Purdue University

Purdue e-Pubs

Department of Computer Science Technical Reports

Department of Computer Science

1975

Corrections to Some Errors in a Version of TSP: (Time Series Processor)

Robert E. Lynch
Purdue University, rel@cs.purdue.edu

Report Number:

75-153

Lynch, Robert E., "Corrections to Some Errors in a Version of TSP: (Time Series Processor)" (1975). *Department of Computer Science Technical Reports.* Paper 100.
https://docs.lib.purdue.edu/cstech/100

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries.
Please contact epubs@purdue.edu for additional information.

CORRECTIONS TO SOME ERRORS IN A VERSION OF TSP (TIME SERIES PROCESSOR)

Robert E. Lynch
Department of Computer Science
Purdue University
Lafayette, Indiana 47907

CSD TR 153

CORRECTIONS TO SOME SEVERE ERRORS IN A VERSION OF TSP
(Time Series Processor)

Robert E. Lynch
Department of Computer Sciences
Purdue University

1. Introduction. TSP (Time Series Processor) is a problem-oriented computer system designed to carry out automatically the computations which occur routinely in econometric research. It has some of the same procedures as SPSS and it has some which SPSS does not have. The simplicity of its language makes TSP suitable for users who have only a casual acquaintance with the use of a computer, for example, the TSP statement PLOTS$ generates informative line printer graphical displays.

Several faculty members at Purdue wanted TSP available for their research activities as well as for student use in courses. A version of TSP, written in IBM360/370 Fortran, was obtained and converted to CDC6500 Fortran.

During the conversion, results from test cases showed that there were some severe errors in the IBM version. Reasonable looking output was produced which is incorrect. No TSP error message was printed. We do not see how a user could tell that these results were incorrect unless he had correct results to compare with or unless he made a detailed examination of the output together with some hand calculation.

The purpose of this report is to bring these errors, and their corrections, to the attention of installations which support TSP.

2. Errors in TSP. Errors occur in the IBM Fortran version of the TSP subroutines PASS1, LOAD (three errors), GRAPH, and CAPITL. These errors involve the way that the TSP processor handles blank common.

If the IBM Fortran compiler handles double precision assignment statements in a way similar to that of the CDC6500 Fortran compiler, then there is an error in the TSP inner produce routine, INPROD.

Section 4 contains listings of the CDC6500 version of PASS1, LOAD, GRAPH, CAPITL, and INPROD. The locations of the errors in the IBM version, and their corrections, are indicated.

Parts of a user's TSP program statements and intermediate results are stored in blank common; blank common is also used for scratch space. Errors include the omission of an appropriate test to see how much scratch space is available as well as incorrect tests.

The errors in the test cases mentioned in Section 1 occurred because the processor used scratch space which overlapped the other part of blank common.

If [1] is used as the TSP manual for this IBM Fortran version of TSP, then there is a misprint on page 3 of [1]. In the display of the size limitation, NOBS*VARS should be replaced with 301 + NOBS*(2*VARS + 1) + 2*VARS.

Remark: The TSP manual [1] does not mention the useful TSP feature which allows a comment to be inserted into the first statement, $$NAME program name$, of a TSP program. The processor stores the name immediately following NAME and everything else up to the terminator, $, is ignored. Consequently, instead of using something like $$NAME CASE1$, the user may give a more detailed program identification, for example:

```
    $$NAME CASE1, GROSS NATIONAL PRODUCT MODELS OF I.M. JONES.  JUNE 19, 1984.
                    MODEL 1 GNP = CONSTANT + A*EXP(TIME)
                    MODEL 2 GNP = CONSTANT + A*EXP(TIME) + B*EXP(2*TIME) $
```

Any name in the identification, such as CASE1, GROSS, etc., may be used as a variable name in the program. The only limitation is a maximum of 8 characters for a name; the use of SEPTEMBER would result in the generation of the TSP error message "name has too many characters."

3. Some differences between the IBM and CDC Fortran versions. In order to locate and correct the errors, detailed knowledge was required about the workings and functions of about half the 119 TSP subroutine which we had. As the purposes of various subroutines became clear, numerous comments were inserted. Consequently, the CDC6500 version is better documented than the IBM Fortran version, which is almost devoid of comments.

A number of inefficiencies were eliminated. Most of the linear searches of lists were replaced by more efficient search techniques.

In the CDC6500 version, a comment can be inserted at any place that a blank or comma separator is required or can occur. /* in columns 1 and 2 of a card denote that columns 3 to 80 contain a comment.

The user can use longer than 8 character names in order to help document his TSP program, for example

```
GENR NEWCASHBALANCE = OLDCASHBALANCE*( 1 + (INTERESTRATE/100)*(MONTHS/12) )$
```

However, the processor only uses the first 8 characters internally and it prints a record of the more-than-8-character names so the user can check for uniqueness.

The input routine, INPT, was completely rewritten. The new version is much more efficient than the older one. Free format data input takes only about 30% longer than formatted data input. The new version examines each non-blank, non-comma character about twice instead of the up to 20 times of the old version.

The user has more flexibility in the choice of storage requirements. He may choose blank common sizes of 3K, 5K, 10K, 15K, 20K, 25K. The 3K size is suitable for student jobs and the 25K size is capable of doing TSP programs which require the 30K IBM Fortran version. At the end of the execution output of each TSP program, the amount of blank common which was used by the program is printed. With this information, the user can, perhaps, decrease the blank common size request on subsequent runs in order to decrease turnaround time and cost.

The most significant improvement in efficiency involves the number of words (NWORD) used to store a TSP keyword or user defined variable name (this improvement is directly transferable to the IBM Fortran version). The IBM version uses NWORD = 2 and two 8-byte words are used for keywords and names; floating point numerical values are stored in single 8-byte words. When a name is moved, two words must be moved and when a comparison or search of a list is done, pairs of comparisons are made (there is a lot of moving and comparisons of names during the execution of a TSP program). The CDC version uses NWORD = 1 and a keyword or name is stored in a single 60-bit word. This change to NWORD = 1 accomplishes the following:

1. Execution time for the movement and comparison of names is decreased by more than a factor of 2.
2. Numerous calls of short subroutines and functions are replaced by single assignment statements or by single IF statements.
3. Memory requirement is reduced (on the CDC6500).

Remark 1: If the change to NWORD = 1 were made in the IBM version, then the increased efficiencies of 1 and 2 are achieved by declaring names to be REAL DOUBLE PRECISION (or whatever the IBM Fortran declaration is). Accuracy is improved by using REAL DOUBLE PRECISION declarations for numerical values so that 16-bytes are used for floating point values; this improvement is made at the sacrifice of memory space. It is well-known that one must use as much precision as economically feasible in matrix computations--especially in least squares computations. Moreover, in some cases, users might not know the importance of the choice of a basis for least squares computations and, perhaps, try to do a regression with a polynomial approximator of the form $a_0 + a_1x + \ldots + a_nx^n$; use of low precision numerical values might result in meaningless results. Thus, the change to NWORD = 1 should also be accompanied by REAL DOUBLE PRECISION declaration for just about all the variables in the TSP IBM Fortran program.

Remark 2: The CDC version uses NWORD = 1 and all changes from the IBM version are noted with comments. To change the IBM version to NWORD = 1, one can make all the appropriate changes Indicated on the complete listing of the CDC version plus Inserting the required REAL DOUBLE PRECISION declarations (there are a lot of these).

4. Listing of PASS1, LOAD, GRAPH, CAPITL, and INPROD. Listing of the CDC versions of the subroutines in which we found errors follows. On these listings, changes from the IBM Fortran version to CDC Fortran version are bracketed by a pair of comments: CIBMTOCDC CCDCTOIBM. Locations of errors in the IBM code and their corrections are bracketed with pairs of lines of dollar signs; notes about efficiencies are also bracketed. The corrections to the errors are CDC independent.

Since we have not seen machine language code generated by the IBM Fortran compiler, we are not sure if there is an error in the inner product routine. The original IBM program contains

```
      DOUBLE PRECISION XPROD
      J = 1
      XPROD = 0.
      DO 100 I = NN, JSA
      XPROD = XPROD + A(I)*B(J)
  100 J = J + JSB
```

The CDC compiler generates code which computes the single precision product A(I)*B(J), then makes it double precision with zero lower half, and finally adds this to the double precision XPROD.

One way to carry out this calculation correctly on the CDC6500 is to introduce two more double precision variables, TEMPA, TEMPB, and to replace the DO loop with

```
      DO 100 I = NN, JSA
      TEMPA = A(I)
      TEMPB = B(J)
      XPROD = XPROD + TEMPA*TEMPB
  100 J = J + JSB
```

Remark: Some of the DATA statement on the listings use the CDC character string specification R, for example DATA NAMES/ 8RNOPRINT   , .../. This causes the BCD representation of the character string to be stored right justified with zero left fill; these can be used as variables declared INTEGER in arithmetic and logical expressions.

```
CIBMT0CDC
      SUBRØUTINE PASS1
CCDCT0IBM
C
C     THIS PRØGRAM SUPERVISES READING AND PARSING THE INPUT
CIBMT0CDC
C
C     REPLACED  JX(400) (NWØRD = 1), ADDED LIMTYP
      CØMMØN
     *        TYPE(200), JX(200), CØDE(600), IX(200), ITYPE(200)
      INTEGER TYPE
      DATA LIMTYP /200 /
C
      DIMENSIØN X(1)
      EQUIVALENCE (X(1),JX(1))
C
C
      CØMMØN /TSPCØM/
     * MEMSIZ, NØB   , NSPARG, NWØRD , LENGTH,
     * NTYPE , IFDBUG, IFTITL, NCHAR , NSUP  ,
     * MEMST , NØREG , IFPLØT, IFFAST, NPAGE ,
     * NUMLIN, IFREPL, PRØFF , SKIP(11), JPHAS ,
     * LIMARG, LINE  , NJARG , NARG  , NAME ,
     *  JARG(4)
C             LENGTH ØF JARG SET IN MAIN ØVERLAY
C
C     DELETED  NAME2  BETWEEN  NAME  AND  JARG  IN  /TSPCØM/ (NWØRD = 1)
C
      LØGICAL IFDBUG, IFTITL, IFPLØT, IFFAST,
     *        IFREPL, PRØFF
C
C
C     NEW CØMMØN BLØCK ADDED
      CØMMØN /MEMCØM/ IMNSZ(7), USRNAM, PSSVL(20)
      INTEGER USRNAM
      EQUIVALENCE
     * (IMNSZ(1),MMMSIZ), (IMNSZ(2),LLMARG), (IMNSZ(3),LLMDØT),
     * (IMNSZ(4),LLMBUF), (IMNSZ(5),LLMSYM), (IMNSZ(6),LLMSMP),
     * (IMNSZ(7),LLMØUT)
C
C     IMMSZ USED TØ KEEP TRACK ØF MEMØRY USE
      DIMENSIØN IPSSVL(20)
      EQUIVALENCE (IPSSVL(1),PSSVL(1))
C     PSSVL AND IPSSVL USED TØ PASS VARIABLES BETWEEN SUBRØUTINES
C     IN DIFFERENT ØVERLAYS
C     IPSSVL(1) IS USED TØ KEEP RECØRD ØF USE ØF CØMMØN IN LØAD
C        AND IN GENR
C     IPSSVL(3) PASSES CØMPUTED GØ TØ INDEX FRØM EXEC TØ MATRIX
C     IPSSVL(2) PASSES CØMPUTED GØ TØ INDEX FRØM EXEC TØ ØVERLAYS
C     IPSSVL(4) IS USED TØ KEEP RECØRD ØF USE ØF JARG-CØMMØN
C     IPSSVL(5) IS USED TØ KEEP RECØRD ØF USE ØF DØT-CØMMØN
C     IPSSVL(6) IS USED TØ KEEP RECØRD ØF USE ØF BUFFER-CØMMØN
C     IPSSVL(7) IS USED TØ KEEP RECØRD ØF USE ØF JSML-CØMMØN
C     IPSSVL(8) IS USED TØ KEEP RECØRD OF USE ØF ØUTBUF-CØMMMØN
      EQUIVALENCE
     * (LØDUSE, IPSSVL(1)), (JTØØVL, IPSSVL(2)), (JTØMAT, IPSSVL(3))
     *,(JARUSE, IPSSVL(4)), (DØTUSE, IPSSVL(5)), (BUFUSE,IPSSVL(6)),
     * (SMPUSE, IPSSVL(7)), (ØBFUSE, IPSSVL(8))
      INTEGER DØTUSE, BUFUSE, SMPUSE, ØBFUSE
C
C     CHANGED DIMENSIØN ØF  NAMES  TØ 5 FRØM 20
C     CHANGED DIMENSIØN ØF NNAMES  TØ 4 FRØM 20
      DIMENSIØN NAMED(2), JCØDE(5), NNAMES(4), NAMES(5), NNAME(2)
CCDCT0IBM
      LØGICAL LAGGED, IFRPAR
CIBMT0CDC
```

```
  C
  C      NEXT ARE ØPERATØR CØDES FØR TSP FUNCTIØNS                          PASS1
  C         11 EXP, 12 ABS, 13 LØG AND ALØG                                 PASS1
  CCDCTØIBM                                                                 PASS1
        DATA JCØDE /0,11,12,13,13/                                          PASS1
  CIBMTØCDC                                                                 PASS1
  C                                                                         PASS1
        DATA G / 1RG /                                                      PASS1
  C                                                                         PASS1
  C                                                                         PASS1
        DATA NNAMES /                                                       PASS1
       * 8REXP     , 8RABS      , 8RALØG     , 8RLØG        /               PASS1
  C                                                                         PASS1
        DATA NAMES /                                                        PASS1
       * 8REND     , 8RGENR     , 8RNAME     , 8RBCD     , 8REBCDIC   /     PASS1
  C                                                                         PASS1
  C / / / / / / / / / / / / / / / / / / / / / / BEGINNING ØF PASS1 / / / /  PASS1
  CCDCTØIBM                                                                 PASS1
        LAGGED = .FALSE.                                                    PASS1
        IFRPAR = .FALSE.                                                    PASS1
  C                                                                         PASS1
  CIBMTØCDC                                                                 PASS1
  C                                                                         PASS1
  C      REPLACED 37 WITH 10 TØ PRINT THE WØRD  LINE  ØN TSP PRØGRAM ØUTPUTPASS1
      1 CALL ØUTEDT( 10, 1, LINE+1 )                                        PASS1
  C                                                                         PASS1
  C      READ NEXT TSP STATEMENT                                            PASS1
  C                                                                         PASS1
  CCDCTØIBM                                                                 PASS1
        CALL INPT(JX,N,TYPE)                                                PASS1
  CIBMTØCDC                                                                 PASS1
  C                                                                         PASS1
  C      NAMES, NUMBERS, ØPERATØRS ØN TSP STATEMENT ARE NØW STØRED          PASS1
  C         IN SUCCESSIVE WØRDS IN ARRAY  JX(I), I=1,N  IN THE ØRDER THAT   PASS1
  C         THEY APPEAR IN THE TSP STATEMENT                                PASS1
  C         STATEMENT KEYWØRD IS IN  JX(1)                                  PASS1
  C         TYPE ØF ELEMENT IN JX(I) IS GIVEN BY VALUE ØF TYPE(I)           PASS1
  C            1  NUMERICAL                                                 PASS1
  C            2  ØPERATØR                                                  PASS1
  C            3  NAME                                                      PASS1
  C            4  DØTNAME                                                   PASS1
  C                                                                         PASS1
  C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$ PASS1
  C      MAJØR ERRØR IN IBM VERSIØN.  INPT CAN DESTRØY DATA IN              PASS1
  C      BLANK CØMMØN SUCH AS PRØGRAM STATEMENTS, DATA, VALUES              PASS1
  C      GENERATED WITH GENR                                                PASS1
  C      A TEST WAS ADDED AT RETURN ØF CALL ØF INPUT                        PASS1
        IF( N .LE. LIMTYP )                                                 PASS1
  C        THEN ARRAYS ARE LARGE ENØUGH FØR INPUT                           PASS1
       *                                                                    PASS1
  C        ELSE TØØ MUCH DATA                                 GØ TØ 10040PASS1
           IF( LIMTYP + N .LT. MEMSIZ )                                     PASS1
  C           THEN HAVE NØT DESTRØYED STØRED VALUES IN BLANK CØMMØN         PASS1
       *                                                                    PASS1
  C           ELSE PART ØF BLANK CØMMØN ØVER-WRITTEN          GØ TØ 10020PASS1
               IRUIN = LIMTYP + N - MEMSIZ                                  PASS1
               WRITE(6,10010) IRUIN                                         PASS1
10010          FØRMAT( 38H0*** ERRØR.  DURING READ ØF STATEMENTS  ,         PASS1
       *         16H DESTRØYED LAST , I10, 14H WØRDS ØF DATA  ,             PASS1
       *         28H IN BLANK CØMMØN.  ABØRT RUN  )                         PASS1
C                                                             CALL ABØRT PASS1
10020          CØNTINUE                                                     PASS1
        WRITE( 6, 10030)                                                    PASS1
10030   FØRMAT( 31H0*** WARNING.  READ IN TØØ MUCH  ,                       PASS1
       *   27H DATA FØR ARRAYS IN PASS1.  ,                                 PASS1
       *   30H ØPERATIØN CØNTINUES CØRRECTL                                 PASS1
```

```
     *         30H IF THIS FØLLØWS AN *** ERRØR   ,
     *         32H MESSAGE.  ØTHERWISE, SUBSEQUENT  ,
     *         25H RESULTS ARE LIKELY TØ BE  ,  10H INCØRRECT )
10040     CØNTINUE
C       END ØF CØRRECTIØN
C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C
C
C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C       SLIGHT IMPRØVEMENT IN EFFICIENCY ØF SEARCH
C       REPLACED  JJ = IUCØMP(JX, NAMES)
C                 GØ TØ (10, 100, 200, 300, 400, 410) JJ
C       WITH THE FØLLØWING
C
C       THE FØLLØWING USES THE CØLLATING SEQUENCE ØF THE CDC6500
C       THE 8-BCD CHARACTER WØRDS IN NAMES(I) SATISFY
C       CHAR. =  12345678    12345678    12345678    12345678    12345678
C                BCD      . LT. EBCDIC  . LT. END    . LT. GENR    . LT. NAME
C       I=        4            5            1            2            3
C
C       DETERMINE WHETHER TSP KEYWØRD IS ØNE ØF
C          BCD, EBCDIC, END, GENR, NAME
C
C       TEST FØR KEYWØRD END
        IF( JX(1) - NAMES(1) )                  3, 100, 4
C          TEST FØR KEYWØRD EBCDIC
    3      IF( JX(1) - NAMES(5) )               5, 410, 10
C          TEST FØR KEYWØRD GENR
    4      IF( JX(1) - NAMES(2) )              10, 200, 6
C             TEST FØR KEYWØRD BCD
    5         IF( JX(1) - NAMES(4) )           10, 400, 10
C             TEST FØR KEYWØRD NAME
    6         IF( JX(1) - NAMES(3) )           10, 300, 10
C       END ØF REPLACEMENT
C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C
C       KEYWØRD IS NØT ØNE ØF  BCD, EBCDIC, END, GENR, NAME
C
CCDCTØIBM
   10 NARG = N - 1
      IF (TYPE(1) .NE. 3) CALL ERG(1, JX)
CIBMTØCDC
C
C       MØVE KEYWØRD TØ NAME STØRAGE
C
C       REPLACED CALL NMØV(NAME, JX(1)) WITH FØLLØWING (NWØRD = 1)
        NAME = JX(1)
   35 CØNTINUE
CCDCTØIBM
        IF (NARG .EQ. 0) GØ TØ 50
CIBMTØCDC
C
C       REPLACED  KK = NWØRD + 1  WITH THE FØLLØWING  (NWØRD = 1)
        KK = 2
C
C       PUT ARGUMENTS ØF TSP STATEMENT INTØ TEMPØRARY STØRAGE IN JARG
C
CCDCTØIBM
        DØ 40 I=2, N
        CALL ARGPUT(I-1, JX(KK), TYPE(I), 0)
        KK = KK + 1
CIBMTØCDC
C
C       DELETED THE FØLLØWING   (NWØPD = 1)
C          IF(TYPE(I) .EQ  3  ØR. TYPE(I) .EQ. 4) KK = KK + NWØPD - 1
CCDCTØIBM
   40 CØNTINUE
```

```
CIBMTØCDC
C
   50 CØNTINUE
C
C      FRØM STATEMENT FØLLØWING 35 ØR 225
C
      LINE = LINE + 1
C
C      HAVE NAMES AND ARGUMENTS ØF CØMPLETE TSP STATEMENT STØRED
C      IN  NAME, JARG(. ).  MØVE THEM TØ BLANK CØMMØN
C
CCDCTØIBM
      CALL LINPUT
      GØ TØ 1
C
CIBMTØCDC
C      HAVE TSP KEYWØRD  *****END*****
C      FINISHED READING TSP PRØGRAM.   STØRE END STATEMENT AND
C      RETURN TØ SUPER TØ EXECUTE TSP PRØGRAM
C
C      REPLACED CALL NMØV(NAME, JX(1)) WITH FØLLØWING (NWØRD = 1)
  100 NAME = JX(1)
CCDCTØIBM
      NARG=0
      LINE=LINE+1
      CALL LINPUT
      CALL ØUTPT
CIBMTØCDC
C
C      NEXT IS ENTRY TØ SUPER
CCDCTØIBM
      CALL RETURN
CIBMTØCDC
C      NØ RETURN TØ THIS PØINT FRØM ENTRY RETURN ØF SUPER
CCDCTØIBM
C
CIBMTØCDC
C      HAVE TSP KEYWØRD  *****GENR*****
C      MØVE KEYWØRD TØ NAME STØRAGE
C
C      REPLACED CALL NMØV(NAME, JX(1)) WITH FØLLØWING (NWØRD = 1)
  200 NAME = JX(1)
C
C      CØNSTRUCT GENR-LINE-IDENTIFIER AND STØRE IN JARG(1)
CCDCTØIBM
      CALL INVNT(NNAME, G, LINE)
      CALL ARGPUT(1, NNAME, 6, 0)
CIBMTØCDC
C
C      PUT LEFT SIDE ØF = INTØ TEMPØRARY STØRAGE IN JARG
C
C      REPLACED  JX(NWØRD + 1)  WITH  JX(2)   (NWØRD = 1)
      CALL ARGPUT( 2, JX(2), TYPE(2), 0 )
CCDCTØIBM
C
C      JJ IS LØCATIØN IN IX.
C      KK IS LØCATIØN IN JX.
C      LL IS LØCATIØN IN JARG.
C
      IX(1) = 9
      ITYPE(1) = 2
      JJ = 2
      LL = 3
CIBMTØCDC
C
C      REPLACED  KK = NWØRD*2 + 2  WITH FØLLØWING   (NWØRD = 1)
C      THE USE ØF KK = 4 (ØR 6 IF NWØRD=2) PICKS UP FIRST ITEM ØN RIGHT
```

```
C         SIDE ØF = IN GENR STATEMENT
      KK = 4
CCDCTØIBM
C
      DØ 220 I=4,N
      NTYPE = TYPE(I)
      IF(LAGGED)  GØ TØ 530
CIBMTØCDC
C     TEST FØR NAME ØR NUMBER
CCDCTØIBM
      IF (NTYPE .NE. 2) GØ TØ 205
CIBMTØCDC
C     TEST FØR LEFT PAREN
CCDCTØIBM
      IF(JX(KK) .EQ. 9)  GØ TØ 500
C
C     ØPERATØR
C
  201 MMM = JX(KK)
CIBMTØCDC
C     TEST TØ SEE IF MINUS (CØDE 3) SHØULD BE CHANGED TØ CØDE 8
C     (9 = LEFT PAREN, 2 = ØPERATØR).  SWITCH MADE IF - IS PRECEDED
C     BY (   CØDE 3 MINUS IS INFIX, CØDE 8 MINUS IS PREFIX
CCDCTØIBM
      IF (MMM .EQ. 3 .AND. IX(JJ-1) .EQ. 9 .AND. ITYPE(JJ-1) .EQ. 2)
     I    MMM = 8
      IX(JJ) = MMM
      ITYPE(JJ) = 2
      JJ = JJ + 1
      KK = KK + 1
      GØ TØ 220
CIBMTØCDC
C     HAVE LEFT PAREN.  SEE IF AN ØPERATØR (CØDE 2) PRECEDES IT.
CCDCTØIBM
  500 IF (ITYPE(JJ-1) .EQ. 2) GØ TØ 201
CIBMTØCDC
C     LEFT PAREN PRECEDED BY NAME, GET NAME AND SEE IF IT IS
C     TSP FUNCTIØN
CCDCTØIBM
      LØØKUP = LL - 1
      CALL ARGGET(LØØKUP,NAMED,JTYPE,LAG)
CIBMTØCDC
C
C     REPLACED  MATCH = IUCØMP(NAMED,NNAMES)  WITH THE FØLLØWING
C
C     DETERMINE IF NAMED(1) IS ØNE ØF TSP  FUNCTIØNS EXP,ABS,LØG,ALØG
      MATCH = 1
      IF( NAMED(1) .EQ. NNAMES(1) ) MATCH = 2
      IF( NAMED(1) .EQ. NNAMES(2) ) MATCH = 3
      IF( NAMED(1) .EQ. NNAMES(3) ) MATCH = 4
      IF( NAMED(1) .EQ. NNAMES(4) ) MATCH = 5
C     END ØF REPLACEMENT
CCDCTØIBM
      IF(MATCH .EQ. 1)  GØ TØ 525
C
CIBMTØCDC
C     HAVE ØNE ØF THE FUNCTIØNS  ABS,EXP,ALØG,LØG
C     INSERT ITS CØDE AND SET TYPE TØ ØPERATØR (2)
CCDCTØIBM
C
  505 LL = LØØKUP
      IX(JJ-1) = JCØDF(MATCH)
      ITYPE(JJ-1) = 2
      GØ TØ 201
C
CIBMTØCDC
C     HAVE NØNE ØF THE FUNCTIØNS  ABS,EXP,ALØG,LØG
```

```
C
C     LEFT PAREN PRECEDED BY NØN-TSP-FUNCTIØN NAME, HAVE LAGGED
C     VARIABLE
CCDCTØIBM
C
  525 LAGGED = . TRUE.
      KK = KK + 1
      GØ TØ 220
  530 IF(NTYPE . NE. 1)  GØ TØ 540
CIBMTØCDC
C
C     PRØCESSING SIGNED INTEGER ARGUMENT ØF LAGGED VARIABLE, STØRED
C     AS REAL TYPE, CØNVERT IT TØ INTEGER TYPE
CCDCTØIBM
      MMM = X(KK)
      CALL ARGPUT(LØØKUP, NAMED, JTYPE, MMM)
      IFRPAR = . TRUE.
      KK = KK + 1
      GØ TØ 220
CIBMTØCDC
C
C     TURN ØF LAGGED-SWITCH WHEN GET TØ RIGHT PAREN
CCDCTØIBM
  540 IF(. NØT. IFRPAR)   CALL ERG(69, JX(KK))
      IF(NTYPE . NE. 2 . ØR. JX(KK) . NE. 1)   CALL ERG(69, JX(KK))
      IFRPAR = . FALSE.
      LAGGED = . FALSE.
      KK = KK + 1
      GØ TØ 220
CIBMTØCDC
C
C     PUT NAME ØR NUMBER INTØ JARG( )
CCDCTØIBM
  205 CALL ARGPUT(LL, JX(KK), NTYPE, 0)
      IF(NTYPE. NE. 1. ØR. (ITYPE(JJ-1). NE. -1. AND. (ITYPE(JJ-1). NE. 2. ØR. IX(JJ
     1-1). NE. 1))) GØ TØ 210
      IX(JJ) = 2
      ITYPE(JJ) = 2
      JJ = JJ + 1
  210 IX(JJ) = LL - 2
      ITYPE(JJ) = -1
      JJ = JJ + 1
      KK = KK + 1
CIBMTØCDC
C
C     DELETED THE FØLLØWING   (NWØRD = 1)
C        IF(NTYPE . EQ. 3 . ØR. NTYPE . EQ. 4) KK = KK + NWØRD - 1
CCDCTØIBM
      LL = LL + 1
  220 CØNTINUE
CIBMTØCDC
C
C     HAVE FINISHED STØRING CØDED GENR STATEMENT, CØNVERT IT TØ
C     STRING WHICH EVAL CAN PRØCESS
CCDCTØIBM
C
      IX(JJ) = 1
      ITYPE(JJ) = 2
      JJ = JJ + 1
      IX(JJ) = 14
      ITYPE(JJ) = 2
      CALL CØMPLR(IX, ITYPE, CØDE, LENGTH, CØDE(501), CØDE(551))
CIBMTØCDC
C
C     STØRE STRING IN BLANK CØMMØN
CCDCTØIBM
      CALL VPUT(NNAME, CØDE, LENGTH)
      NARG = LL - 1
CIBMTØCDC
  225 CØNTINUE
CCDCTØIBM
      GØ TØ 50
C
CIBMTØCDC
  300 CØNTINUE
C     HAVE TSP KEYWØRD  ******NAME******
C     STØRE THE  CASE  PART ØF TSP STATEMENT $$NAME,CASE$
C     IN USRNAM  FØR USE IN END-ØF-EXECUTIØN ØUTPUT
CCDCTØIBM
C
CIBMTØCDC
C     DELETED CALL NMØV(JARG(1),X)   (NWØRD+1)
C
C     REPLACED CALL ØF ØNE LINE SUBRØUTINE
C     UNAME    CALL NMØV(...,JARG)    RETURN    END
C     WITH THE FØLLØWING   (NWØRD = 1)
      USRNAM = JX(2)
CCDCTØIBM
      GØ TØ 1
C
C     SET UP CHARACTER CØDE
CIBMTØCDC
C
  400 CØNTINUE
C     FØLLØWING
C 400 CALL BCD
C     GØ TØ 1
C 410 CALL EBCDIC
C     GØ TØ 1
C     REPLACED WITH
C
C     HAVE TSP KEYWØRD  *****BCD*****
C
      JUMP = 1
                                                 GØ TØ 415
C
  410 CØNTINUE
C     HAVE TSP KEYWØRD  *****EBCDIC*****
C
      JUMP = 2
  415 CØNTINUE
      WRITE(6,420)
  420 FØRMAT(44HO TSP STATEMENTS  BCD  AND  EBCDIC  DISABLED , /
     * 53H  CDC6500 READS BCD.  IF SØURCE TSP DECK IS IN EBCDIC,
     * 47H IT SHØULD BE PREPRØCESSED TØ CØNVERT IT TØ BCD     )
                                                 GØ TØ (1,430), JUMP
  430 CØNTINUE
      STØP
CCDCTØIBM
      END
05.23.17.J0 16 EP30       .   8 FEET
```

```
CIBMTØCDC
      SUBRØUTINE LØAD
CCDCTØIBM
C
C      THIS IS A STRICTLY TEMPØRARY VERSIØN BASED ØN THE ØLD 1620/7094 L
C
CIBMTØCDC
C
      CØMMØN /TSPCØM/
     * MEMSIZ, NØB   , NSPARG, NWØRD , LENGTH,
     * NTYPE , IFDBUG, IFTITL, NCHAR , NSUP  ,
     * MEMST , NØREG , IFPLØT, IFFAST, NPAGE ,
     * NUMLIN, IFREPL, PRØFF , SKIP(11), JPHAS ,
     * LIMARG, LINE  , NJARG , NARG  , NAME ,
     *  JARG(4)
C             LENGTH ØF JARG SET IN MAIN ØVERLAY
C
C     DELETED  NAME2  BETWEEN  NAME  AND  JARG  IN  /TSPCØM/ (NWØRD = 1)
C
      LØGICAL IFDBUG, IFTITL, IFPLØT, IFFAST,
     *          IFREPL, PRØFF
C
C
C     NEW CØMMØN BLØCK ADDED
      CØMMØN /MEMCØM/ IMNSZ(7), USRNAM, PSSVL(20)
      INTEGER USRNAM
      EQUIVALENCE
     * (IMNSZ(1),MMMSIZ), (IMNSZ(2),LLMARG), (IMNSZ(3),LLMDØT),
     * (IMNSZ(4),LLMBUF), (IMNSZ(5),LLMSYM), (IMNSZ(6),LLMSMP),
     * (IMNSZ(7),LLMØUT)
C
C     IMMSZ USED TØ KEEP TRACK ØF MEMØRY USE
      DIMENSIØN IPSSVL(20)
      EQUIVALENCE (IPSSVL(1),PSSVL(1))
C     PSSVL AND IPSSVL USED TØ PASS VARIABLES BETWEEN SUBRØUTINES
C     IN DIFFERENT ØVERLAYS
C     IPSSVL(1) IS USED TØ KEEP RECØRD ØF USE ØF CØMMØN IN LØAD
C        AND IN GENR
C     IPSSVL(3) PASSES CØMPUTED GØ TØ INDEX FRØM EXEC TØ MATRIX
C     IPSSVL(2) PASSES CØMPUTED GØ TØ INDEX FRØM EXEC TØ ØVERLAYS
C     IPSSVL(4) IS USED TØ KEEP RECØRD ØF USE ØF JARG-CØMMØN
C     IPSSVL(5) IS USED TØ KEEP RECØRD ØF USE ØF DØT-CØMMØN
C     IPSSVL(6) IS USED TØ KEEP RECØRD ØF USE ØF BUFFER-CØMMØN
C     IPSSVL(7) IS USED TØ KEEP RECØRD ØF USE ØF JSML-CØMMØN
C     IPSSVL(8) IS USED TØ KEEP RECØRD ØF USE ØF ØUTBUF-CØMMØN
      EQUIVALENCE
     * (LØDUSE, IPSSVL(1)), (JTØØVL, IPSSVL(2)), (JTØMAT, IPSSVL(3))
     *,(JARUSE, IPSSVL(4)), (DØTUSE, IPSSVL(5)), (BUFUSE,IPSSVL(6)),
     * (SMPUSE, IPSSVL(7)), (ØBFUSE, IPSSVL(8))
      INTEGER DØTUSE, BUFUSE, SMPUSE, ØBFUSE
C
C
C     REMØVED /SMPCØM/
C
C     ADDED  JX(1)  EQUIVALENCED TØ  X(1)
      DIMENSIØN JX(1)
      EQUIVALENCE ( JX(1), X(1) )
C
CCDCTØIBM
      INTEGER TYPE
CIBMTØCDC
C
      CØMMØN
     * TYPE(100), MASK(200), X(1)
C
      INTEGER FRØNT, [illegible]
      DATA    FRØNT / 300 /, LIMTYP / 100 /
C             FRØNT IS LENGTH ØF ARRAYS TYPE PLUS MASK
C
C     DELETED LØGICAL IFCØMP
C
C
C     CHANGE DIMENSIØN ØF NAMES TØ6 FRØM 20
      DIMENSIØN NAMES(6)
C
      DATA NAMES /
     * 8RLØAD    , 8REND     , 8RMASK    , 8RSMPL    , 8RNØPRINT ,
     * 8RFØRMAT    /
C
C / / / / / / / / / / / / / / / / / / / / / / / / / BEGINNING ØF LØAD / / / /
CCDCTØIBM
      PRØFF=.F ALSE.
      CALL ØUTPT
      NMASK = 2
      MASK(1) = 1
      MASK(2) = 80
C
CIBMTØCDC
C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C     MAJØR ERRØR IN IBM VERSIØN.   INPT CAN DESTRØY DATA IN BLANK
C     CØMMØN.
C     INSERTED TEST AT RETURN FRØM INPT.
C     REPLACED  1 CALL XNPT(X,N,TYPE)
C                 IF(TYPE(1).EQ.1)  GØ TØ 1
C     WITH THE FØLLØWING
    1 CØNTINUE
C
C     SEARCH FØR TSP STATEMENT KEYWØRD ØR ØPERATØR
      NMLEFT = MEMSIZ - FRØNT
    2 CØNTINUE
      CALL INPT( X, N, TYPE )
      IF( N .LT. NMLEFT )
C       THEN HAVE ENØUGH RØØM
     *
C       ELSE HAVE STØRED ØVER DATA IN BLANK CØMMØN              GØ TØ 10004
          RUINED = N - NMLEFT
          WRITE( 6, 10003 ) RUINED
10003     FØRMAT( 50HØ*** ERRØR.  DURING LØAD SECTIØN, DESTRØYED LAST ,
     *       I5, 30H WØRDS ØF DATA IN BLANK CØMMØN  )

C                                                                CALL ABORT
10004     CØNTINUE
      IF( TYPE(1) .EQ.  1 )
C       THEN CØNTINUE READING INPUT
     *
C       ELSE HAVE FØUND A NØN-NUMERIC INPUT VALUE               GØ TØ 2
C
C     END ØF CØRRECTIØN
C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C
C
C
C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C     INSERTED SLIGHTLY MØRE EFFICIENT SEARCH
C     REPLACED 2 JJ = IUCØMP(X,NAMES)
C                GØ TØ (10,100,200,300,400,500,10), JJ
C     NITH THE FØLLØWING
C
C     THE FØLLØWING USES THE CØLLATING SEQUENCE ØF THE CDC6500
C     THE 8-BCD CHARACTER WORDS IN NAMES(I) SATISFY THE FØLLØWING
C     CHAR. = 12345678   12345678   12345678   12345678   12345678
C             END     .LT. LØAD   .LT. MASK   .LT. NØPRINT  LT. SMPL
```

```
C
C     TEST FØR MASK
      IF( JX(1) - NAMES(3) )                         3,300, 4
C        TEST FØR LØAD
    3    IF( JX(1) - NAMES(1) )                      5,100,10
C        TEST FØR NØPRINT
    4    IF( JX(1) - NAMES(5) )                     10,500, 6
C           TEST FØR END
    5       IF( JX(1) - NAMES(2) )                  10,200,10
C           TEST FØR SMPL
    6       IF( JX(1) - NAMES(4) )                  10,400,10
C     END ØF REPLACEMENT
C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C
C     NØNE ØF THE KEYWØRDS WERE FØUND
CCDCTØIBM
   10 CALL ERG(11,X)
CIBMTØCDC
C     NØ RETURN TØ THIS PØINT FRØM ERG
CCDCTØIBM
C
CIBMTØCDC
C     HAVE TSP KEYWØRD. ******LØAD******
C     HAVE JUST READ INPUT, CHECK TØ SEE IF ALL ARE VARIABLE NAMES
CCDCTØIBM
C
  100 DØ 110 I=2,N
  110 IF(TYPE(I) .NE. 3)  CALL ERG(12,X)
      NØVAR = N - 1
CIBMTØCDC
C     NØVAR IS THE NUMBER ØF VARIABLE NAMES JUST READ
C
C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C     MAJØR ERRØR IN IBM VERSIØN.  INPT CAN READ IN MØRE VARIABLE NAMES
C     THAN ALLØWED FØR BY THE SIZE ØF ARRAY TYPE.
      IF( NØVAR .LT. LIMTYP )
C       THEN THE ARRAYS ARE LARGE ENØUGH
     *                                                         GØ TØ 116
C       ELSE TØØ MANY VARIABLE NAMES
          WRITE( 6, 115 ) NØVAR, LIMTYP
  115     FØRMAT( 20HØ*** ERRØR.  LØADED  , I5,
     *       29H VARIABLE NAMES.  MAXIMUM IS  ,  I5,
     *       17H PER CALL ØF LØAD   )
          CALL ERG( 20, NØVAR )
C         NØ RETURN TØ THIS PØINTFRØM ERG
C
  116     CONTINUE
C
C     END ØF CØRRECTIØN
C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C
C
C
C     REPLACED  NLBUF=NWØRD*N+1  WITH  (NWØRD = 1)
      NLBUF = N + 1
CCDCTØIBM
      NLDAT = NLBUF + NØB
CIBMTØCDC
C
C     MØVED NEXT STATEMENT UP A FEW LINES TØ CUT ØUT SØME MULTIPLICATIØN
      NDATA = NØB*NØVAR
C
C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C     MAJØR ERRØR IN IBM VERSIØN
C     NTEST USED TØ TEST THE SIZE ØF AVAILABLE BLANK CØMMØN
C     SPACE CONTAINS ERRØR.
C
C     REPLACED  NTEST = NLDAT + NØB*NØVAR  WITH
```

```
      NTEST = FRØNT + N + 1 + NØB + 2*NDATA                             LØAD
C       BLANK CØMMØN USE                                                LØAD
C          ARRAYS TYPE, MASK   = FRØNT                                  LØAD
C          INPUT IN X          = N                                      LØAD
C          TEMPØRARY           = NØB                                    LØAD
C          WILL READ           = NØB*NØVAR + 1  (THE PLUS 1 IS FØR THE  LØAD
C                                          TYPE ØF LAST DATUM READ)LØAD
C          WILL STØRE          = NØB*NØVAR                              LØAD
C       TØTAL IS ASSIGNED TØ NTEST ABØVE                                LØAD
C                                                                       LØAD
      IF( NTEST .LT. MEMSIZ )                                           LØAD
C       THEN HAVE ENØUGH RØØM                                           LØAD
     *                                                        GØ TØ 119 LØAD
C       ELSE WILL DESTRØY DATA IN BLANK CØMMØN                          LØAD
          INEED = NTEST - NDATA                                         LØAD
          WRITE(6,118) NØB,NØVAR,NDATA,NTEST,MEMSIZ,INEED,NTEST         LØAD
  118     FØRMAT( 47H0*** ERRØR.  TRYING TØ LØAD (NØ. ØBSERVATIØNS)* ,  LØAD
     *      18H(NØ. VARIABLES) =  ,I6,1H*,I6,3H = ,I10,7H VALUES/       LØAD
     *      13X, 13HTHIS REQUIRES, I10, 23H WØRDS IN BLANK CØMMØN.  ,   LØAD
     *      10H ØNLY HAVE,I10, 11H AVAILABLE.  / 13X, 8HINCREASE  ,     LØAD
     *      53H BLANK CØMMØN SIZE ØR REWRITE SUBRØUTINE LØAD TØ LØAD ,  LØAD
     *      12H VALUES WITH, I10, 17H WØRDS INSTEAD ØF , I10 /          LØAD
     *      10H0ABØRT RUN  )                                            LØAD
                                                             CALL ABØRT LØAD
C        (NØTE THAT A CALL ØF ERG WHICH CALLS SUPER WHICH CALLS         LØAD
C         INPT IN IBM VERSIØN WØULD DESTRØY THE DATA)                   LØAD
C                                                                       LØAD
  119     CØNTINUE                                                      LØAD
C                                                                       LØAD
C     END ØF CØRRECTIØN                                                 LØAD
C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$ LØAD
C                                                                       LØAD
C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$ LØAD
C     NØTE ØN EFFICIENT USE ØF STØRAGE.                                 LØAD
C     THE IBM VERSIØN AND THIS VERSIØN USE  NTEST  WØRDS IN BLANK       LØAD
C     CØMMØN TØ READ  NØB*NØVAR  DATA.  THIS AMØUNT CAN BE REDUCED      LØAD
C     BY. NØB*NØVAR  IF THE LØADING IS DØNE IN A DØ-LØØP.               LØAD
C     EACH ITERATIØN READS ØNLY  NØVAR  DATA, THE VALUES ØF THE         LØAD
C     NEXT ØBSERVATIØN FØR EACH VARIABLE.  THIS REQUIRES THE            LØAD
C     CØNSTRUCTIØN ØF A NEW ENTRY TØ  INPT  TØ READ A SPECIFIED         LØAD
C     NUMBER (NØVAR) ØF VALUES.  SUBRØUTINE  FØRMAT  HAS TØ BE          LØAD
C     CHANGED SØ THAT IT READS ØNLY  NØVAR  VALUES.                     LØAD
C                                                                       LØAD
C     END ØF NØTE                                                       LØAD
C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$ LØAD
C                                                                       LØAD
C     RECØRD THE USE ØF BLANK CØMMØN BY LØAD                            LØAD
      LØDUSE = MAX0( LØDUSE, NTEST - NØB*NØVAR )                        LØAD
CCDCTØIBM                                                               LØAD
      NDT = 0                                                           LØAD
      NPØINT = NLDAT                                                    LØAD
CIBMTØCDC                                                               LØAD
C                                                                       LØAD
C     REPLACED  NWØRD+1, NWØRD+2  WITH  2,3   (NWØRDD = 1)              LØAD
      JARG( 2 ) = 3                                                     LØAD
      JARG( 3 ) = 0                                                     LØAD
C                                                                       LØAD
  120 CALL XNPT( X(NPØINT), N, X(NPØINT+1) )                            LØAD
C           XNPT IS AN ENTRY TØ NEW VERSIØN ØF INPT                     LØAD
C                                                                       LØAD
C     REPLACED                                                          LØAD
C     IF(IFCØMP(X(NPØINT),NAMES(13))) CALL FØRMAT(X(NPØINT),N,NØVAR)    LØAD
C                                                                       LØAD
C     TEST FØR KEYWØRD ******FØRMAT******                               LØAD
      IF( JX(NPØINT) .EQ. NAMES(6) )                                    LØAD
     *     CALL FØRMAT( X(NPØINT), N, NØVAR )                           LØAD
CCDCTØIBM                                                               LØAD
```

```
      IF (NØVAR .GT. 1) GØ TØ 121
      NLBUF = NLDAT
CIBMTØCDC
C
C     REPLACED  NWØRD + 1  WITH  2   (NWØRD = 1)
      L = 2
CCDCTØIBM
      GØ TØ 140
CIBMTØCDC
  121 CØNTINUE
C
C     REMØVED FØLLØWING FRØM INSIDE DØ-130-LØØP
      K = NLBUF + NØB - 1
C
      DØ 130 I = 1, NØVAR
C
C     REPLACED  I*NWØRD + 1  WITH  I + 1   (NWØRD = 1)
      L = I + 1
CCDCTØIBM
      NPØINT = NLDAT + I - 1
      DØ 125 J=NLBUF,K
      X(J) = X(NPØINT)
  125 NPØINT = NPØINT + NØVAR
CIBMTØCDC
C
C     REPLACED  CALL NMØV(JARG(1),X(L))  (NWØRD = 1)
      JARG(1) = JX(L)
CCDCTØIBM
  130 CALL TSPUT(JARG(1),X(NLBUF))
  135 IF(N.NE.NDATA) CALL ERG(10,N)
      GØ TØ 1
CIBMTØCDC
C     REPLACED  CALL NMØV(JARG(1),X(L))  (NWØRD = 1)
  140 JARG(1) = JX(L)
CCDCTØIBM
      CALL TSPUT(JARG(1),X(NLBUF))
      GØ TØ 135
C
CIBMTØCDC
C     HAVE TSP KEYWØRD  ******END******
CCDCTØIBM
C
  200 PRØFF=.FALSE.
CIBMTØCDC
C
C     NEXT IS ENTRY TØ SUPER
CCDCTØIBM
      CALL RETURN
CIBMTØCDC
C     NØ RETURN TØ THIS PØINT FRØM ENTRY RETURN ØF SUPER
CCDCTØIBM
C
C     SECTIØN FØR MASK
C
  300 DØ 310 I=1,N
  310 MASK(I) = X(I)
      GØ TØ 1
C
CIBMTØCDC
C     HAVE TSP KEYWØRD  ******SMPL******
CCDCTØIBM
C
  400 NARG = N - 1
CIBMTØCDC
C
C     REPLACE  NWØRD + 1  WITH  2
      KK = 2
```

```
CCDCTØIBM
      DØ 520 I=2,N
      CALL ARGPUT(I-1,X(KK),TYPE(I),0)
  520 KK = KK + 1
      CALL SMPL
      GØ TØ 1
C
CIBMTØCDC
C     HAVE TSP KEYWØRD  ******NØPRINT******
CCDCTØIBM
C     TURN ØFF PRINTING ØF DATA AS LØADED
C
  500 PRØFF=.TRUE.
      GØ TØ 1
      END
05. 23. 33. J0 16 EP30          6 FEET
```

```
      SUBRØUTINE GRAPH                                                  GRAPH     1
C                                                                       GRAPH     2
C           THIS IS AN ADAPTATIØN ØF **PLØT-C**                         GRAPH     3
C                                                                       GRAPH     4
CIBMTØCDC                                                               GRAPH     5
C                                                                       TSPCØM    1
      CØMMØN /TSPCØM/                                                   TSPCØM    2
     * MEMSIZ, NØB   , NSPARG, NWØRD , LENGTH,                          TSPCØM    3
     * NTYPE , IFDBUG, IFTITL, NCHAR , NSUP  ,                          TSPCØM    4
     * MEMST , NØREG , IFPLØT, IFFAST, NPAGE ,                          TSPCØM    5
     * NUMLIN, IFREPL, PRØFF , SKIP(11), JPHAS ,                        TSPCØM    6
     * LIMARG, LINE  , NJARG , NARG  , NAME ,                           TSPCØM    7
     *  JARG(4)                                                         TSPCØM    8
C            LENGTH ØF JARG SET IN MAIN ØVERLAY                         TSPCØM    9
C                                                                       TSPCØM   10
C     DELETED  NAME2  BETWEEN  NAME  AND  JARG  IN  /TSPCØM/ (NWØRD = 1)TSPCØM   11
C                                                                       TSPCØM   12
      LØGICAL IFDBUG, IFTITL, IFPLØT, IFFAST,                           TSPCØM   13
     *         IFREPL, PRØFF                                            TSPCØM   14
C                                                                       TSPCØM   15
C                                                                       MEMCØM    1
C     NEW CØMMØN BLØCK ADDED                                            MEMCØM    2
      CØMMØN /MEMCØM/ IMNSZ(7), USRNAM, PSSVL(20)                       MEMCØM    3
      INTEGER USRNAM                                                    MEMCØM    4
      EQUIVALENCE                                                       MEMCØM    5
     * (IMNSZ(1),MMMSIZ), (IMNSZ(2),LLMARG), (IMNSZ(3),LLMDØT),         MEMCØM    6
     * (IMNSZ(4),LLMBUF), (IMNSZ(5),LLMSYM), (IMNSZ(6),LLMSMP),         MEMCØM    7
     * (IMNSZ(7),LLMØUT)                                                MEMCØM    8
C                                                                       MEMCØM    9
C     IMMSZ USED TØ KEEP TRACK ØF MEMØRY USE                            MEMCØM   10
      DIMENSIØN IPSSVL(20)                                              MEMCØM   11
      EQUIVALENCE (IPSSVL(1),PSSVL(1))                                  MEMCØM   12
C     PSSVL AND IPSSVL USED TØ PASS VARIABLES BETWEEN SUBRØUTINES       MEMCØM   13
C     IN DIFFERENT ØVERLAYS                                             MEMCØM   14
C     IPSSVL(1) IS USED TØ KEEP RECØRD ØF USE ØF CØMMØN IN LØAD         MEMCØM   15
C        AND IN GENR                                                    MEMCØM   16
C     IPSSVL(3) PASSES CØMPUTED GØ TØ INDEX FRØM EXEC TØ MATRIX         MEMCØM   17
C     IPSSVL(2) PASSES CØMPUTED GØ TØ INDEX FRØM EXEC TØ ØVERLAYS       MEMCØM   18
C     IPSSVL(4) IS USED TØ KEEP RECØRD ØF USE ØF JARG-CØMMØN            MEMCØM   19
C     IPSSVL(5) IS USED TØ KEEP RECØRD ØF USE ØF DØT-CØMMØN             MEMCØM   20
C     IPSSVL(6) IS USED TØ KEEP RECØRD ØF USE ØF BUFFER-CØMMØN          MEMCØM   21
C     IPSSVL(7) IS USED TØ KEEP RECØRD ØF USE ØF JSML-CØMMØN            MEMCØM   22
C     IPSSVL(8) IS USED TØ KEEP RECØRD ØF USE ØF ØUTBUF-CØMMMØN         MEMCØM   23
      EQUIVALENCE                                                       MEMCØM   24
     * (LØDUSE, IPSSVL(1)),, (JTØØVL, IPSSVL(2)), (JTØMAT, IPSSVL(3))   MEMCØM   25
     *,(JARUSE, IPSSVL(4)), (DØTUSE, IPSSVL(5)), (BUFUSE, IPSSVL(6)),   MEMCØM   26
     * (SMPUSE, IPSSVL(7)), (ØBFUSE, IPSSVL(8))                         MEMCØM   27
      INTEGER DØTUSE, BUFUSE, SMPUSE, ØBFUSE                            MEMCØM   28
C                                                                       MEMCØM   29
C                                                                       GRAPH     8
C     INSERTED LIMITS ØN ARRAY LENGTHS IN GRAPHX (CALLED BELØW)         GRAPH     9
      DATA LIMGRX / 5000 /                                              GRAPH    10
C                                                                       GRAPH    11
CCDCTØIBM                                                               GRAPH    12
      CØMMØN SPACE(1)                                                   GRAPH    13
      IF(NARG.GT.2)CALL ERG(130,NARG)                                   GRAPH    14
      IF(NØB.GT.1500)CALL ERG(131,NØB)                                  GRAPH    15
      I2=NØB+1                                                          GRAPH    16
      I3=NSPARG+1                                                       GRAPH    17
      I4=I2+NØB                                                         GRAPH    18
      IF(I4.GT.MEMSIZ)CALL ERG(132,I4)                                  GRAPH    19
CIBMTØCDC                                                               GRAPH    20
C                                                                       GRAPH    21
C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$GRAPH    22
C     MAJØR ERRØR IN IBM VEPSIØN                                        GRAPH    23
C                                                                       GRAPH    24
```

```
C     CALL ØF GRAPHX CAN DESTRØY DATA IN BLANK CØMMØN.
C     THE CØRRECTIØN BELØW IS CØNSERVATIVE--MAXIMUM PØSSIBLE USE
C     ØF BLANK CØMMØN BY GRAPHX IS TESTED FØR
C
      IF(LIMGRX . LT.  MEMSIZ)
C       THEN HAVE ENØUGH RØØM IN BLANK CØMMØN
     *                                                       GØ TØ 20
C       ELSE NØ ENØUGH RØØM
          WRITE(6,10) LIMGRX, MEMSIZ
   10     FØRMAT(45H0*** ERRØR.  EXECUTIØN ØF GRAPH MIGHT DESTRØY  ,
     *       28H DATA IN BLANK CØMMØN.  NEED, I10, 6H WØRDS, ,
     *       10H ØNLY HAVE, I10, 10H AVAILABLE )
          CALL ERG(132, ITEST)
C         NØ RETURN TØ THIS PØINT FRØM ERG
C
   20     CØNTINUE
C
C     END ØF CØRRECTIØN
C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C
C     ADDED A RECØRD ØF USE ØF BLANK CØMMØN USE
      LØDUSE = MAX0(LØDUSE, ITEST)
CCDCTØIBM
      CALL TSMGET(NARG, JARG, SPACE)
      CALL GRAPHX(SPACE(1), SPACE(I2))
      RETURN
      END
      SUBRØUTINE GRAPHX(YVAR, XVAR)
C
C         THIS IS A REVISIØN ØF **PLØT C**
C         22 JUNE 1968
C
C          NØTE THAT SPACE MUST BE DIMENSIØNED . GE.  5000 IN FRESH
C          THIS RØUTINE IS GENERALLY DØRPPED IN THE 7094 VERSIØN
CIBMTØCDC
C
      CØMMØN /TSPCØM/
     * MEMSIZ, NØB   , NSPARG, NWØRD , LENGTH,
     * NTYPE , IFDBUG, IFTITL, NCHAR , NSUP  ,
     * MEMST , NØREG , IFPLØT, IFFAST, NPAGE ,
     * NUMLIN, IFREPL, PRØFF , SKIP(11), JPHAS ,
     * LIMARG, LINE  , NJARG , NARG  , NAME ,
     *  JARG(4)
C               LENGTH ØF JARG SET IN MAIN ØVERLAY
C
C     DELETED  NAME2  BETWEEN  NAME  AND  JARG  IN  /TSPCØM/ (NWØRD = 1)
C
      LØGICAL IFDBUG, IFTITL, IFPLØT, IFFAST,
     *          IFREPL, PRØFF
C
CCDCTØIBM
      CØMMØN SPACE(3000),IBEGIN(250),IEND(250),INDY(1500)
      DIMENSIØN A(16),NØTIE(100),TAB(100),TA(100),TABLE(100),X(100),Y(10
     D4),XVAR(1),YVAR(1)
      INTEGER TA
      LØGICAL JQ
      DATA A/1H ,1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1HX,1H ,1H+,1H.
     C,1H'/
      DØ 7 I=1,250
      IEND(I)=0
    7 IBEGIN(I)=0
C
C     ARRANGE Y DATA BY DECREASING ØRDER
C
    5 DØ 32 ID=1,NØB
      DØ 32 JD=ID,NØB
      IF(YVAR(ID). GE. YVAR(JD))GØ TØ 32
      DIAGS=YVAR(ID)
      YVAR(ID)=YVAR(JD)
      YVAR(JD)=DIAGS
      DIAGS=XVAR(ID)
      XVAR(ID)=XVAR(JD)
      XVAR(JD)=DIAGS
   32 CØNTINUE
C
C     FIND RANGES AND INCREMENTS
C
      XMA=XVAR(1)
      XMIN=XMA
      DØ 18 JB=2,NØB
      IF(XMA. LT. XVAR(JB))XMA=XVAR(JB)
      IF(XMIN. GT. XVAR(JB))XMIN=XVAR(JB)
   18 CØNTINUE
      IF(XMA. LE. 0)GØ TØ 404
      IF(XMIN. GE. 0)GØ TØ 404
      RANGEX=XMA+ABS(XMIN)
      GØ TØ 405
  404 RANGEX=ABS(XMA-XMIN)
  405 IF(YVAR(1). LE. 0)GØ TØ 408
      IF(YVAR(NØB). GE. 0)GØ TØ 408
      RANGEY=YVAR(1)+ABS(YVAR(NØB))
      GØ TØ 409
  408 RANGEY=ABS(YVAR(1)-YVAR(NØB))
  409 XINC=RANGEX/99.
      YINC=RANGEY/103.
      X(1)=XMIN
      Y(1)=YVAR(1)
      DØ 27 IA=1,99
   27 X(IA+1)=X(IA)+XINC
      DØ 28 IB=1,103
   28 Y(IB+1)=Y(IB)-YINC
      Y(104)=Y(104)-.00049
      JI1=1
      DØ 102 II=1,NØB
      DØ 101 JI=JI1,104
      IF(YVAR(II). LT. Y(JI))GØ TØ 101
      INDY(II)=JI
      JI1=JI
      GØ TØ 102
  101 CØNTINUE
  102 CØNTINUE
C
C     CHECK FØR Y TIES, INITIAL + FINAL INDICES STØRED IN IBEGIN + IEND
C          (IT=TIE NØ. )
C
   97 IT=0
      NA1=1
      NA2=NØB-1
   34 CØNTINUE
      DØ 38 NA=NA1,NA2
      NB=NA+1
      IF(INDY(NA). NE. INDY(NB))GØ TØ 38
      IT=IT+1
      IF(IT. GT. 250)CALL ERG(133,IT)
      IBEGIN (IT) = NA
      DØ 37 MB=NB,NØB
      IF(INDY(NA). EQ INDY(MB))GØ TØ 37
      NA1=MB
      IEND(IT) = MB - 1
      GØ TØ 34
   37 CØNTINUE
      IEND(IT)=NØB
      GØ TØ 111
   38 CØNTINUE
  111 CALL ØUTPT
      IF(NUMLIN.GT.0) CALL HEDING
C
C     ØUTPUT LØØP (THRØUGH 87), WRITES ØNE LINE ØF GRAPH EACH TIME THRU.
C
      IM1=0
      JE1 = 1
      LUCK = 0
      DØ 86 IQ=1,104,2
      II=IQ+1
      DØ 123 NC=1,100
      TA(NC)=0
  123 TAB(NC)=A(1)
   40 JE2 = JE1 + 1
      JE3 = JE1
      DØ 70 NC = 1,100
   70 TABLE(NC) = A(1)
      JQ=.TRUE.
      IF(INDY(JE1).NE.IQ)GØ TØ 71
      JQ=.FALSE.
      GØ TØ 72
   71 IF(INDY(JE1).NE.II)GØ TØ 48
   72 IF(INDY(JE1).NE.INDY(JE2))GØ TØ 74
      IM1=IM1+1
      JE1 = IBEGIN(IM1)
      JE3 = IEND(IM1)
   74 DØ 77 JG = JE1,JE3
      DØ 113 IJ=1,100
      IF(XVAR(JG).LE.X(IJ))GØ TØ 75
  113 CØNTINUE
      IJ=100
   75 IF(TABLE(IJ).NE.A(1))GØ TØ 170
      IF(TAB(IJ).EQ.A(1))GØ TØ 69
      IF(TAB(IJ).EQ.A(16))GØ TØ 45
      TA(IJ)=TA(IJ)+1
   51 TAB(IJ)=A(1)
      GØ TØ 54
  170 TA(IJ)=TA(IJ)+1
   54 IF(TA(IJ)-10)62,64,63
   62 LAZY=TA(IJ)+2
      TABLE(IJ)=A(LAZY)
      TAB(IJ)=A(1)
      GØ TØ 77
   64 LUCK = LUCK+1
      NØTIE(LUCK)=10
      TABLE(IJ)=A(12)
      TAB(IJ)=A(1)
      GØ TØ 77
   63 NØTIE(LUCK)=NØTIE(LUCK) + 1
      GØ TØ 77
   69 IF(JQ)GØ TØ 45
      TABLE(IJ)=A(16)
      GØ TØ 65
   45 TABLE(IJ)=A(15)
   65 TA(IJ)=TA(IJ)+1
   77 CØNTINUE
      JE1=JE3 + 1
      IF(JQ)GØ TØ 48
      DØ 47 J=1,100
   47 TAB(J)=TABLE(J)
      GØ TØ 40
   48 IF(II.EQ.2)GØ TØ 12
   13 WRITE(6,2017) (TAB(ME),ME=1,100)
 2017 FØRMAT(15X100A1)
      MBB=MØD(II,3)-2
      IF(MBB)85,84,85
CIBMTØCDC
```

```
C
C       DELETED IF(NWØRD.EQ.1) GØ TØ 22    (NWØRD = 1)
C               WRITE(6,2016) JARG(1),JARG(2)
C          2016 FØRMAT(7X,2A4)
C               GØ TØ 13
C
   12 CØNTINUE
CCDCTØIBM
   22 WRITE(6,3016) JARG(1)
CIBMTØCDC
C
C      REPLACED 9X,A6
 3016 FØRMAT( 9X,  R8 )
CCDCTØIBM
      GØ TØ 13
   84 WRITE(6,2008) Y(II),(TABLE(ME),ME=1,100)
 2008 FØRMAT(1H+F12.3,2H *100A1)
      GØ TØ 86
   85 WRITE(6,2009) (TABLE(ME),ME=1,100)
 2009 FØRMAT(1H+13X1H*100A1)
   86 CØNTINUE
      I3=1+NSPARG
CIBMTØCDC
C
C      REMØVED IF(NWØRD.EQ.1) GØ TØ 83   (NWØRD = 1)
C      AND A PRINT SIMILAR TØ THE ØNE AT 83
CCDCTØIBM
   83 WRITE(6,3010)JARG(I3),X(1),X(20),X(40),X(60),X(80),X(100),X(10),
     WX(30),X(50),X(70),X(90)
CIBMTØCDC
C
C      CHANGED A6 TØ R8 (CDC6500)
 3010 FØRMAT( 15X,  100(1H*),  R8 / 7X,  2(8X,1H*),  9(9X,
CCDCTØIBM
     F1H*)/F19.3,5(F19.3,1X)/F28.3,4F20.3)
   89 CALL HEDING
      IF(LUCK.EQ.0)GØ TØ 60
      WRITE(6,2004)
 2004 FØRMAT(105H LIST ØF TIED PØINT CØUNTS WHERE NUMBER ØF TIES IS GREA
     1TER THAN 9 (READING DØWN Y-AXIS AND ACRØSS X-AXIS))
      LØV1=0
      LØV2=0
  150 LØV2=LØV1+30
      LØV1=LØV2
      IF(LØV1.EQ.LUCK)GØ TØ 152
      LØV2=LØV2-(LØV1-LUCK)
  152 L=LØV1-29
      WRITE(6,2015) (NØTIE(J),J=L,LØV2)
 2015 FØRMAT(1H,30I4)
      IF(LØV1.LT.LUCK)GØ TØ 150
   60 RETURN
      END
05 23.49.JØ 16 EP30          6 FEET
```

```
      SUBRØUTINE CAPITL
CIBMTØCDC
C
      CØMMØN /TSPCØM/
     * MEMSIZ, NØB   , NSPARG, NWØRD , LENGTH,
     * NTYPE , IFDBUG, IFTITL, NCHAR , NSUP  ,
     * MEMST , NØREG , IFPLØT, IFFAST, NPAGE ,
     * NUMLIN, IFREPL, PRØFF , SKIP(11), JPHAS ,
     * LIMARG, LINE  , NJARG , NARG  , NAME ,
     *  JARG(4)
C              LENGTH ØF JARG SET IN MAIN ØVERLAY
C
C     DELETED  NAME2  BETWEEN  NAME  AND  JARG  IN  /TSPCØM/ (NWØRD = 1)
C
      LØGICAL IFDBUG, IFTITL, IFPLØT, IFFAST,
     *          IFREPL, PRØFF
C
CCDCTØIBM
      CØMMØN DINV(600),CAP(600),DNINV(600),REPL(600),XX(2)
CIBMTØCDC
C
C     NEXT IS BLANK CØMMØN SPACE NEEDED BY ARRAYS ABØVE
      DATA LIMSIZ / 2402 /
C
C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C     MAJØR ERRØR IN IBM VERSIØN, NØ TEST ØN AVAILABLE BLANK
C     CØMMØN SIZE
C
C     ADDED TEST
      IF( LIMSIZ .LT. MEMSIZ )
C       THEN HAVE ENØUGH RØØM
     *                                                        GØ TØ 30
C       ELSE NØT ENØUGH RØØM
          WRITE(6,20) LIMSIZ, MEMSIZ
   20     FØRMAT(25H0*** ERRØR.  CAPITL NEEDS , I10,
     *      29H WØRDS IN BLANK CØMMØN.  ØNLY  , I10,
     *      42H AVAILABLE.  PRØCEDURE CAPITL NØT EXECUTED   // )
          RETURN
C
   30     CØNTINUE
C
C     END ØF CØRRECTIØN
C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
CCDCTØIBM
      CALL TSGET(JARG,DINV)
      CALL ARGGET(5,XX,JJ,JJ)
      NBENCH = XX(1)
      CALL ARGGET(6,XX,JJ,JJ)
      DELTA = XX(1)
      CALL ARGGET(7,XX,JJ,JJ)
      BENCH = XX(1)
      CAP(NBENCH) = BENCH
      XDELT = 1. - DELTA
      NN = NBENCH + 1
      NNN = NBENCH - 1
      IF (NN .GT. NØB) GØ TØ 101
      DØ 100 J=NN,NØB
  100 CAP(J) = XDELT*CAP(J-1) + DINV(J)
  101 IF(NNN .EQ. 0) GØ TØ 106
      DØ 105 J=1,NNN
      K = NBENCH - J
  105 CAP(K)=(CAP(K+1)-DINV(K+1))/XDELT
  106 DØ 110 J=1,NØB
      REPL(J) = DELTA*CAP(J)
  110 DNINV(J) = DINV(J) - REPL(J)
      CALL ARGGET(2,XX,ITYPE,JJ)
      IF(ITYPE .EQ. 3)  CALL TSPUT(JARG(NSPARG+1),CAP)                  CAPITL     53
      CALL ARGGET(3,XX,ITYPE,JJ)                                        CAPITL     54
      IF(ITYPE .EQ. 3)  CALL TSPUT(JARG(2*NSPARG+1),DNINV)              CAPITL     55
      CALL ARGGET(4,XX,ITYPE,JJ)                                        CAPITL     56
      IF(ITYPE .EQ. 3)  CALL TSPUT(JARG(3*NSPARG+1),REPL)               CAPITL     57
      RETURN                                                            CAPITL     58
      END                                                               CAPITL     59
05.24.02.J0 16 EP30        2 FEET
```

```
      SUBRØUTINE INPRØD(N, JSA, JSB, A, B, PRØD)
CIBMTØCDC
C
C     CALLED BY GGGMLT GTGMLT G2YMLT ØRTHØS
C               TGGMLT T2YMLT TINV  UNTRAN VGVMLT
C               YFACT
CCDCTØIBM
C
C     THIS SUBRØUTINE CALCULATES THE INNER PRØDUCT, AØB, ØF THE VECTØRS
C     A AND B.
CIBMTØCDC
C     SUM I = 1, N   A(I, JSA)*A(I, JSB)
C
C     CØRRECTED MARCH 75 WITH ADDITIØN ØF TEMP
C
CCDCTØIBM
C
C     NØTE THAT THE INNER PRØDUCT ØF VECTØRS ØF ZERØ LENGTH IS RETURNED
C     AS ZERØ
C
      DIMENSIØN A(1), B(1)
      DØUBLE PRECISIØN XPRØD
CIBMTØCDC
     * , TEMPA, TEMPB
CCDCTØIBM
      J = 1
      XPRØD = 0.
      IF (N) 150, 150, 50
   50 NN = JSA*N
      DØ 100 I=1, NN, JSA
CIBMTØCDC
      TEMPA = A(I)
      TEMPB = B(J)
      XPRØD = XPRØD + TEMPA*TEMPB
CCDCTØIBM
  100 J = J + JSB
  150 PRØD = XPRØD
      RETURN
      END
```

```
05. 24. 15. J0 16 EP30        2 FEET
```

References

[1] Time Series Processor User's Manual, Princeton University Department of Economics, TSP (360-91), February 1971.