TAD PCKRBA          /TEMP BUF ADDRESS
        DCA PCKRP
        JMP I PCKRES
/
/
/VARIABLES
/
PCKSWT,Ø
PCKPTR,Ø
PCKRP, Ø
PCKRCT,Ø
PCKCNT,Ø
PCKRBA,PCKRB
PCKBFL,-4ØØ
PCKRB, Ø
        Ø
        Ø
/
/GENERAL CONSTANTS
M3,     -3
C17,    17
C74ØØ, 74ØØ
```

```
/032 SUBROUTINE TO PACK CHARACTERS ONE BY ONE (TSS8)
/THREE CHARACTERS IN TWO WORDS (TSS8 FORMAT)
/PACKED:111111112222
/       222233333333
/
/CALL   :JMS DSOUT WITH CHAR IN AC
/        RETURN BUFFER FULL
/        RETURN NORMAL
/
/INITIALIZE ONCE DSPTR TO BUFFERADDRESS
/AND DSCNT:=DSBFL DEVIDED BY 2
/
BSW=7002
DSBUF=400
/
/
DSOUT,  0
        DCA DSTMP           /TEMP. STORAGE
        RAR
        DCA DSLNK           /SAVE LINK
        TAD DSCNTW          /FIRST,SECOND OR THIRD CHAR
        CLL RTR
        SNL SMA CLA
        JMP DSFRST          /FIRST CHAR OF THREE
        SNL
        JMP DSSEC           /SECOND CHAR OF THREE
        TAD DSTMP           /THIRD CHAR
        TAD I DSPTR
        DCA I DSPTR         /PUT IN BUFFER
        DCA DSCNTW          /RESET CHAR COUNT
        ISZ DSPTR
        ISZ DSCNT           /BUFFER FULL ?
        JMP DSEX3           /NO, EXIT
        TAD DSBFA           /YES, RESET POINTER
        DCA DSPTR
        TAD DSBFL           /-BUFFERLENGTH
        STL RAR             /DEVIDE BY 2
        DCA DSCNT
        TAD DSLNK           /RESTORE LINK
        CLL RAL
        JMP I DSOUT         /EXIT BUFFER FULL
DSSEC,  TAD DSTMP
        CLL RTL
        BSW                 /BYTE SWAP
        AND C77
        TAD I DSPTR
        DCA I DSPTR
        ISZ DSPTR
        TAD DSTMP
        AND C17
        BSW
        CLL RTL
        DCA I DSPTR
        JMP DSEX2
DSFRST, TAD DSTMP
        CLL RTL
        RTL
        DCA I DSPTR
DSEX2,  ISZ DSCNTW
DSEX3,  TAD DSLNK           /RESTORE LINK
```

```
          CLL RAL
          ISZ DSOUT
          JMP I DSOUT       /NORMAL EXIT
/
/VARIABLES
/
DSBFL, -400
DSBFA, DSBUF      /OUTPUT BUFFER ADDRESS
DSLNK, 0
DSTMP, 0
DSCNTW,0
DSCNT, 0
DSPTR, 0
/
/GENERAL CONSTANTS
C17,      17
C77,      77
```

```
/Ø33 SUBROUTINE TO UNPACK CHARACTERS (TSS8)
/PACKED THREE CHARACTERS IN TWO WORDS (TSS8 FORMAT)
/
/PACKED:111111112222
/       222233333333
/CALL   :JMS UNPACK
/        ADDRESS OF INPUTBUFFER
/        ADDRESS OF OUTPUTBUFFER
/        RETURN
/
/ROUTINE USES AUTO-INDEX 1Ø
/
UNPACK, Ø
        TAD UNPBFL      /-BUFFERLENGTH INPUTBUFFER
        STL RAR         /DEVIDE BY 2
        DCA UNPCNT
        TAD I UNPACK    /ADDRESS INPUTBUFFER
        DCA UNPPTR
        ISZ UNPACK
        CLA CMA         /-1
        TAD I UNPACK    /ADDRESS OUTPUTBUFFER
        DCA 1Ø
        ISZ UNPACK
UNPLOP, TAD I UNPPTR
        RTR
        RTR
        AND C377
        DCA I 1Ø        /FIRST CHAR
        TAD I UNPPTR    /PICK UP CHAR AGAIN
        CLL RTL
        RTL
        AND C36Ø
        DCA UNPTMP      /TEMP. STORAGE
        ISZ UNPPTR
        TAD I UNPPTR
        CLL RAL
        RTL
        RTL
        AND C17
        TAD UNPTMP
        DCA I 1Ø        /SECOND CHAR
        TAD I UNPPTR
        AND C377
        DCA I 1Ø        /THIRD CHAR
        ISZ UNPPTR
        ISZ UNPCNT      /READY ?
        JMP UNPLOP      /NO,CONTINUE
        JMP I UNPACK    /YES, EXIT
/
/VARIABLES
/
UNPPTR,Ø
UNPTMP,Ø
UNPCNT,Ø
UNPBFL,-4ØØ
/
/GENERAL CONSTANTS
C17,    17
C36Ø,   36Ø
C377,   377
```

```
/Ø34 SUBROUTI   UNPACKS CHARACTERS ONE BY ONE (TSS8)
/PACKED THREE CHARACTERS IN TWO WORDS (TSS8 FORMAT)
/PACKED:111111112222
/       222233333333
/
/CALL   :JMS UNPSGL
/       ADDRESS INPUTBUFFER
/       RETURN BUFFER EMPTY     AC=Ø
/       NORMAL RETURN AC=CHAR.
/
/INITIALIZE ONCE UNPRBF:=UNPBEF:=UNPCNT:=Ø
/
/
UNPSGL, Ø
        CLA CLL
        TAD UNPRBF          /ARE THERE CHAR'S IN
        SZA CLA             /TEMP. BUFFER ?
        JMP UNPGET          /YES, GET ONE
        TAD UNPBEF          /NO, INPUTBUFFER EMPTY ?
        SZA CLA
        JMP UNPEMP          /YES,RETURN END OF BUFFER
        TAD UNPCNT          /NO OR YES,MUST I
        SNA CLA             /START UP POINTERS ?
        JMS UNPINI          /YES, PLEASE DO
        TAD UNPRBA          /NO,JUST UNPACK NEXT WORDS
        DCA UNPRP
        TAD I UNPPTR        /NEXT WORD FROM INPUTBUF
        RTR
        RTR
        AND C377
        DCA I UNPRP         /FIRST CHAR IN TEMP. BUF
        ISZ UNPRP
        TAD I UNPPTR        /GET WORD AGAIN
        CLL RTL
        RTL
        AND C36Ø
        DCA I UNPRP         /TEMP. STORAGE
        ISZ UNPPTR
        TAD I UNPPTR        /NEXT WORD
        CLL RAL
        RTL
        RTL
        AND C17
        TAD I UNPRP
        DCA I UNPRP         /SECOND CHAR
        ISZ UNPRP
        TAD I UNPPTR        /THAT WORD AGAIN
        ISZ UNPPTR
        AND C377
        DCA I UNPRP         /THIRD CHAR
        TAD UNPRBA          /RESET POINTER TEMP. BUF
        DCA UNPRP
        CLA CLL CMA RTL     /-3
        DCA UNPRCT          /3 CHAR'S IN TEMP. BUF
        ISZ UNPCNT          /INPUTBUFFER EMPTY ?
        JMP UNPGET          /NO,GET CHAR NOW
        IAC                 /YES,SET FLAG BUFFER EMPTY
        DCA UNPBEF          /AND THAN GET CHAR
UNPGET, ISZ UNPRCT          /LAST FROM TEMP. BUF ?
        IAC                 /NO,SET FLAG
        DCA UNPRBF      /YES RESET FLAG
        TAD I UNPRP     /GET CHAR
        ISZ UNPRP
        ISZ UNPSGL      /NORMAL EXIT
UNPEMT, ISZ UNPSGL
        JMP I UNPSGL
/
UNPEMP, DCA UNPBEF      /RESET FLAG
        JMP UNPEMT      /AND EMPTY BUFFER RETURN
/
UNPINI, 0
        DCA UNPRBF      /RESET FLAG
        TAD I UNPSGL    /ADDRESS INPUTBUFFER
        DCA UNPPTR
        TAD UNPBFL      /-LENGTH OF BUFFER
        STL RAR         /DIVIDE BY 2
        DCA UNPCNT
        JMP I UNPINI
/
/VARIABLES
UNPBFL,-400
UNPCNT,0
UNPRCT,0
UNPRP, 0
UNPPTR,0
UNPRBF,0
UNPBEF,0
UNPRBA,UNPRB
UNPRB, 0
        0
        0
/
/GENERAL CONSTANTS
C17,    17
C360,   360
C377,   377
```

```
/035 SUBROUTINE TO READ A NAME FROM KEYBOARD
/
/CALL    :JMS RDNAME
/         WORD 1,2 CHAR'S FROM NAME IN EXCESS-40 CODE
/         WORD 2,2 CHAR'S FROM NAME
/         WORD 3,2 CHAR'S FROM NAME
/ERROR RETURN
/NORMAL RETURN
/
/ROUTINE USES AUTO INDEX 10,ROUTINES READ,PRINT
/AND CRLF
/
BSW=7002
BUFADR=400
/
/
RDNAME, 0
        TAD RDNMBF          /ADDRESS ASCII BUFFER
        DCA RDPTR
        DCA RDCNT           /CHAR. COUNTER
RDIN,   JMS READ            /READ CHAR FROM KEYB.
        DCA RDCHAR
        TAD RDCHAR
        TAD RDMRO           /RUBOUT ?
        SNA
        JMP RDROS           /YES, TO SERVICE
        TAD RDMCRN          /NO,CARRIAGE RETURN ?
        SNA
        JMP RDTWNR          /YES,TO SERVICE
        TAD RDMLFD          /NO,LINE FEED
        SNA
        JMP RDTWNR          /YES,SAME SERVICE AS CR
        TAD RDMSPE          /NO,CHAR>240 ?
        SPA SNA CLA
        JMP RDFTNM          /NO,ERRORRETURN
        TAD RDCHAR          /YES,IN BUFFER
        DCA I RDPTR
        ISZ RDCNT           /+# CHAR'S
        ISZ RDPTR
        JMP RDIN            /NEXT CHAR
        JMP RDFTNM          /4K BUFFER FULL,ERROR
/
RDROS,  TAD RDCNT           /ALREADY SOMETHING IN BUFFER?
        SNA CLA
        JMP RDIN            /NO,STUPID RO-TYPER!
        CMA                 /YES COUNTER BACK 1
        TAD RDCNT
        DCA RDCNT
        CMA                 /AND POINTER BACK 1
        TAD RDPTR
        DCA RDPTR
        TAD I RDPTR         /PRINT REMOVED CHAR
        JMS PRINT
        JMP RDIN            /END RO-SERVICE
/
RDTWNR, JMS CRLF            /PRINT CR LF
        TAD RDCNT
        SNA
        JMP RDFTNM          /NAME WITHOUT CHAR'S IS RUBBISH
        TAD M6
        SMA SZA         /SIX OR LESS CHAR'S
        CLA             /MORE, THAN MAKE IT SIX
        TAD C6
        CIA
        DCA RDCNT       /-# CHAR'S
        TAD RDNMBF      /BUFFER ADDRESS
        DCA RDPTR
        TAD RDNAME      /PACKED NAME ADDRESS
        DCA RDTMP       /PLACED UNDER CALLING
        TAD RDTMP
        DCA 10
        DCA I 10
        DCA I 10        /CLEAR BUFFER
RDNXT,  TAD I RDPTR     /MAKE EXCESS-40 CODE
        TAD C240
        AND C77
        BSW
        DCA I RDTMP
        ISZ RDPTR
        ISZ RDCNT
        SKP
        JMP RDNMOK      /READY READING NAME
        TAD I RDPTR     /NOT READY NEXT CHAR
        TAD C240
        AND C77
        TAD I RDTMP
        DCA I RDTMP
        ISZ RDTMP
        ISZ RDPTR
        ISZ RDCNT
        JMP RDNXT       /NEXT CHAR'S
RDNMOK, ISZ RDNAME      /NORMAL RETURN
RDFTNM, ISZ RDNAME
        ISZ RDNAME
        ISZ RDNAME
        JMP I RDNAME    /EXIT
/
/
/VARIABLES
/
RDNMBF,BUFADR   /ADDRESS BUFFER
RDPTR,  0
RDCNT,  0
RDCHAR,0
RDTMP,  0
RDMRO,  -377
RDMCRN,377-215
RDMLFD,215-212
RDMSPE,212-240
/
/GENERAL CONSTANTS
M6,     -6
C6,     6
C77,    77
C240,   240
```

```
/Ø36 SUBROUTINE SEARCHES NAME IN DN-BLOCKS (DISKMON.)
/(DISK MONITOR SYSTEM)
/
/CALL    :JMS DNSRC
/        NA       FIRST TWO CHAR'S IN EXCESS-4Ø 6 BIT
/        ME       LAST     "         "          "        "          "
/        RETURN NAME NOT FOUND    AC=Ø
/        RETURN NAME FOUND        AC=INT. FILE NR
/
/SUBROUTINE USES AUTO INDEX 11 AND MONITOR DISK HANDLER
/
BUFFER=4ØØ
/
DNSRC,   Ø

         TAD C177           /# FIRST DN-BLOCK
         JMS DNSRBK         /READ BLOCK
         TAD I DNSRC
         CIA
         DCA DNSMNA         /- TWO CHAR'S OF NAME
         ISZ DNSRC
         TAD I DNSRC
         CIA
         DCA DNSMME         /- LAST CHAR'S
         ISZ DNSRC
DNSBLK,  CLA CLL IAC RAL /+2
         TAD DNSBFA         /BUFFER ADDRESS
         DCA 11
         TAD M31            /# ENTRIES IN ONE BLOCK
         DCA DNSCNT
DNSNXT,  TAD I 11           /FIRST HALF OF NAME
         TAD DNSMNA         /COMPARE WITH NAME TO LOOK FOR
         SZA CLA            /EQUAL?
         JMP DNSNOT         /NO TRY NEXT NAME
         TAD I 11           /YES, TEST 2ND. HALF TOO
         TAD DNSMME
         SZA CLA            /EQUAL?
         JMP DNSNT1         /NO NEXT NAME
         ISZ 11
         ISZ 11
         TAD I 11
         AND C7             /MASK OF INT FILE #
         ISZ DNSRC
DNSERR,  JMP I DNSRC
/
DNSNOT,  CLA IAC
DNSNT1,  TAD C3
         TAD 11
         DCA 11
         ISZ DNSCNT         /END OF THIS BLOCK?
         JMP DNSNXT         /NO,COMPARE NEXT NAME
         TAD DNSLNK         /YES NEXT BLOCK?
         SNA
         JMP DNSERR         /NO, NAME NOT FOUND
         JMS DNSRBK         /READ THAT BLOCK
         JMP DNSBLK
/
DNSRBK,  Ø
         DCA FSTBLK
         TAD C3
```

```
        DCA FUNCTI
        TAD DNSBFA
        DCA BUFADR
        DCA DNSLNK
        JMS I SYSIO          /MONITOR DISK HANDLER
FUNCTI,0
FSTBLK,0
BUFADR,0
DNSLNK,0
        HLT                  /ERROR RETURN
        JMP I DNSRBK
/
/VARIABLES
/
SYSIO, 7642
DNSMNA,0
DNSMME,0
DNSCNT,0
DNSBFA,BUFFER
/
/GENERAL CONSTANTS
C3,     3
C7,     7
C177,   177
M31,    -31
```

```
/037 SUBROUTINE SEARCHES UNUSED BLOCK ON DISK (DISKMON)
/AND RESERVES IT FOR FILE        (DISK MONITOR SYSTEM)
/
/CALLING:JMS SAMFIL       WITH INT. FILE NR IN AC
/        RETURN DISK FULL
/        RETURN NORMAL WITH BLOCKNR IN AC
/
/
SAMFIL, 0
        DCA SAMSAV        /SAVE INT FILE #
        JMS SAMSRC        /SEARCH FOR EMPTY BLOCK
        JMP I SAMFIL      /NOT FOUND SO DISK FULL
        CLA CMA           /BLOCKNR STILL IN SAMBKN
        TAD 10            /AUTO INDEX STILL ON SPOT
        DCA 10
        TAD SAMMSK        /WHICH HALF IS MASK
        TAD M77
        SNA CLA           /LEFT OR RIGHT?
        JMP SAMRGT        /MASK IS ON RIGHT HALF
        TAD SAMSAV
        CLL RTL           /PUT INT FILE # ON LEFT HALF
        RTL
        RTL
        DCA SAMSAV
        JMP .+3
SAMRGT, TAD I 10
        TAD SAMSAV        /ADD INT FILE #
        DCA SAMSAV        /TEMP. STORAGE
        CMA
        TAD 10
        DCA 10
        TAD SAMSAV        /PUT IN BUFFER
        DCA I 10
        TAD C5
        DCA FUNCTI
        TAD SAMBFA
        DCA BUFADR
        JMS SAMRDB        /RESTORE SAM ON DISK
        ISZ SAMFIL
        TAD SAMBKN        /GET BLOCKNR
        JMP I SAMFIL      /RETURN
/
/VARIABLES
/
SAMSAV,0
/
/GENERAL CONSTANTS
/
M77,    -77
C5,     5
```

```
/038 SUBROUTINE SEARCHES INT. FILE # (DISKMON)
/IN SAMBLOCKS (DISK MONITOR SYSTEM)
/
/CALL    :JMS SAMSRC WITH INT. FILE # IN AC
/         RETURN NUMBER NOT FOUND; AC=Ø
/         RETURN NR FOUND,AC=# FIRST BLOCK FROM FILE
/
/SUBROUTINE USES AUTO INDEX 1Ø AND MONITOR DISK HANDLER
/
/
BUFFER=4ØØ
/
/
SAMSRC, Ø
        DCA SAMIFN          /INT FILE # TO SEARCH FOR
        TAD SAMIFN          /MAKE IT TWO IN ONE WORD
        CLL RTL
        RTL
        RTL
        TAD SAMIFN
        DCA SAMIFN
        DCA SAMBKN          /COUNTER FOR BLOCKNR
        TAD C2ØØ            /# FIRST SAMBLOCK
SAMRBK, DCA BLKNR
        TAD C3              /READ FUNCTION
        DCA FUNCTI
        TAD SAMBFA          /BUFFER ADDRESS
        DCA BUFADR
        JMS SAMRDB          /READ BLOCK
SAMSR,  TAD C77
        DCA SAMMSK          /SEARCH RIGHT HALF
        TAD M2ØØ
        DCA SAMCNT          /2ØØ WORDS
        CMA
        TAD SAMBFA
        DCA 1Ø
        SKP
SAMNXT, ISZ SAMBKN          /COUNT BLOCKNR
        TAD I 1Ø            /GET WORD
        AND SAMMSK          /MASK
        CIA                 /NEGATIV
        DCA SAMTMP          /TEMP. STORAGE
        TAD SAMIFN          /INT FILE # TO SEARCH FOR
        AND SAMMSK          /MASK CORRECT HALF
        TAD SAMTMP          /SAME #?
        SNA CLA
        JMP SAMFND          /YES, FOUND IT
        ISZ SAMCNT          /NO, MORE IN THIS HALF?
        JMP SAMNXT          /YES, SEARCH
        ISZ SAMBKN          /NO, UPDATE BLOCKNR
        TAD SAMMSK          /WHERE WERE WE SEARCHING?
        AND C77ØØ
        SZA CLA             /LEFT OR RIGHT HALF
        JMP .+3             /LEFT HALF, BOTH SIDES DONE
        TAD C77ØØ           /RIGHT HALF, SO DO LEFT NOW
        JMP SAMSR+1
        TAD SAMLNK          /LAST SAMBLOCK?
        SNA
        JMP SAMNOT          /YES, SO NOT FOUND
        JMP SAMRBK          /NO, READ NEXT BLOCK
```

```
/
/
SAMFND, ISZ SAMSRC
        TAD SAMBKN
SAMNOT, JMP I SAMSRC
/
/
SAMRDB,Ø
        JMS I SYSIO       /MONITOR DISK HANDLER
FUNCTI,Ø          /READ=3, WRITE=5
BLKNR, Ø          /BLOCKNR
BUFADR,Ø          /BUFFERADDRESS
SAMLNK,Ø          /NR NEXT BLOCK, Ø=LAST BLOCK
        HLT       /ERROR RETURN, SYSTEM ERROR
        JMP I SAMRDB
/
/VARIABLES
/
SAMTMP,Ø
SAMIFN,Ø
SAMBKN,Ø
SAMMSK,Ø
SAMCNT,Ø
SAMBFA,BUFFER
SYSIO, 7642
/
/GENERAL CONSTANTS
/
C3,     3
C77,    77
C2ØØ,   2ØØ
C77ØØ,  77ØØ
M2ØØ,   -2ØØ
```

```
/039 SUBROUTINE READS OR WRITES ON DISK (TSS-8)
/
/BEFORE CALLING CALCULATE DISKADDRESS AND
/PUT IN HIOR AND LOWOR
/
/CALL    :JMS DFILE
/         FUNCTION (RFILE OR WFILE)
/         INTERNAL FILE NUMBER
/         -# WORDS
/         CORE ADDRESS
/         ERROR RETURN
/         NORMAL RETURN
/
/FILE MUST BE OPEN !!!!!!!!!
/
/
DFILE,   0
         TAD I DFILE       /GET FUNCTION
         DCA DFINST
         ISZ DFILE
         TAD I DFILE       /GET INT. FILE NR
         DCA W6BUF+1
         ISZ DFILE
DFTRY,   TAD I DFILE       /-#WORDS
         DCA W6BUF+2
         ISZ DFILE
         CLA CMA
         TAD I DFILE       /CORE ADDRESS
         DCA W6BUF+3
         ISZ DFILE
         TAD W6AD          /ADDRESS 6 WORD BUFFER
DFINST,  0                 /DO FUNCTION
         TAD W6BUF+5       /ERROR WORD
         SNA
         JMP DFOKE         /NO ERROR
         CLL RTR           /ERROR
         SZL SNA CLA
         SKP CLA           /ERROR=2
         JMP DFERR         /ERRORIS NOT 2
         IAC
         DCA W2BUF+2       /ERROR IS FILE FULL
         TAD W2AD          /SO MUST EXTEND FILE
         EXT               /EXTENDING WITH ONE SEGMENT
         SZA CLA
         JMP DFERR         /ERROR: DISK FULL
         TAD DFSEGA        /ADDRESS LIST SEGMENTCOUNTERS
         TAD W2BUF         /INT FILE NR
         DCA W6BUF+2       /TEMP USE
         ISZ I W6BUF+2     /INCREMENT COUNTER
         CLL CLA CMA RAL   /-2
         TAD DFILE
         DCA DFILE
         JMP DFTRY         /GO TRY AGAIN NOW
DFOKE,   ISZ DFILE
DFERR,   JMP I DFILE
/
/VARIABLES
/
DFSEGA,DFSEG0
DFSEG0,0            /# SEGMENTS FILE 0
```

```
DFSEG1,Ø            /# SEGMENTS FILE 1
DFSEG2,3            /# SEGMENTS FILE 2
DFSEG3,Ø            /# SEGMENTS FILE 3
W6AD,     W6BUF
W2AD,     W2BUF
W6BUF,
HIOR,     Ø         /HIGH ORDER DISK ADDRESS
W2BUF,    Ø         /INT FILE NR
          Ø         /-# WORDS;# SEG'S TO EXT
          Ø         /CORE ADDRESS-1
LOWOR,    Ø         /LOW ORDER DISK ADDRESS
          Ø         /ERROR WORD
```