DOCUMENT RESUME

ED 241 552 TM 840 048

AUTHOR Lawson, Edwin D.; And Others
TITLE Computer Programs for the Semantic Differential: Further Modifications.
PUB DATE [84]
NOTE 85p.; For related documents, see ED 163 033 and ED 187 759.
PUB TYPE Computer Programs (101)

EDRS PRICE MF01/PC04 Plus Postage.
DESCRIPTORS Adjectives; Attitude Measures; *Computer Programs; Models; *Semantic Differential; *Statistical Analysis; Statistical Significance; Test Interpretation
IDENTIFIERS FORTRAN Programing Language

ABSTRACT

The original nine programs for semantic differential analysis have been condensed into three programs which have been further refined and augmented. They yield: (1) means, standard deviations, and standard errors for each subscale on each concept; (2) Evaluation, Potency, and Activity (EPA) means, standard deviations, and standard errors; (3) Osgood D values for all concepts; (4) the correlation of distance measures obtained with EPA scores with D values; and (5) tests of significance of D values with key reference concepts. The additional programming provides a converted standard score for each concept and tests of significance for D values on each concept. There are four new programs, providing semantic differential scale packets ready for use of respondents; a three-dimensional plot of how the concepts position themselves, based upon the EPA scores; t-tests between the means of two samples on E, P, and A scores; and correlations between scores on each concept with two samples. All but one of the programs are written in standard FORTRAN 77 for use on a Burroughs 6800. The remaining program, also written for use on a Burroughs 6800, is a FORTRAN IV program with a COBOL sort routine bound into it. (BW)

COMPUTER PROGRAMS FOR THE SEMANTIC DIFFERENTIAL: FURTHER MODIFICATIONS

EDWIN D. LAWSON, BARBARA L. METIVIER, AND MARCY L. METIVIER

STATE UNIVERSITY OF NEW YORK, FREDONIA.

Computer Programs for the Semantic Differential: Further Modifications

Edwin D. Lawson, Barbara L. Metivier, and Marcy L. Metivier
State University of New York, Fredonia

Description. Previous publications (Lawson, 1979; Lawson & Metivier, 1980) described the rationale of some types of semantic differential analysis and gave listings of programs with examples. The original nine programs have been condensed into three programs which have been further refined and augmented. There are also four new programs. The investigator is now able to perform a number of tasks.

Input. The initial program reads data in the form of scale responses by concept (nine subscales for each concept) for each respondent. Concepts may be presented to respondents in randomized order and polarity, thus avoiding position effects.

Output. The programs yield: (1) means, standard deviations, and standard errors for each subscale on each concept; (2) Evaluation, Potency, and Activity (EPA) means, standard deviations, and standard errors; (3) Osgood D values for all concepts; (4) the correlation of distance measures obtained with EPA scores with D values; (5) tests of significance of D values with key reference concepts such as, Good, Bad, Strong, Weak, Active, and Passive (the user may choose any reference concepts).

The additional programming provides: (1) a converted standard score for each concept on a scale from 0 to 100 along the selected dimensions, and (2) tests of significance for D

values on each concept between two samples using the Wilcoxon test for matched samples, or the Mann-Whitney for independent samples, as appropriate.

There are four new programs. The first prints semantic differential scale packets ready for use of respondents. Each concept to be rated has nine subscales. There are six sets of scales on a standard 14-inch-width computer sheet. The polarity of the scales is alternated and the order of the concepts can be randomized so that no two respondents have the same order. T.e second program does a three-dimensional plot of how the concepts position themselves (this plot is based upon the EPA scores and, on the average, correlates about .95 with results from the more precise Osgood D method. This plot gives a good approximation of how a three-dimensional model built from either method would appear. The third program computes t-tests between the means of two samples on E, P, and A scores, as men vs. women. The fourth program calculates correlations beween scores on each concept with two samples, as men vs. women or test-retest.

Capacity. All but one of the programs are written in standard FORTRAN 77 for use on a Burroughs 6800. The remaining program, also written for use on a Burroughs 6800, is a FORTRAN IV program with a COBOL sort routine bound into it. With another computer, adjustments might be needed such as the COBOL routine being replaced by a FORTRAN sort routine. The programs handle up to 109 concepts on nine subscales. The number of cases is limited by the size of the arrays used in the programs. The Fredonia programs can handle up to 50 cases. However, where two groups

are compared, as men vs. women, 50 cases in each group can be compared.

Availability. A listing of the programs and of the directions for running the programs follows this abstract. For further information about the programs contact: Barbara L. Metivier, Computing Services, State University College, Fredonia, New York 14063.

References

Lawson, E. D. (1979). An easy semantic differential technique. Fredonia, NY: State University College, Department of Psychology. (ERIC Document Reproduction Service No. ED 163 033)

Lawson, E. D., & Metivier, B. L. (1980). Computer programs for the semantic differential: An update and expansion. Fredonia, NY: State University College, Department of Psychology. (ERIC Document Reproduction Service No. 187 759)

```
$RESET FREE SET LIST
FILE  5(KIND="REMOTE",MYUSE="IO")
FILE  6(FILE="SEMDIF/PRT.",KIND="PRINTER")
FILE  7(FILE="SEMDIF/OUT.",KIND="DISK",MAXRECSIZE=16,BLOCKSIZE=240,
     *STATUS="NEW",AREASIZE=60)
FILE  10(FILE="SEMDIF/INPUT.",KIND="DISK",FILETYPE=7)
FILE  12(FILE="SEMDIF/D01.",KIND="DISK",MAXRECSIZE=14,BLOCKSIZE=420,
     *STATUS="SCRATCH",MYUSE="IO",UPDATEFILE="TRUE")
FILE  14(FILE="SEMDIF/D02.",KIND="DISK",MAXRECSIZE=14,BLOCKSIZE=420,
     *STATUS="SCRATCH",MYUSE="IO",UPDATEFILE="TRUE")
FILE  16(FILE="SEMDIF/D03.",KIND="DISK",MAXRECSIZE=14,BLOCKSIZE=420,
     *STATUS="SCRATCH",MYUSE="IO",UPDATEFILE="TRUE")
FILE  17(FILE="SEMDIF/D04.",KIND="DISK",MAXRECSIZE=14,BLOCKSIZE=420,
     *STATUS="SCRATCH",MYUSE="IO",UPDATEFILE="TRUE")
C*
C*****************************************************************
C*                                                               *
C*          SEMANTIC DIFFERENTIAL SEMDIF*- SOURCE FILE           *
C*                        OBJECT/SEMDIF - OBJECT FILE            *
C*                                                               *
C*                                                               *
C*     THIS PROGRAM CAN PERFORM THE COMPUTATIONS FOR:            *
C*          1. MEANS AND STANDARD DEVIATIONS ON SUBSCALES        *
C*          2. MEANS AND STANDARD DEVIATIONS ON EVALUATION, POTENCY,  *
C*             AND ACTIVITY FACTORS (EPA SCORES)                 *
C*          3. OSGOOD D VALUES FOR ALL CONCEPTS                  *
C*          4. THE CORRELATION BETWEEN DISTANCE MEASURES OBTAINED*
C*             FROM EPA SCORES AND OSGOOD'S D                    *
C*          5. TESTS OF SIGNIFICANCE OF DISTANCES OBTAINED WITH  *
C*             OSGOOD D.                                         *
C*                                                               *
C*     THE PROGRAM CHECKS AN INPUT DISK FILE (SEMDIF/INPUT) FOR  *
C*     ERRORS.  IF OK, IT THEN FINDS THE EPA VALUES AND THE DISTANCES  *
C*     BETWEEN CONCEPTS USING THESE EPA'S.  NEXT IT CALCULATES THE     *
C*     MEANS AND THE D VALUES FOR THE SEMANTIC DIFFERENTIAL USING      *
C*     THESE MEANS.  FINALLY IT FINDS THE SPEARMAN RANK COEFFICIENT    *
C*     FOR THESE TWO DISTANCE TABLES.  IT WILL HANDLE ONE CASE OR      *
C*     MANY CASES WITH UP TO 109 CONCEPTS WITH 9 SUBSCALES PER         *
C*     CONCEPT.                                                  *
C*                                                               *
C*     THE INPUT FILE (SEMDIF/INPUT) HAS THE FOLLOWING FORMAT:   *
C*                                                               *
C*        RECORD     COL       CONTENTS                         *
C*          1        1-3       NUMBER OF CASES                  *
C*                   4-6       SPACE                            *
C*                   7-9       NUMBER OF CONCEPTS               *
C*                  10-12      SPACE                            *
C*                  13-14      PRINT CODE                       *
C*                             01 - PRINTS MEANS AND MEANS DOUBLED  *
C*                             02 - PRINTS MEANS, MEANS DOUBLED,SUM,*
C*                                  SUM SQUARES,STANDARD DEVIATION,  *
C*                                  STANDARD ERROR OF THE MEANS      *
C*                  15-17      SPACE                            *
C*                  18-20      IDENT NUMBER OF 1ST CASE         *
```


366 A

```
C*                  21-23      SPACE                                        *
C*                  24-26      IDENT NUMBER OF LAST CASE                    *
C*                  27-28      SPACE
C*                  29-42      FILE NAME
C*                                                                          *
C*          2        1-3       CASE IDENT NUMBER - NUMERIC                  *
C*          3        1-3       CASE IDENT NUMBER                            *
C*                   4-6       CONCEPT IDENT NUMBER - NUMERIC               *
C*                   7-21      CONCEPT NAME - ALPHA                         *
C*                  22-23      SPACE                                        *
C*                  24-32      SUBSCALE SCORES - NUMERIC                    *
C*                                                                          *
C*              REPEAT RECORD 3 AS NEEDED - 1 FOR EACH CONCEPT              *
C*              ENDING WITH A FINISH RECORD.                                *
C*                                                                          *
C*        LAST      1-3       -99   FINISH RECORD                          *
C*                                                                          *
C*                                                                          *
C*      THE DISK FILES CREATED ARE ALL DEFINED AS SCRATCH FILES             *
C*      WHICH DISAPPEAR WHEN THE JOB IS FINISHED.                           *
C*                                                                          *
C*      IF THERE IS NOT THE PROPER NUMBER OF DATA RECORDS (1 PER            *
C*      CONCEPT) FOR EACH CASE AS DEFINED IN RECORD 1 OR IF ONE OF          *
C*      THE CASE IDENT RECORDS IS MISSING, THE RUN WILL BE ABORTED          *
C*      AND AN ERROR MESSAGE WILL BE PRINTED AS OUTPUT.  AT THAT            *
C*      POINT PLEASE, CHECK THE INPUT DATA.                                 *
C*                                                                          *
C*      THIS PROGRAM WAS DEVELOPED BY B.L.METIVIER FOR E.D.LAWSON           *
C*      AT STATE UNIVERSITY COLLEGE, FREDONIA, NEW YORK, 14063 IN           *
C*      SEPTEMBER, 1981.                                                    *
C*                                                                          *
C*      THE FOLLOWING SUBROUTINES WERE REPRINTED BY PERMISSION FROM         *
C*      SYSTEM 1360 SCIENTIFIC SUBROUTINE PACKAGE (360A-CM-03X)             *
C*      VERSION III PROGRAMMER'S MANUAL, COPYRIGHT 1966,1967,               *
C*      1968 BY INTERNATIONAL BUSINESS MACHINES CORPORATION:                *
C*           RANK   PAGE 71                                                 *
C*           TIE    PAGE 74                                                 *
C*           SRANK  PAGE 73                                                 *
C*                                                                          *
C*                          **** UPDATE****                                 *
C*                                                                          *
C*       THIS PROGRAM HAS BEEN MODIFIED SO THAT THE MEANS, THE STANDARD     *
C*       ERRORS OF THE MEANS, AND THE OSGOOD DEES ARE SAVED IN A FILE       *
C*       WITH THE EXTENSION "/OUTPUT". THIS FILE IS FOR USE WITH "STAT/     *
C*       SEMDIF/CORR", "STAT/SEMDIF/PLOT", AND "STAT/SEMDIF/TTEST".         *
C*       THE FORMAT OF THE RECORDS IS:                                      *
C*              RECORD        COLS      CONTENT                           *
C*                1            1-3      NUMBER OF CONCEPTS                *
C*                2            1-3      CONCEPT IDENT NUMBER              *
C*                             4-10     MEAN  OF E                        *
C*                            11-18     MEAN  OF P                        *
C*                            19-25     MEAN  OF A                        *
C*                            26-30     SPACE                             *
C*                            31-41     CONCEPT NAME - ALPHA              *
```

```
C*            3 - (N+1)                    SAME AS RECORD 2 - ONE FOR        *
C*                                         EACH CONCEPT                      *
C*               (N+2)          1-3        CONCEPT IDENT NUMBER              *
C*                              4-10       STAND. ERROR OF MEAN OF E         *
C*                             11-18         "      "   "   "   "  P         *
C*                             19-25         "      "   "   "   "  A         *
C*        (N+3) - 2*N+1                    SAME AS RECORD N+2 - ONE FOR      *
C*                                         EACH CONCEPT                      *
C*       (2*N)+2 - END          1-90       OSGOOD DEE VALUES FOR CONCEPTS    *
C*                                                                           *
C*      MODIFICATIONS DONE BY MARCY METIVIER. COMPLETED AUGUST               *
C*      1983 AT STATE UNIVERSITY COLLEGE, AT FREDONIA.                       *
C*                                                                           *
C*                                                                           *
C*                                                                           *
C*                                                                           *
C*****************************************************************************
C*                                                                           *
C*
C*
C*
C*
      DIMENSION ICONC(9)
      CHARACTER*3 CONCID, CONCNM*15, TITLE*18, TITLE1*18, FIDOUT*26
      CHARACTER*8 WHEN
C*
C*          PRINT DATE ON PRINTOUT
C*
      CALL DATIT(WHEN)
      WRITE(6,950)WHEN
C*
*C          ENTER NAME OF INPUT FILE FROM REMOTE
*C
      WRITE(5,*)"ENTER NAME OF INPUT FILE.  NO MORE THAN 18 CHARACTERS,
     *PLEASE"
      READ(5,905)TITLE1
      IF (TITLE1(18:).NE." ")THEN
         TITLE1(18:)="."
         GO TO 70
      ELSE
         DO 50 I=17,1,-1
   50       IF(TITLE1(I:I).NE." ")GO TO 60
   60       I=I+1
         TITLE1(I:I)="."
      END IF
C*
C*          OPEN INPUT FILE
C*
   70 OPEN (10, FILE=TITLE1)
C*
C*     CREATE NAME OF OUTPUT FILE FOR MEANS DOUBLED
C*
      I=I-1
      FIDOUT=TITLE1(1:I)//"/OUTPUT."
      WRITE(5,*)" THE FOLLOWING IS THE NAME OF THE FILE TO BE USED FOR G
```

```
     *RAPHING:              ",FIDOUT
C*
C*   OPEN OUTPUT FILE FOR MEANS DOUBLED
C*
      OPEN(7, FILE=FIDOUT)
C*
C*
C*        READ RUN DESCRIPTION CARD - # OF CASES, # OF CONCEPTS,
C*          PRINT CODE, CASE IDENT NUMBERS (1ST AND LAST)
C*
C*
      READ(10,903)L,N,IPSW,LTAG1,LTAG2,TITLE
       INERR=0
       M=N-6
      DO 205 J1=0,L
  100    READ(10,901,END=800)NCASE,CONCID,CONCNM,ICONC
C*
C*
C*              CHECK FOR FINISH CARD
C*
C*
  120    IF(NCASE.EQ.-99)GO TO 600
C*
C*
C*              IF NOT CASE IDENT CARD, GO TO ERROR ROUTINE
C*
C*
        IF(CONCID.EQ.' ')GO TO 140
C*
C*
C*          ERROR ON CASE IDENT CARD - CONCEPT NUMBER NOT BLANK
C*
C*          WRITE ERROR MESSAGE
C*          ADD TO ERROR COUNT
C*
C*
        WRITE(6,904)NCASE,IDENT,CONC
        INERR=INERR+1
C*
C*
C*          CHECK TO SEE IF START OF NEW CASE
C*
C*
      IF (NCASE.EQ.IDENT)GO TO 100
C*
C*
C*        READ DATA CARDS
C*
C*          IF IDENT NE NCASE, ERROR IN CASE IDENT NUMBER
C*
C*          CASE IDENT CARD OR FINISH CARD OUT OF ORDER.
C*          WRITE ERROR MESSAGE AND CHECK FOR CASE IDENT CARD.
C*
C*
```

```
  140    DO 205 J=1,N
         READ(10,901,END=700)IDENT,CONCID,CONCNM,ICONC
         IF(IDENT.NE.NCASE)THEN
C*            WRITE(6,904)NCASE,IDENT,CONCID
            INERR=INERR+1
            IF(CONCID.EQ.' ')THEN
               NCASE=IDENT
               GO TO 120
            END IF
         END IF
C*
  205    CONTINUE
C*
C*       FILE CONTAINS TOO MANY RECORDS FOR NUMBER OF CASES.
C*
      WRITE(6,*)"TOO MANY RECORDS FOR NUMBER OF CASES.  RUN ABORTED."
  300 WRITE(6,904)NCASE,IDENT,CONCID
      STOP
C*
C*
C*       END OF RUN
C*          IF NO ERRORS, CLOSE DISK FILE AND CALL SDEPA
C*          IF ERRORS EXIST, ABORT RUN
C*
C*
  600 IF(INERR.NE.0)THEN
         WRITE(6,904)NCASE,IDENT,CONCID
         STOP
      END IF
      ENDFILE 10
      REWIND 10
C*
C* CALL PROGRAM TO FIND E,P,A VALUES
C*
      CALL SDEPA
C*
C* CALL PROGRAM TO FIND DISTANCES BETWEEN CONCEPTS
C*
      CALL SD2PTD
C*
C* CALL PROGRAM TO FIND MEANS
C*
      CALL SDMSM
C*
C* CALL PROGRAM TO FIND D VALUES FOR SEMANTIC DIFFERENTIAL
C*
      CALL SDGPD
C*
C* CALL PROGRAM TO FIND SPEARMAN RANK COEFFICIENT
C*
      CALL LRANK
C*
C*    END OF REPORTS - WRITE INSTRUCTIONS
        WRITE(6,906)
```

```
C*
C*
C*
      STOP
C*
C*       ERROR - NOT ENOUGH RECORDS FOR NUMBER OF CASES.
C*
  700 WRITE(6,*)"NOT ENOUGH RECORDS FOR NUMBER OF CASES. ABORT RUN."
      GO TO 300
C*
C*          ERROR -- NO END CARD
C*
  800 IF(INERR,EQ.0) THEN
          N=-99
          WRITE(6,*)"NO END CARD,  FILE CORRECTED,"
          WRITE(10,901)N
          GO TO 600
          END IF
      WRITE(6,*)"NO END CARD.  PREVIOUS ERRORS, RUN ABORTED."
      GO TO 300
  901 FORMAT(I3,A3,A15,2X,9I1)
  903 FORMAT(2(I3,3X),I2,2(3X,I3),A18 )
  904 FORMAT(1X,'ERROR IN INPUT DECK.  CASE=',I3,'   IDENT=', I3,
     1'   CONCEPT=',A3,'   RUN ABORTED - ERROR CHECKING CONTINUES')
  905 FORMAT(A18)
  906  FORMAT(///1X,"IF THE CORRELATION IS ,90 OR BETTER, YOU CAN BEGIN
     1CONSTRUCTION OF YOUR MODEL."//1X,"IF NOT PLEASE CHECK YOUR DATA CA
     2REFULLY ")
  950 FORMAT(1X,A8)
      END
C*
      SUBROUTINE SDEPA
C*****************************************************************
C*                                                               *
C*                        SUBROUTINE SDEPA                       *
C*                                                               *
C*                                                               *
C*      THIS SUBROUTINE USES THE OSGOOD SEMANTIC DIFFERENTIAL THEORY,
C*      IT COMPUTES EPA SCORES (EVALUATION, POTENCY, ACTIVITY)   *
C*      BASED UPON SCORES OF SELECTED SUBSCALES, SCORES ARE      *
C*      AVERAGED,                                                *
C*                                                               *
C*      THE PROGRAM CAN ALSO COMPUTE THE SUM, SUMS OF SQUARES,   *
C*      THE STANDARD DEVIATIONS,STANDARD ERROR OF THE MEAN       *
C*      IF THESE ARE NEEDED, SEE BELOW FIRST RECORD OF INPUT     *
C*      FILE,                                                    *
C*                                                               *
C*      VARIABLES N1 THRU N9 CONTAIN SUBSCALE NUMBERS OF         *
C*      VARIABLES COMPOSING AVERAGE E-VALUE, P-VALUE, A-VALUE.   *
C*           N1=1ST E SUBSCALE NUMBER                            *
C*           N2=2ND E SUBSCALE NUMBER                            *
C*           N3=3RD E SUBSCALE NUMBER                            *
C*           N4=1ST P SUBSCALE NUMBER                            *
C*           N5=2ND P SUBSCALE NUMBER                            *
```

```
C*          N6=3RD P SUBSCALE NUMBER                                    *
C*          N7=1ST A SUBSCALE NUMBER                                    *
C*          N8=2ND A SUBSCALE NUMBER                                    *
C*          N9=3RD A SUBSCALE NUMBER                                    *
C*                                                                      *
C*     THE INPUT DATA IS A DISK FILE (SEMDIF/INPUT) CONTAINING          *
C*     THE FOLLOWING INFORMATION:                                       *
C*                                                                      *
C*     RECORD    COL      CONTENTS                                      *
C*       1       1-3      NUMBER OF CASES                               *
C*               4-6      SPACE                                         *
C*               7-9      NUMBER OF CONCEPTS                            *
C*              10-12     SPACE                                         *
C*              13-14     PRINT CODE                                    *
C*                        01 - PRINTS MEANS AND MEANS DOUBLED           *
C*                        02 - PRINTS MEANS, MEANS DOUBLED, SUM,        *
C*                             SUM SQUARES, STANDARD DEVIATION,         *
C*                             STANDARD ERROR OF THE MEANS              *
C*              15-17     SPACE                                         *
C*              18-20     IDENT NUMBER OF 1ST CASE                      *
C*              21-23     SPACE                                         *
C*              24-26     IDENT NUMBER OF LAST CASE                     *
C*              27-28     SPACE                                         *
C*              29-42     FILE NAME                                     *
C*                                                                      *
C*       2       1-3      CASE IDENT NUMBER - NUMERIC                   *
C*       3       1-3      CASE IDENT NUMBER                             *
C*               4-6      CONCEPT IDENT NUMBER - NUMERIC                *
C*               7-21     CONCEPT NAME - ALPHA                          *
C*              22-23     SPACE                                         *
C*              24-32     SUBSCALE SCORES - NUMERIC                     *
C*                                                                      *
C*          REPEAT RECORD 3 AS NEEDED - 1 FOR EACH CONCEPT              *
C*                                                                      *
C*     LAST     1-3       -99                                          *
C*                                                                      *
C*                                                                      *
C*                                                                      *
C**********************************************************************
C*
C*
C*
C*
      INTEGER CASENO
      REAL MEANS2(109,3), SCORE(109,9)
      CHARACTER CONCNM(109)*15,TITLE*18
      REAL MEANS(109,3) ,NSUM(109,3),NSMSQ(109,3)
      DIMENSION STDEV(109,3), SEOM(109,3)
C*
      N1=1
      N2=6
      N3=7
      N4=2
      N5=5
```

```
      N6=8
      N7=3
      N8=4
      N9=9
      READ(10,901)L,N,IPSW,LTAG1,LTAG2,TITLE
      WRITE(6,901)L,N,IPSW,LTAG1,LTAG2,TITLE
C*
C*       INITIALIZE VARIABLES
C*
C*
    5 CONTINUE
      RNOP=0
      DO 11 J=1,3
         DO 11 IC=1,N
            NSUM(IC,J)=0
            STDEV(IC,J)=0
            SEOM(IC,J)=0
            MEANS(IC,J)=0
   11       NSMSQ(IC,J)=0
      DO 12 IC=1,N
         DO 12 J=1,9
   12       SCORE(IC,J)=0
      NC1=0
C*
C*
C*
C*
C*        READ IN ONE CASE IDENT RECORD PLUS N CONCEPT RECORDS
C*
C*
C*
   20 READ(10,902,END=550)  LTAG
      IF( NC1.EQ.0)NC1=LTAG
      If(LTAG.LT.0) GO TO 320
      ITAG =LTAG
      DO 200 K = 1,N
          READ(10,903)CASENO, IC, CONCNM(IC), (SCORE(IC,J),J=1,9)
          IF(L.EQ.1)WRITE(6,904)CASENO,CONCNM(IC),IC,(SCORE(IC,J),J=1,9)
C*    REVERSE  EVEN  SCALES
          DO 100 J=1,9,2
  100        SCORE(IC,J)=8 - SCORE(IC,J)
             SCORE(IC,1)=SCORE(IC,N1)+SCORE(IC,N2)+SCORE(IC,N3)
             SCORE(IC,1)=SCORE(IC,1)/3.
             SCORE(IC,2)=SCORE(IC,N4)+SCORE(IC,N5)+SCORE(IC,N6)
             SCORE(IC,2)=SCORE(IC,2)/3.
             SCORE(IC,3)=SCORE(IC,N7)+SCORE(IC,N8)+SCORE(IC,N9)
             SCORE(IC,3)=SCORE(IC,3)/3.
  200 CONTINUE
C*
C*
C*       COUNT NUMBER OF CASES
C*
C*
  300 RNOP=RNOP+1
```

```
      DO 310 J=1,3
        DO 310 IC=1,N
C*
C*    THIS COMPUTES THE SUMS
C*
            NSUM(IC,J)=NSUM(IC,J)+SCORE(IC,J)
C*
C*    THIS COMPUTES THE SUMS OF THE SQUARES
C*
  310       NSMSQ(IC,J)=(SCORE(IC,J)**2)+NSMSQ(IC,J)
      GO TO 20
  320 DO 330 J=1,3
        DO 330 IC=1,N
C*
C*    THIS COMPUTES THE MEAN
C*
            MEANS (IC,J)= NSUM(IC,J)/RNOP
C*
C*    THIS COMPUTES THE STANDARD DEVIATION
C*
            STDEV(IC,J)=((NSMSQ(IC,J)/RNOP)-(MEANS(IC,J)**2))**.5
C*
C*    THIS COMPUTES THE STANDARD ERROR OF THE MEANS
C*
            IF(RNOP.EQ.1)GO TO 330
            SEOM(IC,J)=STDEV(IC,J)/((RNOP-1)**.5)
  330       CONTINUE
C*
C*
C*
C*****************************************************************
C*                                                               *
C*        PRINT MEANS AND MEANS DOUBLED AND CREATE OUTPUT FILE   *
C*        (SEMDIF/D02) WITH THE FOLLOWING FORMAT:                *
C*          RECORD    COLS     CONTENTS                          *
C*            1       1-3      NUMBER OF CONCEPTS - NUMERIC      *
C*            2       1        SPACE                             *
C*                    2-4      CONCEPT IDENT NUMBER              *
C*                    5-11     MEAN DOUBLED OF E                 *
C*                   12-18     MEAN DOUBLED OF P                 *
C*                   19-25     MEAN DOUBLED OF A                 *
C*                   26-30     SPACE                             *
C*                   31-41     CONCEPT NAME - ALPHA              *
C*        3 - (N+1)            SAME AS RECORD 2 - ONE FOR EACH   *
C*                             CONCEPT                           *
C*                                                               *
C*****************************************************************
C*
C*
      WRITE(6,905)
      NC2=NC1+L-1
      WRITE(12,906)N,NC1,NC2
      WRITE(7,902)N
      WRITE(6,907)(IC,(MEANS(IC,J),J=1,3),CONCNM(IC),IC=1,N)
```

```
      WRITE(7,900)(IC,(MEANS(IC,J),J=1,3),CONCNM(IC),IC=1,N)
      DO 400 J=1,N
      DO 400 M=1,3
C*
C*
C*       CALCULATE MEANS2 - MEANS DOUBLED
C*
C*
  400    MEANS2(J,M)=MEANS(J,M)*2.0
      WRITE(12,900)(IC,(MEANS2(IC,J),J=1,3),CONCNM(IC),IC=1,N)
      WRITE(6,908)
      WRITE(6,907)(IC,(MEANS2(IC,J),J=1,3),CONCNM(IC),IC=1,N)
      GO TO (600,500) IPSW
C*              SUMS
  500 WRITE(6,909)
      WRITE(6,907) (IC,(NSUM(IC,J),J=1,03),CONCNM(IC),IC=1,N)
C*              SUMS OF THE SQUARES
      WRITE(6,910)
      WRITE(6,907)(IC,(NSMSQ(IC,J),J=1,03),CONCNM(IC),IC=1,N)
C*              STANDARD DEVIATIONS
      WRITE(6,911)
      WRITE(6,907)(IC,(STDEV(IC,J),J=1,03),CONCNM(IC),IC=1,N)
      WRITE(6,912)
      WRITE(6,907) (IC,(SEOM(IC,J),J=1,03),CONCNM(IC),IC=1,N)
      WRITE(7,920)   (IC,(SEOM(IC,J),J=1,03),IC=1,N)
  550 REWIND 10
      REWIND 12
      RETURN
  600 GO TO 5
  900 FORMAT(1X,I3,3F7.3,5X,A15)
  901 FORMAT(I3,3X,I3,3X,I2,2(3X,I3),A18)
  902 FORMAT(I3)
  903 FORMAT(I3,I3, A15,2X,9F1.0)
  904 FORMAT(1X,I3,2X,A15,2X,I3,3X,9F6.0)
  905 FORMAT(/////,1X,'THE MEANS ARE'/7X'E'6X'P'6X'A'/)
  906 FORMAT(3(I3,3X))
  907 FORMAT(1X,I3,'-', 3F8.3,5X,A15)
  908 FORMAT(///,1X'THE MEANS DOUBLED ARE'/7X'E'6X'P'6X'A'/)
  909 FORMAT(/////,1X,'THE SUMS ARE',/)
  910 FORMAT(/////,1X,'THE SUMS OF THE SQUARES ARE',/)
  911 FORMAT(/////,1X,'THE STANDARD DEVIATIONS ARE',/)
  912 FORMAT(/////,1X,'THE STANDARD ERRORS OF THE MEANS ARE',/)
  915 FORMAT(1X,3F8.3,5X,A15)
  920 FORMAT(1X,I3,3F7.3)
      END
C*
      SUBROUTINE SD2PTD
C*****************************************************************
C*                                                               *
C*                  SUBROUTINE SD2PTD                            *
C*                                                               *
C*                                                               *
C*     THE SUBROUTINE COMPUTES THE DISTANCE BETWEEN TWO CONCEPTS *
C*     (VARIABLES) AS DEVELOPED IN SUBROUTINE SDEPA.  SDEPA YIELDS *
```

```
C*      THREE VALUES ON THE EVALUATIVE, POTENCY AND ACTIVITY          *
C*      FACTORS OF THE SEMANTIC DIFFERENTIAL.                         *
C*                                                                    *
C*      THE SUBROUTINE WILL HANDLE UP TO 109 CONCEPTS, I.E. CONCEPT   *
C*      N1 TO CONCEPT N2 WHERE N2 = N1 + (N-1) AND CONCEPT K IS       *
C*      FOUND IN RECORD(K+1) IN THE INPUT FILE, SEMDIF/D01.  THE      *
C*      INPUT FILE IS A DISK FILE CREATED BY THE SUBROUTINE SDEPA     *
C*      WITH THE FOLLOWING FORMAT:                                    *
C*                                                                    *
C*         RECORD     COL     CONTENTS                               *
C*           1         1-3    NUMBER OF CONCEPTS                     *
C*           2         1      SPACE                                  *
C*                     2-4    CONCEPT IDENT NUMBER                   *
C*                     5-11   VALUE OF E                             *
C*                    12-18   VALUE OF P                             *
C*                    19-25   VALUE OF A                             *
C*                    26-30   SPACE                                  *
C*                    31-41   CONCEPT - ALPHA                        *
C*                    42-80   SPACE                                  *
C*        3-(N+1)     SAME AS RECORD 2 - ONE RECORD FOR EACH CONCEPT *
C*                                                                    *
C*                                                                    *
C*      OUTPUT FROM THIS PROGRAM IS A DISK FILE, SEMDIF/D02, FOR USE  *
C*      IN LRANK, THE CORRELATION SUBROUTINE.  THE VALUES RE-         *
C*      CORDED WILL BE THE DISTANCES BETWEEN CONCEPT 1 AND 2 TO N     *
C*      IN THE 1ST (N-1) RECORDS, CONCEPT 2 AND 3 TO N IN THE NEXT    *
C*      (N-2) RECORDS, CONCEPT 3 AND 4 TO N IN THE NEXT (N-3)         *
C*      RECORDS, ETC.                                                 *
C*                                                                    *
C*                                                                    *
C*                                                                    *
C**********************************************************************
C*
C*
      CHARACTER CONCNM(109)*15
      DIMENSION E(109),P(109),A(109),D(109,108),G(109)
C*
      READ(12,900)N,N1,N2
      WRITE(14,900)N,N1,N2
      M=N-1
      DO 5 I=1,N
         READ(12,930)IC,E(I),P(I),A(I),CONCNM(I)
    5    CONTINUE
C*
C*       WRITE HEADING
C*
      WRITE(6,920)
      DO 15 I=1,M
           DO 15 K=1,N
   15         D(K,I)   = 0.0
      DO 20 K=1,M
         L=K+1
         DO 20 J=L,N
            I=J-1
```

```
   20     D(K,I)=SQRT(((E(J)-E(K))**2)+((P(J)-P(K))**2)+((A(J)-A(K))**2))
       J=0
       DO 110 K=1,M
          DO 100 I=K,M
             J=J+1
             G(J)=D(K,I)
C*
C*
C*        WRITE DISTANCE MATRIX TO DISK FILE (SEMDIF/D02)
C*
C*        PRINT OUT DISTANCE MATRIX
C*
C*
  100        WRITE(14,940)G(J)
          L2=15
          DO 105 L=1,J,15
             IF(L2.GT.J)L2=J
             WRITE(6,950)K, (G(J1),J1=L,L2)
  105        L2=L2+15
          WRITE(6,960)
  110     J=0
  900 FORMAT(3(I3,3X))
  920 FORMAT(///" THE DISTANCES BETWEEN THE CONCEPTS ARE SHOWN BELOW"/
     *" FIRST GROUP OF ROWS SHOWS CONCEPT 01 AGAINST 02, 03, 04 ETC."/
     *" SECOND GROUP OF ROWS SHOWS CONCEPT 02 AGAINST 03, 04, 05, ETC."/
     *" THIRD GROUP SHOWS CONCEPT 03 AGAINST  THE OTHERS. FOURTH GROUP-
     *CONCEPT 04 ETC. --"//
     *16X,"2      3      4      5      6      7      8      9     10     11     12
     *    13     14     15     16 "//)
  930 FORMAT(1X,I3,3F7.3,5X,A15)
  940 FORMAT(F6.2)
  950 FORMAT(1X,"CONC ",I3,3X, 15F6.2)
  960 FORMAT(1X,//)
       REWIND 14
       RETURN
       END
C*
       SUBROUTINE SDMSM
C*****************************************************************************
C*                                                                           *
C*                    SUBROUTINE SDMSM                                       *
C*                                                                           *
C*                                                                           *
C*                                                                           *
C*                                                                           *
C*     THIS SUBROUTINE COMPUTES THE MEAN, MEAN DOUBLED, THE SUM,             *
C*     THE SUM OF THE SQUARES, THE STANDARD DEVIATIONS, STANDARD             *
C*     ERROR OF THE MEANS FOR SUB-SCALES 1-9 ON THE OSGOOD SEMAN-            *
C*     TIC DIFFERENTIAL.  IT WILL PROCESS UP TO AND INCLUDING                *
C*     109 CONCEPTS AND WILL PRINT OUT RESULTS ACCORDING TO PRINT            *
C*     CODE DESCRIBED BELOW.                                                 *
C*                                                                           *
C*     THE INPUT DATA IS A DISK FILE (SEMDIF/D01) CONTAINING                 *
C*      THE FOLLOWING INFORMATION:                                           *
C*                                                                           *
```

```
C*      RECORD     COL       CONTENTS                                    *
C*        1         1-3      NUMBER OF CASES                             *
C*                  4-6      SPACE                                       *
C*                  7-9      NUMBER OF CONCEPTS                          *
C*                 10-12     SPACE                                       *
C*                 13-14     PRINT CODE                                  *
C*                           01 - PRINTS MEANS                           *
C*                           02 - PRINTS MEANS, SUM, SUM SQUARES,        *
C*                                STANDARD DEVIATION, STANDARD           *
C*                                ERROR OF THE MEANS                     *
C*                 15-17     SPACE                                       *
C*                 18-20     IDENT NUMBER OF 1ST CASE                    *
C*                 21-23     SPACE                                       *
C*                 24-26     IDENT NUMBER OF LAST CASE                   *
C*                 27-28     SPACE
C*                 29-42     FILE NAME
C*                                                                       *
C*          2       1-3      CASE IDENT NUMBER - NUMERIC                 *
C*          3       1-3      CASE IDENT NUMBER                           *
C*                  4-6      CONCEPT IDENT NUMBER - NUMERIC              *
C*                  7-21     CONCEPT NAME - ALPHA                        *
C*                 22-23     SPACE                                       *
C*                 24-32     SUBSCALE SCORES - NUMERIC
C*
C*                                                                       *
C*              REPEAT RECORD 3 AS NEEDED - 1 FOR EACH CONCEPT           *
C*                                                                       *
C*           REPEAT THIS RECORD GROUP, ONE FOR EACH CASE, ENDING         *
C*           WITH A FINISH RECORD FOLLOWING THE LAST CONCEPT OF          *
C*           THE LAST CASE.  (GROUP=RECORDS 2 AND 3)                     *
C*                                                                       *
C*       LAST        1-3       -99                                       *
C*                                                                       *
C*                                                                       *
C*************************************************************************
C*
C*
      REAL SCORE(109,9)
      REAL MEANS(109,9) ,NSUM(109,9),NSMSQ(109,9),MEANS2(109,9)
      DIMENSION STDEV(109,9),SEOM(109,9)
      CHARACTER CONCNM(109)*15
C*
      READ(10,900)L,N,IPSW
       IF(L.EQ.1) THEN
           ASSIGN 500 TO NUM
           ELSE
           ASSIGN 600 TO NUM
           END IF
C*
C*
C*       INITIALIZE VARIABLES
C*
C*
      RNOP=0
```

```
      DO 100 J=1,9
         DO 100 IC=1,N
         NSUM(IC,J)=0
         SCORE(IC,J)=0
         STDEV(IC,J)=0
         SEOM(IC,J)=0
         MEANS(IC,J)=0
         MEANS2(IC,J)=0
  100    NSMSQ(IC,J)=0
      NC1=0
C*
C*
C*       READ INPUT DATA
C*
C*
  200 READ(10,910,END=800)  LTAG
      IF(NC1.EQ.0)NC1=LTAG
      IF(LTAG.LT.0) GO TO 320
      DO 210 K = 1,N
      READ(10,930)IC, CONCNM(IC), (SCORE(IC,J),J=1,9)
      DO 205 J=1,9,2
  205    SCORE(IC,J)=8-SCORE(IC,J)
  210 CONTINUE
C*
C*       COUNT THE NUMBER OF CASES
C*
C*
  300 RNOP=RNOP+1
      DO 310 J=1,9
        DO 310 IC=1,N
C*    THIS COMPUTES THE SUMS
           NSUM(IC,J)=NSUM(IC,J)+SCORE(IC,J)
C*    THIS COMPUTES THE SUMS OF THE SQUARES
  310      NSMSQ(IC,J)=(SCORE(IC,J)**2)+NSMSQ(IC,J)
      GO TO 200
  320 DO 400 J=1,9
         DO 400 IC=1,N
C*    THIS COMPUTES THE MEAN
          MEANS (IC,J)= NSUM(IC,J)/RNOP
C*    THIS COMPUTES THE STANDARD DEVIATION
          STDEV(IC,J)=((NSMSQ(IC,J)/RNOP)-(MEANS(IC,J)**2))**.5
C*    THIS COMPUTES THE STANDARD ERROR OF THE MEANS
          IF(RNOP.EQ.1)GO TO 400
          SEOM(IC,J)=STDEV(IC,J)/((RNOP-1)**.5)
  400     CONTINUE
       WRITE(6,935)
      GO TO NUM,(500,600)
  500 WRITE(6,940)
C*
C*****************************************************************
C*                                                               *
C*         WRITE 4 GROUPS OF OUTPUT AND CREATE DISK FILE,        *
C*         SEMDIF/D03 WITH THE FOLLOWING FORMAT:                 *
C*                                                               *
```

```
C*         RECORD     COL     CONTENTS                                    *
C*           1         1-3    NUMBER OF CONCEPTS                          *
C*           2         1      SPACE                                       *
C*                     2-4    NUMBER OF CONCEPTS                          *
C*                     5-11   VALUE OF 1ST MEAN , XXX.XXX                 *
C*                    12-18   VALUE OF 2ND MEAN  XXX.XXX                  *
C*                    19-67   VALUES OF 3RD THRU 9TH MEANS                *
C*                    68-69.  SPACE                                       *
C*                    70-80   CONCEPT - ALPHA                             *
C*           3       SAME FORMAT AS RECORD 2 BUT FOR 2ND CONCEPT          *
C*        4-(N+1)    INFORMATION FOR REMAINING CONCEPTS                   *
C*                                                                        *
C**************************************************************************
C*
      GO TO 620
  600 WRITE(6,945)
C*              MEANS
  620 WRITE(6,950)(IC,(MEANS(IC,J),J=1,9),CONCNM(IC),IC=1,N)
      NC2=NC1+L-1
      WRITE(16,955)N,NC1,NC2
      WRITE(16,960)(IC,(MEANS(IC,J),J=1,9),CONCNM(IC),IC=1,N)
      IF(IPSW.NE.2) GO TO 700
C*
C*THESE STATEMENTS ARE EXECUTED WITH PRINT CODE 2
C*              SUMS
      WRITE(6,965)
      WRITE(6,970)   (IC,(NSUM(IC,J),J=1,9),IC=1,N)
C*              SUMS OF THE SQUARES
      WRITE(6,975)
      WRITE(6,980)   (IC,(NSMSQ(IC,J),J=1,9),IC=1,N)
C*              STANDARD DEVIATIONS
      WRITE(6,985)
      WRITE(6,990) (IC,(STDEV(IC,J),J=1,9),IC=1,N)
      WRITE(6,995)
      WRITE(6,990)   (IC,(SEOM (IC,J),J=1,9),IC=1,N)
  700 REWIND 16
      RETURN
  800 STOP
  900 FORMAT(I3,3X,I3,3X,I2)
  910 FORMAT(I3)
  930 FORMAT(3X,I3,,A15,2X,9F1.0)
  935 FORMAT(1H1,/" COMPUTER PRINTOUT FOR THE SEMANTIC DIFFERENTIAL"//)
  940 FORMAT(1H@,"THE SCORES OF THE SUBSCALES ARE"/)
  945 FORMAT (1H ,"THE MEANS OF THE SUBSCALES ARE "//)
  950 FORMAT(1X,I3,"-",9F7.3,5X,A15)
  955 FORMAT(3(I3,3X))
  960 FORMAT(1X,I3,9F7.3,2X,A15)
  965 FORMAT(/////,1X,"THE SUMS ARE",/)
  970 FORMAT(1X,I3,"-",9F6.0)
  975 FORMAT(/////,1X,"THE SUMS OF THE SQUARES ARE",/)
  980 FORMAT(1X,I3,"-",9F8.0)
  985 FORMAT(/////,1X,"THE STANDARD DEVIATIONS ARE",/)
  990 FORMAT(1X,I3,"-",9F7.3)
  995 FORMAT(/////,1X,"THE STANDARD ERRORS OF THE MEANS ARE",/)
```

```
      END
C*
      SUBROUTINE SDGPD
C*****************************************************************
C*                                                               *
C*                                                               *
C*                      SUBROUTINE SDGPD                         *
C*                                                               *
C*                                                               *
C*     THIS SUBROUTINE COMPUTES THE D VALUES FOR THE SEMANTIC    *
C*     DIFFERENTIAL USING THE GENERALIZED DISTANCE FORMULA:      *
C*                                                               *
C*               D=SQUARE ROOT OF SUM OF D SQUARED               *
C*                                                               *
C*               (SEE OSGOOD, SUCI & TANNENBAUM P.91)            *
C*                                                               *
C*     DISTANCES ARE COMPUTED BETWEEN EACH CONCEPT AND EVERY OTHER *
C*     CONCEPT.  THIS PROGRAM WILL HANDLE UP TO 109 CONCEPTS, I.E. *
C*     CONCEPT N1 TO CONCEPT N2 WHERE N2=N1+N-1 AND CONCEPT K IS  *
C*     FOUND IN RECORD (K+1) IN THE INPUT FILE.                  *
C*                                                               *
C*     DATA INPUT IS THE DISK FILE (SEMDIF/D03), CREATED BY SUBROUTINE *
C*     SDMSM, WHICH IS THE MEANS OF THE SUBSCALES.  IT IS STORED IN *
C*     THE FOLLOWING FORMAT:                                     *
C*                                                               *
C*        RECORD     COL     CONTENTS                           *
C*          1        1-3     NUMBER OF CONCEPTS                 *
C*          2        1       BLANK                              *
C*                   2-4     CONCEPT IDENT NUMBER - INTEGER     *
C*                   5-11    VALUE OF MEAN OF SUBSCALE 1        *
C*                  12-18    VALUE OF MEAN OF SUBSCALE 2        *
C*                  19-25    VALUE OF MEAN OF SUBSCALE 3        *
C*                  26-32    VALUE OF MEAN OF SUBSCALE 4        *
C*                  33-39    VALUE OF MEAN OF SUBSCALE 5        *
C*                  40-46    VALUE OF MEAN OF SUBSCALE 6        *
C*                  47-53    VALUE OF MEAN OF SUBSCALE 7        *
C*                  54-60    VALUE OF MEAN OF SUBSCALE 8        *
C*                  61-67    VALUE OF MEAN OF SUBSCALE 9        *
C*                  68-69    SPACE                              *
C*                  70-80    CONCEPT NAME - ALPHA               *
C*                                                               *
C*        REPEAT RECORD 2 AS NEEDED - ONE FOR EACH CONCEPT       *
C*                                                               *
C*        RECORD 2 HAS THE MEANS OF THE SUBSCALES FOR THE 1ST    *
C*        CONCEPT, RECORD 3 FOR THE 2ND CONCEPT, RECORD 4 FOR THE *
C*        3RD CONCEPT AND SO ON UNTIL ALL CONCEPTS (N1 TO N2) HAVE *
C*        BEEN INCLUDED,                                         *
C*                                                               *
C*     OUTPUT FROM THIS SUBROUTINE IS A DISK FILE, SEMDIF/D04, FOR *
C*     USE IN LRANK, THE CORRELATION SUBROUTINE.  THE VALUES RECORDED *
C*     WILL BE THE DISTANCES BETWEEN CONCEPT 1 AND (2 TO N) IN THE *
C*     1ST (N-1) RECORDS, CONCEPT 2 AND (3 TO N) IN THE NEXT (N-2) *
C*     RECORDS, CONCEPT 3 AND (4 TO N) IN THE NEXT (N-3) RECORDS, ETC. *
C*                                                               *
```

```
C*                                                                      *
C*                                                                      *
C*************************************************************************
C*
C*
C*
C*
      REAL MEAN(109)
      DIMENSION BA(109,9),SUM (109,108)
      CHARACTER CONCNM(109)*15
C*    M=NUMBER OF CONCEPTS RATED
      READ(16,900)M,N1,N2
      WRITE(17,900)M,N1,N2
      MM=M-1
      DO 40 I=1,M
         DO 40 J=1,MM
            SUM(J,I)=0
   40       MEAN(I)=0
C*
C*       READ IN INPUT FILE
C*
      DO 100 I=1,M
         READ(16,910)IC,(BA(I,J),J=1,9),CONCNM(I)
  100    CONTINUE
      WRITE(6,950)
      DO 210 K=1,MM
         L=K+1
         DO 210 J=L,M
            I=J-1
            DO 200 NN=1,9
  200          SUM(K,I)=((BA(J,NN)-BA(K,NN))**2) + SUM(K,I)
  210       SUM(K,I)=SQRT(SUM(K,I))
C*
C*
C*       WRITE HEADING
C*
C*
      WRITE(6,920)
      DO 500 K=1,MM
         J=0
         DO 400 I=K,MM
            J=J+1
  400       MEAN(J)=SUM(K,I)
         L2=15
C*
C*
C*       WRITE DISTANCE MATRIX TO DISK FILE (SEMDIF/D04)
C*
C*       PRINT OUT DISTANCE MATRIX
C*
C*
         WRITE(17,930)(MEAN(L),L=1,J)
         DO 450 L=1,J,15
            IF(L2.GT.J)L2=J
```

```
            WRITE(6,940)K,(MEAN(J1),J1=L,L2)
            WRITE(7,960)(MEAN(J1),J1=L,L2)
  450       L2=L2+15
          WRITE(6,950)
  500     CONTINUE
      REWIND 17
      CLOSE (7,STATUS="NEW")
      RETURN
      STOP
  900 FORMAT(3(I3,3X))
  905 FORMAT(1X,"CONC",3X,"1",6X,"2",6X,"3",6X,"4",6X,"5",6X,"6",6X,
     *"7",6X,"8",6X,"9"/" NO."/)
  910 FORMAT(1X,I3,9F7.3,2X,A15)
  920 FORMAT(" THE OSGOOD DEES BETWEEN THE CONCEPTS ARE SHOWN BELOW"/
     *" FIRST GROUP OF ROWS SHOWS CONCEPT 01 AGAINST 02, 03, 04 ETC."/
     *" SECOND GROUP OF ROWS SHOWS CONCEPT 02 AGAINST 03, 04, 05 ETC."/
     *" THIRD GROUP SHOWS CONCEPT 03 AGAINST THE OTHERS. FOURTH GROUP-
     * CONCEPT 04 ETC. --"//
     *16X,"2      3      4      5      6      7      8      9     10     11     12
     *     13     14     15      16"//)
  930 FORMAT(F6.2)
  940 FORMAT(1X,"CONC ",I3,3X,15F6.2)
  950 FORMAT(1X,//)
  960 FORMAT(1X,15F6.2)
      END
C*
      SUBROUTINE LRANK
C*****************************************************************************
C*                                                                           *
C*                                                                           *
C*                      SUBROUTINE LRANK                                     *
C*           (CORRELATION BETWEEN DEES AND EPA DISTANCES)                    *
C*                                                                           *
C*      THIS SUBROUTINE RANKS THE D VALUES COMPUTED IN SUBRQUTINE            *
C*      SDGPD AND THE EPA DISTANCES COMPUTED IN SUBROUTINE SD2PTD.           *
C*      IT THEN COMPUTES THE SPEARMAN RANK CORRELATION COEFFICIENT           *
C*      FOR THE TWO GROUPS OF DATA.  IT HANDLES A DISTANCE MATRIX            *
C*      FOR UP TO 109 CONCEPTS.                                              *
C*                                                                           *
C*      DATA IS INPUT FROM 2 DISK FILES:                                     *
C*           SEMDIF/D02 - CREATED BY SUBROUTINE SD2PTD                       *
C*           SEMDIF/D04 - CREATED BY SUBROUTINE SDGPD                        *
C*                                                                           *
C*      BOTH FILES HAVE THE FOLLOWING FORMAT:                                *
C*                                                                           *
C*           RECORD     CONTENTS                                             *
C*           1-(N-1)    DISTANCES BETWEEN THE 1ST CONCEPT AND CONCEPTS       *
C*                      2 TO N - XXX.XX                                      *
C*           N-(2N-3)   DISTANCES BETWEEN THE 2ND CONCEPT AND CONCEPTS       *
C*                      3 TO N - XXX.XX                                      *
C*                .                                                          *
C*                .                                                          *
C*           N(N-1)/2   DISTANCE BETWEEN THE (N-1) CONCEPT AND CONCEPT N     *
C*                                                                           *
```

```
C*      THE RESULTS OF THIS SUBROUTINE PLUS THE FIRST TEN VALUES         *
C*      FROM EACH OF THE ABOVE INPUT FILES ARE SENT TO THE PRINTER.      *
C*                                                                       *
C*      THE SUBROUTINE USES THE IBM SUBROUTINES RANK, TIE, AND SRANK.    *
C*                                                                       *
C*                                                                       *
C*************************************************************************
C*
C*
C*
      DIMENSION A(5886),B(5886),R(11772)
C*
C*
C*        K = K3 = NUMBER OF CONCEPTS TO BE PROCESSED
C*        K1= K4 = NUMBER OF 1ST CASE
C*        K2= K5 = NUMBER OF LAST CASE
C*
C*
      READ(14,900)K,K1,K2
      READ(17,900)K3,K4,K5
      WRITE(6,910)K,K1,K2
      NR=0
      N=(K*(K-1))/2
C*
C*
C*        READ IN INPUT DATA
C*
C*                 ARRAY A=SD2PTD
C*                 ARRAY B=SDGPD
C*
C*
      DO 100 I=1,N
         READ(14,920) A(I)
  100    READ(17,920) B(I)
      WRITE(6,940)K,K, K1,K2
      DO 200 I=1,10
  200    WRITE(6,950)A(I),B(I)
      CALL SRANK (A,B,R,N,RS,T,NDF,NR)
      WRITE(6,960) RS,T,NDF
      RETURN
  900 FORMAT(3(I3,3X))
  910 FORMAT(1X,3(I3,3X)//)
  920 FORMAT ( F6.2)
  940 FORMAT(" NO. OF CONCEPTS=",I3,",  CONCEPT NO. 1 TO CONCEPT NO. ",
     *I3,"    CASE NO. ",I3," TO CASE NO. ",I3//)
  950 FORMAT(1X,"A(I)=",F6.2,3X,"B(I)=",F6.2)
  960 FORMAT ("0 SPEARMAN RANK CORRELATION COEFFICIENT=",F5.2/
     *" SIGNIFICANCE=",F8.3/" NUMBER OF DEGREES OF FREEDOM=",I5)
      END
      SUBROUTINE SRANK(A,B,R,N,RS,T,NDF,NR)
C*************************************************************************
C*
C*    SUBROUTINE SRANK
C*
```

```
C*    PURPOSE
C*       TEST CORRELATION BETWEEN TWO VARIABLES BY MEANS OF SPEARMAN
C*       RANK CORRELATION COEFFICIENT
C*
C*    USAGE
C*       CALL SRANK(A,B,R,N,RS,T,NDF,NR)
C*
C*    DESCRIPTION OF PARAMETERS
C*       A    - INPUT VECTOR OF N OBSERVATIONS FOR FIRST VARIABLE
C*       B    - INPUT VECTOR OF N OBSERVATIONS FOR SECOND VARIABLE
C*       R    - OUTPUT VECTOR FOR RANKED DATA, LENGTH IS 2*N. SMALLEST
C*              OBSERVATION IS RANKED 1, LARGEST IS RANKED N. TIES
C*              ARE ASSIGNED AVERAGE OF TIED RANKS.
C*       N    - NUMBER OF OBSERVATIONS
C*       RS   - SPEARMAN RANK CORRELATION COEFFICIENT (OUTPUT)
C*       T    - TEST OF SIGNIFICANCE OF RS (OUTPUT)
C*       NDF  - NUMBER OF DEGREES OF FREEDOM (OUTPUT)
C*       NR   - CODE, 0 FOR UNRANKED DATA IN A AND B, 1 FOR RANKED
C*              DATA IN A AND B (INPUT)
C*
C*    REMARKS
C*       T IS SET TO ZERO IF N IS LESS THAN TEN
C*
C*    SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C*       RANK
C*       TIE
C*
C*    METHOD
C*       DESCRIBED IN S. SIEGEL, "NONPARAMETRIC STATISTICS FOR THE
C*       BEHAVIORAL SCIENCES", MCGRAW-HILL, NEW YORK, 1956,
C*       CHAPTER 9
C**********************************************************************
C
      DIMENSION A(1),B(1),R(1)
C
      FN=FLOAT(N)
      FNNN=(FN**3)-FN
C
C*       DETERMINE WHETHER DATA IS RANKED
C
      IF(NR-1) 5, 10, 5
C*
C*       RANK DATA IN A AND B VECTORS AND ASSIGN TIED OBSERVATIONS
C*       AVERAGE OF TIED RANKS
C
      NS=0
    5 CALL RANK (A,R,N,RS,T,NDF,NS)
      NS=N
      CALL RANK (B,R,N,RS,T,NDF,NS)
      GO TO 40
C*
C*       MOVE RANKED DATA TO R VECTOR
C*
   10 DO 20 I=1,N
```

```
   20 R(I)=A(I)
      DO 30 I=1,N
      J=I+N
   30 R(J)=B(I)
C*
C*       COMPUTE SUM OF SQUARES OF RANK DIFFERENCES
C*
   40 D=0.0
      DO 50 I=1,N
      J=I+N
   50 D=D+(R(I)-R(J))*(R(I)-R(J))
C*
C*       COMPUTE TIED SCORE INDEX
C*
      KT=1
      NS=0
      CALL TIE (R,N,KT,TSA.NS)
      NS=N
      CALL TIE (R,N,KT,TSB,NS)
C*
C*       COMPUTE SPEARMAN RANK CORRELATION COEFFICIENT
C*
      IF(TSA) 60,55,60
   55 IF(TSB) 60,57,60
   57 RS=1.0-6.0*D/FNNN
      GO TO 70
   60 X=FNNN/12.0-TSA
      Y=X+TSA-TSB
      RS=(X+Y-D)/(2.0*(SQRT(X*Y)))
      WRITE(6,900)FNNN,X,Y,D
C*
C*       COMPUTE T AND DEGREES OF FREEDOM IF N IS 10 OR LARGER
C*
      T=0.0
   70 IF(N-10) 80,75,75
   75 CONTINUE
      T=RS*SQRT(FLOAT(N-2)/(1.0-RS*RS))
   80 NDF=N-2
      RETURN
  900 FORMAT(1X,"FNNN=",G10.4,"  X=",G10.4,"  Y=",G10.4,"  D=",G10.4)
      END
      SUBROUTINE TIE(R,N,KT,T,NS)
C*
C*************************************************************************
C*     SUBROUTINE TIE
C*
C*     PURPOSE
C*        CALCULATE CORRECTION FACTOR DUE TO TIES
C*
C*     USAGE
C*        CALL TIE(R,N,KT,T)
C*
C*     DESCRIPTION OF PARAMETERS
C*        R  - INPUT VECTOR OF RANKS OF LENGTH N CONTAINING VALUES
```

```
C*              1 TO N
C*        N  - NUMBER OF RANKED VALUES
C*        KT   - INPUT CODE FOR CALCULATION OF CORRECTION FACTOR
C*                     1   SOLVE EQUATION 1
C*                     2   SOLVE EQUATION 2
C*        T  - CORRECTION FACTOR (OUTPUT)
C*                  EQUATION 1   T=SUM(CT**3-CT)/12
C*                  EQUATION 2   T=SUM(CT*(CT-1)/2)
C*                WHERE CT IS THE NUMBER OF OBSERVATIONS TIED FOR A
C*                     GIVEN RANK
C*     REMARKS
C*        NONE
C*
C*     SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C*        NONE
C*
C*     METHOD
C*        VECTOR IS SEARCHED FOR SUCCESSIVELY LARGER RANKS. TIES ARE
C*        COUNTED AND CORRECTION FACTOR 1 OR 2 SUMMED.
C*
C*************************************************************************
C*
      DIMENSION R(1)
C*
C*        INITIALIZATION
C*
      T=0.0
      Y=0.0
    5 X=1.0E38
      IND=0
C*
C*        FIND NEXT LARGEST RANK
C*
      DO 30 I=1,N
      IF(R(NS+I)-Y) 30,30,10
   10 IF(R(NS+I)-X) 20,30,30
   20 X=R(NS+I)
      IND=IND+1
   30 CONTINUE
C*
C*        IF ALL RANKS HAVE BEEN TESTED, RETURN
C*
      IF(IND) 90,90,40
   40 Y=X
      CT=0.0
C*
C*        COUNT TIES
C*
      DO 60 I=1,N
      IF(R(NS+I)-X) 60,50,60
   50 CT=CT+1.0
   60 CONTINUE
C*
C*        CALCULATE CORRECTION FACTOR
```

```
C*
      IF(CT) 70,5,70
   70 IF(KT-1) 75,80,75
   75 T=T+CT*(CT-1.)/2.0
      GO TO 5
   80 T=T+(CT*CT*CT-CT)/12.0
      GO TO 5
   90 CONTINUE
      RETURN
      END
      SUBROUTINE RANK(A,R,N,RS,T,NDF,NS)
C
C***********************************************************************
C*       RANK A VECTOR OF VALUES
C*    USAGE
C*       CALL RANK(A,R,N)
C*    DESCRIPTION OF PARAMETERS
C*       A - INPUT VECTOR OF N VALUES
C*       R - OUTPUT VECTOR OF LENGTH N. SMALLEST VALUE IS RANKED 1,
C*           LARGEST IS RANKED N. TIES ARE ASSIGNED AVERAGE OF TIED
C*           RANKS
C*       N - NUMBER OF VALUES
C*    REMARKS
C*       NONE
C*    SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C*       NONE
C*    METHOD
C*       VECTOR IS SEARCHED FOR SUCCESSIVELY LARGER ELEMENTS. IF TIES
C*       OCCUR, THEY ARE LOCATED AND THEIR RANK VALUE COMPUTED.
C*       FOR EXAMPLE, IF 2 VALUES ARE TIED FOR SIXTH RANK, THEY ARE
C*       ASSIGNED A RANK OF 6.5 (=(6+7)/2)
C***********************************************************************
C*
      DIMENSION A(1),R(1)
C*
C*       INITIALIZATION
      DO 10 I=1,N
   10 R(NS+I)=0.0
C*
C*       FIND RANK OF DATA
C*
      DO 100 I=1,N
C*
C*       TEST WHETHER DATA POINT IS ALREADY RANKED
C*
      IF(R(NS+I)) 20, 20, 100
C*
C*       DATA POINT TO BE RANKED
C*
   20 SMALL=0.0
      EQUAL=0.0
      X=A(I)
      DO 50 J=1,N
      IF(A(J)-X) 30, 40, 50
```

```
C*        COUNT NUMBER OF DATA POINTS WHICH ARE SMALLER
C*
C*
   30 SMALL=SMALL+1.0
      GO TO 50
C*
C*        COUNT NUMBER OF DATA POINTS WHICH ARE EQUAL
C*
   40 EQUAL=EQUAL+1.0
         R(NS+J)=-1.0
   50 CONTINUE
C*
C*        TEST FOR TIE
C*
      IF(EQUAL-1.0) 60, 60, 70
C*
C*        STORE RANK OF DATA POINT WHERE NO TIE
C*
   60 R(NS+I)=SMALL+1.0
      GO TO 100
C*
C*        CALCULATE RANK OF TIED DATA POINTS
C*
   70 P=SMALL + (EQUAL + 1.0)*0.5
      DO 90 J=I,N
      IF(R(NS+J)+1.0) 90, 80, 90
   80 R(NS+J)=P
   90 CONTINUE
  100 CONTINUE
      RETURN
      END
      SUBROUTINE DATIT(WHEN)
      CHARACTER*8 WHEN
      CHARACTER*6 B
      DATA WHEN/"  /  /  "/
      B=DATE("MMDDYY")
      WHEN(1:2)=B(1:2)
      WHEN(4:5)=B(3:4)
      WHEN(7:8)=B(5:6)
      RETURN
      END
```

```
$RESET FREE SET LIST
FILE  5(KIND="REMOTE", MYUSE="IO")
FILE  6(KIND="PRINTER")
FILE  2(KIND="DISK",TITLE="SEMDIF/INPUT.",FILETYPE=7,STATUS="OLD")
FILE  7(KIND="DISK",TITLE="SEMDIF/OSGOOD/OUT.",MAXRECSIZE=7,
     *BLOCKSIZE=420,PROTECTION="SAVE",FLEXIBLE="TRUE",AREAS=20,
     *STATUS="NEW")
C*****************************************************************
C*                                                               *
C*                                                               *
C*            SEMDIF/OSGOOD - SOURCE                             *
C*                                                               *
C*          OBJECT/SEMDIF/OSGOOD - OBJECT                        *
C*                                                               *
C*                                                               *
C*    THIS PROGRAM COMPUTES OSGOOD D (DISTANCE) VALUES FOR EACH  *
C*    RATED CONCEPT AND OTHER CONCEPTS ON THE SUBSCALES IN       *
C*    PREPARATION FOR USE WITH THE WILCOXON OR OTHER RANK        *
C*    STATISTICAL TEST.  IT WAS DEVELOPED BY BARBARA L. METIVIER *
C*    AND EDWIN D. LAWSON AT THE COMPUTER CENTER, STATE UNIVERSITY*
C*    COLLEGE AT FREDONIA, N.Y. 14063.                           *
C*                                                               *
C*    THE PROGRAM IS SET UP TO HANDLE 6 CONTROL CONCEPTS WHOSE   *
C*    IDENT NUMBERS ARE THE LAST 6 NUMBERS OF THE CONCEPT LIST   *
C*    (I.E. IF THERE ARE 106 CONCEPTS, THE RATED ONES ARE 101-106).*
C*                                                               *
C*     THE INPUT FILE (SEMDIF/INPUT) HAS THE FOLLOWING FORMAT:   *
C*                                                               *
C*       RECORD     COL       CONTENTS                          *
C*         1        1-3       NUMBER OF CASES                   *
C*                  4-6       SPACE                             *
C*                  7-9       NUMBER OF CONCEPTS                *
C*                 10-12      SPACE                             *
C*                 13-14      PRINT CODE (NOT USED BY THIS PROGRAM)
C*                            01 - PRINTS MEANS AND MEANS DOUBLED
C*                            02 - PRINTS MEANS, MEANS DOUBLED,SUM,
C*                                 SUM SQUARES,STANDARD DEVIATION,
C*                                 STANDARD ERROR OF THE MEANS   *
C*                 15-17      SPACE                             *
C*                 18-20      IDENT NUMBER OF 1ST CASE          *
C*                 21-23      SPACE                             *
C*                 24-26      IDENT NUMBER OF LAST CASE         *
C*                 27-28      SPACE                             *
C*                 29-42      FILE NAME                         *
C*                                                              *
C*         2        1-3       CASE IDENT NUMBER - NUMERIC       *
C*         3        1-3       CASE IDENT NUMBER                 *
C*                  4-6       CONCEPT IDENT NUMBER - NUMERIC    *
C*                  7-21      CONCEPT NAME - ALPHA              *
C*                 22-23      SPACE                             *
C*                 24-32      SUBSCALE SCORES - NUMERIC         *
C*                                                              *
C*              REPEAT RECORD 3 AS NEEDED - 1 FOR EACH CONCEPT  *
C*              (RECORDS NEEDED FOR ONE CASE).                  *
```

```
C*                                                                      *
C*            REPEAT ABOVE GROUP - ONE FOR EACH CASE - ENDING           *
C*            WITH A FINISH RECORD.                                     *
C*                                                                      *
C*      LAST        1-3       -99    (FINISH RECORD)                   *
C*                                                                      *
C*    THE OUTPUT FILE (SEMDIF/OSGOOD/OUTPUT) HAS THE FOLLOWING          *
C*    FORMAT:                                                           *
C*                                                                      *
C*       RECORD      COL       CONTENTS                                 *
C*         1          1-17     NAME OF 1ST CONTROL CONCEPT              *
C*                   18-34     NAME OF 2ND CONTROL CONCEPT              *
C*         2          1-17     NAME OF 3RD CONTROL CONCEPT              *
C*                   18-34     NAME OF 4TH CONTROL CONCEPT              *
C*         3          1-17     NAME OF 5TH CONTROL CONCEPT              *
C*                   18-34     NAME OF 6TH CONTROL CONCEPT              *
C*         4          1-3      IDENT NO. FOR 1ST CONTROL CONCEPT        *
C*                    4-6      IDENT NO. FOR 1ST NON-CONTROL CONCEPT    *
C*                    7-9      CASE NO. FOR 1ST NON-CONTROL CONCEPT     *
C*                   10-19     OSGOOD D BETWEEN ABOVE 2 CONCEPTS        *
C*                   20-22     BLANK
C*                   23-33     NAME OF 1ST NON-CONTROL CONCEPT          *
C*         5          1-3      IDENT NO. FOR 2ND CONTROL CONCEPT        *
C*                    4-33     SAME AS ABOVE                            *
C*         6          1-3      IDENT NO. FOR 3RD CONTROL CONCEPT        *
C*                    4-33     SAME AS ABOVE                            *
C*       7-9                   SAME AS ABOVE ONLY FOR 4TH,5TH,6TH       *
C*                             CONTROL CONCEPTS                         *
C*        10          1-3      IDENT NO. FOR 1ST CONTROL CONCEPT        *
C*                    4-6      IDENT NO. FOR 2ND NON-CONTROL CONCEPT    *
C*                    7-9      CASE NO. FOR 2ND NON-CONTROL CONCEPT     *
C*                   10-19     OSGOOD D BETWEEN ABOVE 2 CONCEPTS        *
C*                   20-22     BLANK                                    *
C*                   23-33     NAME OF 2ND NON-CONTROL CONCEPT          *
C*        11          1-3      IDENT NO. FOR 2ND CONTROL CONCEPT        *
C*                    4-33     SAME AS ABOVE (RECORD 7)                 *
C*      12-15                  SAME AS ABOVE ONLY FOR 3RD,4TH,5TH,      *
C*                             6TH CONTROL CONCEPTS                     *
C*                                                                      *
C*                REPEAT THESE GROUPS OF 6 RECORDS - ONE GROUP FOR      *
C*                EACH NON-CONTROL CONCEPT (RECORDS FOR 1 CASE).        *
C*                                                                      *
C*                REPEAT ABOVE GROUP (6 X M1 RECORDS) - 1 FOR EACH      *
C*                CASE.                                                 *
C*                                                                      *
C*             M=TOTAL NUMBER OF CONCEPTS                               *
C*             M1=TOTAL NUMBER OF NON-CONTROL CONCEPTS                  *
C*             N=TOTAL NUMBER OF CASES                                  *
C*             LTAG1=IDENT NUMBER OF 1ST CASE                           *
C*             LTAG2=IDENT NUMBER OF LAST CASE                          *
C*                                                                      *
C*                                                                      *
C************************************************************************
C*
```

```
C*
      INTEGER CNUMBR,CASE(109)
      CHARACTER*15 CONCNM(109), FIDIN*18, FIDOUT*24, FIDSRT*30
      DIMENSION BA(109,9), A(6)
C*
C*
C*       ENTER NAME OF INPUT FILE THROUGH THE TERMINAL.
C*
C*
      WRITE(5,*)"ENTER NAME OF INPUT FILE.  NO MORE THAN 17 CHARACTERS,
     *PLEASE"
      READ(5,906)FIDIN
      IF (FIDIN(18:).NE." ")THEN
         FIDIN(18:)="."
         GO TO 70
      ELSE
          DO 50 I=17,1,-1
   50        IF(FIDIN(I:I).NE." ")GO TO 60
   60     I=I+1
          FIDIN(I:I)="."
      END IF
C*
C*          OPEN INPUT FILE
C*
   70 OPEN (2, FILE=FIDIN)
C*
C*
C*       CREATE NAME OF OUTPUT FILE AND NAME OF SORTED OUTPUT FILE.
C*
C*
      I=I-1
      FIDOUT=FIDIN(1:I)//"/SD."
      I=I+3
      FIDSRT=FIDOUT(1:I)//"/SORT."
      WRITE(5,*)"THE FOLLOWING NAMES ARE YOUR FILE NAMES.  THE FIRST IS
     *YOUR INPUT FILE AND THE SECOND IS YOUR OUTPUT FILE. ",FIDIN,FIDOUT
C*
C*       OPEN OUTPUT FILE
C*
      OPEN(7,FILE=FIDOUT)
      READ(2,901) N,M,LTAG1,LTAG2
      B=0
      M1=M-6
C*
C*          FOR DEBUGGING PURPOSES ONLY, WRITE FILE IDENTIFICATION
C*          INFORMATION.
C*
C*    WRITE(6,903)N,M,LTAG1,LTAG2
C*    WRITE(6,920)M1
C*
      DO 230 II=1,N
          READ(2,902) ITAG
          DO 103 I=1,M
              READ(2,905)ICASE,IC,CONCNM(IC),(BA(IC,K),K=1,9)
```

```
                CASE(IC)=ICASE
                DO 103 JJ=1,9,2
                   BA(IC,JJ)=8-BA(IC,JJ)
  103           CONTINUE
          M2=M1+1
C*
C*       WRITE CONTROL CONCEPT NAMES
C*
              IF (II.GT.1) GO TO 185
              WRITE(7,930)CONCNM(M2),CONCNM(M2+1)
              WRITE(7,930)CONCNM(M2+2),CONCNM(M2+3)
              WRITE(7,930)CONCNM(M2+4),CONCNM(M2+5)
  185       DO 220 J=1,M1
              DO 190 L=1,6
  190            A(L)=0
              DO 210 I=M2,M
                 L=I-M1
                 DO 200 K=1,9
                    B=(BA(I,K)-BA(J,K))**2
  200               A(L)=A(L)+B
                 A(L)=SQRT(A(L))
                 WRITE(7,900)I,J,CASE(J),A(L),CONCNM(J)
  210            CONTINUE
C*
C*            FOR DEBUGGING PURPOSES ONLY, WRITE CONCEPT NAME AND
C*            CALCULATED DISTANCES.
C*
C*      WRITE(6,910)CONCNM(J),(A(L),L=1,6)
  220         CONTINUE
  230       CONTINUE
            STOP
  900 FORMAT(3(I3),F10.5,3X,A15)
  901 FORMAT(2(I3,3X),2X,2(3X,I3))
  902 FORMAT(I3)
  903 FORMAT(1X,2(I3,3X),2X,2(3X,I3))
  905 FORMAT(2I3,A15,2X,9F1.0)
  906 FORMAT(A18)
  910 FORMAT(1X,A15,6(4X,F10.5))
  920 FORMAT(1X,"THE OSGOOD D VALUES FOR EACH CONTROL CONCEPT AND ",
     1I3," OTHER CONCEPTS ARE SHOWN BELOW."//3X,"CONCEPTS",9X,"GOOD",
     211X,"BAD",9X,"STRONG",9X,"WEAK",9X,"ACTIVE",9X,"PASSIVE"//)
  930 FORMAT(1X,A15,1X,A15)
            END
```

```
$SET LIST
$RESET$FREE
$SET$AUTOBIND
$BIND FILSRT FROM OBJECT/STAT/SEMDIF/SORT
FILE  2(KIND=DISK,TITLE="SEMDIF/DEBUG.",MAXRECSIZE=14,BLOCKSIZE=420)
FILE  3(KIND=DISK,TITLE="DISKIN1",FILETYPE=7)
FILE  4(KIND=DISK,TITLE="DISKIN2",FILETYPE=7)
FILE  5(KIND=REMOTE,MYUSE=IO,MAXRECSIZE=80)
FILE  6(KIND=DISK,TITLE="DISKOUT1",MAXRECSIZE=14,BLOCKSIZE=420)
FILE  7(KIND=DISK,TITLE="DISKOUT2",MAXRECSIZE=14,BLOCKSIZE=420)
FILE  8(KIND=DISK,TITLE="DISKOUT3",MAXRECSIZE=14,BLOCKSIZE=420)
FILE  9(KIND=DISK,TITLE="DISKOUT4",MAXRECSIZE=14,BLOCKSIZE=420)
FILE  10(KIND=PRINTER)
FILE  11(KIND=DISK,TITLE="DISKSRT1",MAXRECSIZE=7,BLOCKSIZE=420)
FILE  12(KIND=DISK,TITLE="DISKSRT2",MAXRECSIZE=7,BLOCKSIZE=420)
FILE  14(KIND=DISK,TITLE="DEBUG",MAXRECSIZE=14,BLOCKSIZE=420)
FILE  15(KIND=PRINTER)
C*
C*************************************************************************
C*                                                                       *
C*                                                                       *
C*                    SEMDIF/SIGTEST - SOURCE                            *
C*                                                                       *
C*             OBJECT/SEMDIF/SIGTEST - OBJECT                            *
C*                                                                       *
C*                                                                       *
C*                                                                       *
C*     THIS PROGRAM IS AN ADAPTATION OF THE MPAIR SUBROUTINE OF THE      *
C*     SYSTEM/360 SCIENTIFIC SUBROUTINE PACKAGE OF THE  WILCOXON TEST    *
C*     FOR USE WITH  SEMANTIC DIFFERENTIAL DATA.  THE PROGRAM TAKES      *
C*     DATA INPUT FROM PROGRAM SEMDIF/OSGOOD AND MAKES A SERIES OF       *
C*     COMPARISONS.  THIS PROGRAM WAS MODIFIED FOR USE WITH THE          *
C*     SEMANTIC DIFFERENTIAL PROGRAM BY BARBARA L. METIVIER,  EDWIN      *
C*     D. LAWSON, & MARCY METIVIER AT THE STATE UNIVERSITY COLLEGE       *
C*     OF NEW YORK AT FREDONIA, NEW YORK 14063.                          *
C*
 *     THIS PROGRAM TESTS THE SIGNIFICANCE OF THE DISTANCE (CONVERTED    *
C*     INTO RANKS) AS A TWO-TAILED TEST BETWEEN ONE CONCEPT AND TWO      *
C*     OTHER CONCEPTS.  THUS WHERE 101=CONCEPT GOOD AND 102=CONCEPT      *
C*     BAD WE ARE INTERESTED IN LEARNING WHETHER ANOTHER CONCEPT SUCH    *
C*     AS SELF=07 IS SIGNIFICANTLY CLOSER TO THE CONCEPT GOOD OR TO      *
C*     BAD.                                                              *
C*                                                                       *
C*     THE IDENT NUMBERS OF THE CONTROL CONCEPTS ARE THE LAST 6          *
C*     NUMBERS OF THE CONCEPT LIST (I.E. IF THERE ARE 106 CONCEPTS,      *
C*     THE CONTROL (OR REFERENCE) CONCEPTS ARE 101-106).  THE PROGRAM    *
C*     WILL HANDLE UP TO 109 CONCEPTS INCLUDING THE LAST SIX.  THE       *
C*     NAMES OF THE CONTROL CONCEPTS ARE CONTAINED IN THE FIRST 3        *
C*     RECORDS.  ADDITIONAL CONTROL OR REFERENCE CONCEPTS CAN BE LISTED*
C*     BY MOVING ADDITIONAL CONCEPTS TO THE LAST SIX POSITIONS           *
C*                                                                       *
C*     THE PROGRAM IS ALSO SET UP TO USE TWO INPUT FILES AND COM-        *
C*     PARE THE DISTANCE BETWEEN ONE CONCEPT AND THE CONTROL CON-        *
C*     CEPT IN THE FIRST FILE AND THE IDENTICAL CONCEPT,AND CON-         *
```

```
C*    TROL CONCEPT IN THE SECOND FILE.  INITIALLY IT PROCESSES          *
C*    THE FIRST FILE AS DESCRIBED ABOVE AND THEN DOES THE SAME          *
C*    WITH THE SECOND FILE.  THEN IT TAKES FROM EACH FILE THE           *
C*    DISTANCES BETWEEN EACH CONCEPT AND THE POSITIVE CONTROLS          *
C*    (I.E. GOOD,STRONG,ACTIVE) AND COMPARES LIKE CONCEPTS AS           *
C*    AS DESCRIBED ABOVE.  FINALLY IT DOES THE SAME FOR THE             *
C*    DISTANCES BETWEEN THE CONCEPTS AND THE NEGATIVE CONTROLS          *
C*    (I.E. BAD, WEAK, PASSIVE).                                        *
C*                                                                      *
C*    THIS AN INTERACTIVE PROGRAM WITH THE PROGRAM CONTROL INFOR-        *
C*    BEING ENTERED THROUGH THE TERMINAL IN ANSWER TO A SERIES          *
C*    OF QUESTIONS.  A REPORT OF THE RESULTS IS SENT TO A PRINTER       *
C*    FILE.                                                             *
C*                                                                      *
C*    THE NAME(S) OF THE INPUT FILE(S) ARE REQUESTED AT THE BE-         *
C*    GINNING OF THE RUN FROM THE TERMINAL.  FROM THIS INFORMATION      *
C*    THE NAMES OF THE OUTPUT FILES ARE CREATED AND WRITTEN BACK        *
C*    TO THE TERMINAL FOR YOUR INFORMATION.                             *
C*                                                                      *
C*     THE INPUT FILE HAS THE FOLLOWING FORMAT:                         *
C*                                                                      *
C*        RECORD      COL       CONTENTS                               *
C*          1          1-17     NAME OF FIRST CONTROL CONCEPT          *
C*                    18-34     NAME OF 2ND CONTROL CONCEPT            *
C*          2          1-17     NAME OF 3RD CONTROL CONCEPT            *
C*                    18-34     NAME OF 4TH CONTROL CONCEPT            *
C*          3          1-17     NAME OF 5TH CONTROL CONCEPT            *
C*                    18-34     NAME OF 6TH CONTROL CONCEPT            *
C*          4          1-3      IDENT NO. FOR 1ST CONTROL CONCEPT      *
C*                     4-6      IDENT NO. FOR 1ST NON-CONTROL CONCEPT  *
C*                     7-9      CASE NO. FOR 1ST NON-CONTROL CONCEPT   *
C*                    10-19     OSGOOD D BETWEEN ABOVE 2 CONCEPTS      *
C*                    20-22     BLANK                                  *
C*                    23-36     NAME OF 1ST NON-CONTROL CONCEPT        *
C*          5          1-3      IDENT NO. FOR 2ND CONTROL CONCEPT      *
C*                     4-36     SAME AS ABOVE                          *
C*          6          1-3      IDENT NO. FOR 3RD CONTROL CONCEPT      *
C*                     4-36     SAME AS ABOVE                          *
C*         7-9                  SAME AS ABOVE ONLY FOR 4TH,5TH,6TH     *
C*                              CONTROL CONCEPTS                       *
C*          10         1-3      IDENT NO. FOR 1ST CONTROL CONCEPT      *
C*                     4-6      IDENT NO. FOR 2ND NON-CONTROL CONCEPT  *
C*                     7-9      CASE NO. FOR 2ND NON-CONTROL CONCEPT   *
C*                    10-19     OSGOOD D BETWEEN ABOVE 2 CONCEPTS      *
C*                    20-22     BLANK                                  *
C*                    23-36     NAME OF 2ND NON-CONTROL CONCEPT        *
C*          11         1-3      IDENT NO. FOR 2ND CONTROL CONCEPT      *
C*                     4-36     SAME AS ABOVE (RECORD 7)               *
C*        12-15                 SAME AS ABOVE ONLY FOR 3RD,4TH,5TH,    *
C*                              6TH CONTROL CONCEPTS                   *
C*                                                                     *
C*                REPEAT THESE GROUPS OF 6 RECORDS - ONE GROUP FOR     *
C*                EACH NON-CONTROL CONCEPT (RECORDS FOR 1 CASE).       *
C*                                                                     *
```

```
C*              REPEAT ABOVE GROUP (6 X M1 RECORDS) - 1 FOR EACH      *
C*              CASE.                                                 *
C*                                                                    *
C*    THE OUTPUT FILE HAS THE FOLLOWING FORMAT:                       *
C*                                                                    *
C*       RECORD       COL       CONTENTS                             *
C*         1          1-3       CONCEPT IDENT NUMBER                 *
C*                     4         BLANK                                *
C*                    5-19       CONCEPT NAME                         *
C*                   20-26       P FOR GOOD VS BAD                    *
C*                   27-33       Z FOR GOOD VS BAD                    *
C*                   34-37       T FOR GOOD VS BAD                    *
C*                   38-40       + OR -                               *
C*                   41-61       P, Z, T, + OR - (AS ABOVE) FOR       *
C*                               STRONG VS WEAK                       *
C*                   62-82       P, Z, T, + OR - (AS ABOVE) FOR       *
C*                               ACTIVE VS PASSIVE                    *
C*                                                                    *
C*              REPEAT THIS RECORD - ONE FOR EACH NON-CONTROL         *
C*              CONCEPT.                                              *
C*                                                                    *
C*                                                                    *
C*     NOTE:                                                          *
C*       A NEGATIVE SIGN MEANS THAT THE COMPARED CONCEPT IS           *
C*       CLOSER TO THE FIRST OF THE TWO CONTROL CONCEPTS.             *
C*                                                                    *
C*       P = PROBABILITY OF BEING LESS THAN (TWO-TAILED TEST)         *
C*                                                                    *
C*       Z = STANDARD SCORE                                           *
C*                                                                    *
C*       T = SUM OF RANKS + OR - DIFFERENCES WHICHEVER IS             *
C*           SMALLER                                                  *
C*     CON1-CON6 ARE THE CONTROL OR REFERENCE CONCEPTS                *
C*                                                                    *
C*       M=TOTAL NUMBER OF CONCEPTS                                   *
C*       M1=TOTAL NUMBER OF NON-CONTROL CONCEPTS                      *
C*       N=TOTAL NUMBER OF CASES                                      *
C*           CARD ABOVE                                               *
C*                                                                    *
C*  THE FOLLOWING SUBROUTINE WERE REPRINTED BY PERMISSION FROM        *
C*  SYSTEM 1360 SCIENTIFIC SUBROUTINE PACKAGE (360A-CM-03X)           *
C*  VERSION III PROGRAMMER'S MANUAL, COPYIRIGHT 1966, 1967, 1968      *
C*  BY INTERNATIONAL BUSINESS MACHINES CORPORATION:                   *
C*           RANK   PAGE 71                                           *
C*           NDTR   PAGE 78                                           *
C*           MPAIR  PAGE 70                                           *
C*           SRANK  PAGE 73                                           *
C*           TIE    PAGE 74                                           *
C*           UTEST  PAGE 75                                           *
C*                                                                    *
C*                                                                    *
C*                                                                    *
C*                 UPDATE (1982)                                      *
C*                                                                    *
```

```
C*    THIS PROGRAM HAS BEEN MODIFIED TO NOW INCLUDE THREE TESTS
C*    FROM WHICH THE USER CAN CHOOSE, THE WILCOXON TEST, THE MANN-
C*    WHITNEY TEST AND THE SPEARMAN RANK CORRELATION FOR THE INDIVIDUAL
C*    AGAINST THE CONTROL CONCEPTS.  THESE TESTS CAN ALSO BE PERFORMED
C*    WITH TWO (2) INPUT FILES.
C*    IF THE USER HAS TWO (2) INPUT FILES AND CHOOSES EITHER THE
C*    WILCOXON OR MANN-WHITNEY TESTS, THE PROGRAM WILL ASK IF THE USER
C*    WOULD LIKE THE FILES TO BE ALSO RANKED.  THIS IS ASKED AFTER THE
C*    ORIGINAL TEST IS COMPLETED.
C*
C*    IF THE USER REQUESTS THE MANN-WHITNEY TEST, THE WILCOXON TEST IS
C*    FIRST DONE ON THE SEPARATE FILES.  IF THE SPEARMAN RANK IS
C*    REQUESTED, AFTER IT IS DONE THE USER HAS THE CHOICE OF ENDING
C*    THE PROGRAM OR PERFORMING EITHER THE WILCOXON OR THE MANN-WHITNEY
C*    TESTS.
C*
C*    THE WILCOXON (MATCHED PAIRS PROCEDURE) IS USED WITH ONE GROUP
C*    WHEN DIFFERENCES ARE TESTED BETWEEN A CONCEPT AND 2 REFERENCE
C*    POINTS SUCH AS GOOD AND BAD.  THE WILCOXON CAN BE USED WITH TWO
C*    GROUPS, FOR EXAMPLE A PRE- AND POST- TEST ALSO.  HERE TESTING
C*    WOULD BE TO SEE WHETHER A CONCEPT WAS CLOSER TO GOOD ON ONE
C*    TESTING OR THE OTHER. THE MANN-WHITNEY IS USED WITH INDEPENDENT
C*    GROUPS.
C*
C*    THE SPEARMAN CORRELATION DETERMINES THE DEGREE OF ASSOCIATION
C*    BETWEEN TWO SETS OF DEE MEASUREMENTS, AS GOOD VS. CONCEPT
C*    IN ONE GROUP VS. A SECOND.
C*
C*    MODIFICATIONS DONE BY BARBARA L. METIVIER AND MARCY L.          *
C*    METIVIER COMPLETED SEPT. 1982.                                  *
C*                                                                    *
C*                                                                    *
C*                                                                    *
C*                        UPDATE (1983)                               *
C*                                                                    *
C*    ANOTHER COLUMN HAS BEEN ADDED TO THE PRINTOUT IN ORDER TO       *
C*    SHOW P ON A SCALE FROM 0 TO 100. THIS SCALE IS A SOMEWHAT       *
C*    MODIFIED PERCENTILE SCORE.  THE PROBABILITY LEVELS ARE          *
C*    SHOWN IN THE TABLE UNDER "P" AND THE SCALED VALUES UNDER        *
C*    "SC".  THIS SCALE IS USED ONLY WITH THE WILCOXON MATCHED        *
C*    PAIRS TEST.                                                     *
C*                                                                    *
C*    WHEN THE PROGRAM IS RUN, A TABLE CONSISTING OF "NUMBER OF       *
C*    CASES" VS. "FACTORS" IS GIVEN TO THE USER.  IN ORDER TO MAKE    *
C*    THE CORRECT CONVERSION, THE USER MUST ENTER THE FACTOR          *
C*    CORRESPONDING TO THE NUMBER OF CASES IN THE SAMPLE.  THE        *
C*    USER CAN USE ANYWHERE FROM 15-50 CASES IN STEPS OF FIVE.        *
C*    (I.E. 15, 20, 25, ETC.)                                         *
C*                                                                    *
C*    KEY IS THE VARIABLE WHICH CONTROLS HOW MANY TIMES EACH TEST     *
C*    IS RUN. NP CONTROLS WHICH TEST IS RUN (1=WILCOXON,2=MANN-WHITNEY*
C*    3=SPEARMAN RANK).                                               *
C*                                                                    *
C*    MODIFICATIONS COMPLETED AUGUST, 1983  BY M.L. METIVIER AND      *
```

```
C*     V.THADEN                                                          *
C*                                                                       *
C*                                                                       *
C************************************************************************
C*
C*
C*
      INTEGER G,C,F,KEY
      DIMENSION A(5886),B(5886),D(50),E(50),L(50),S(12),S2(15),SN(3),
     *CONCNM(15), FILENM(4),FIDIN1(4), FIDIN2(4),FIDST1(5), FIDST2(5),
     *FOUT1(5), FOUT2(5), FOUT3(5), FOUT4(5), CON1(3), CON2(3), CON3(3),
     *CON4(3), CON5(3), CON6(3), DATE(2),SPE(6)
      DATA SN/"-","+","="/, KEY/0/, ADD1,  ADD2,ADD3,ADD4/"/SORT.",
     *"/OUT.", "/POS.", "/NEG."/,N1,N2,N3,N4/4*0/
C*
C*        INITIALIZE FILE NAME ARRAYS AND ENTER CONTROL INFORMATION.
C*
      CALL DATIT(DATE)
      KEY=0
      DO 10 I=1,4
         FIDIN1(I)=" "
         FIDIN2(I)=" "
         FIDST1(I)=" "
         FIDST2(I)=" "
         FOUT1(I)=" "
         FOUT2(I)=" "
         FOUT3(I)=" "
   10    FOUT4(I)=" "
         FIDST1(5)=" "
         FIDST2(5)=" "
         FOUT1(5)=" "
         FOUT2(5)=" "
         FOUT3(5)=" "
         FOUT4(5)=" "
C*
C*          ENTER CONTROL INFORMATION
C*
      WRITE(5,*/)"PLEASE ENTER TOTAL NUMBER OF CONCEPTS."
      READ(5,/)M
      WRITE(5,*/)"PLEASE ENTER NUMBER OF CASES IN EACH FILE. (THIS TEST
     *AS WRITTEN IN THIS PROGRAM CAN ONLY BE DONE WITH SAMPLES OF 15-50
     * CASES IN 5-STEP INTERVALS.)"
      READ(5,/)N
      CALL TABLE
      WRITE(5,*/)"PLEASE ENTER THE CONVERSION FACTOR (EX. FOR 35 CASES,
     *ENTER 5.15943)."
      READ(5,916)CONV
      WRITE(5,*/)"ENTER NUMBER OF INPUT FILES."
      READ (5,/)NF
      IF (NF.EQ.1) GO TO 20
      WRITE(5,*/)"ENTER 1 IF MATCHED PAIRS, 2 FOR INDEPENDENT GROUPS,"
      WRITE(5,*/)"OR 3 FOR SPEARMAN RANK CORRELATION."
      READ(5,/)NP
   20 WRITE(5,*/)"PLEASE ENTER NAME OF FIRST INPUT FILE - NO MORE THAN
```

```
     *20 CHARACTERS."
      WRITE(5,*/)">"
      READ(5,906) FIDIN1
      IF (NE.EQ.1)GO TO 30
      WRITE(5,*/)"PLEASE ENTER NAME OF SECOND INPUT FILE - NO MORE THAN
     *20 CHARACTERS. "
      WRITE(5,*/)">"
      READ(5,906) FIDIN2
C*
C*        CREATE NAMES FOR SORTED FILES
C*
   30 CALL NEWNAM(FIDIN1,FIDST1,ADD1,N1,N2)
      IF(FIDIN2(1).NE." ")CALL NEWNAM (FIDIN2,FIDST2,ADD1,N3,N4)
C*
C*        CREATE OUTPUT FILE NAMES.
C*
      CALL NEWNAM(FIDIN1,FOUT1,ADD2,N1,N2)
      IF(FIDIN2(1).EQ." ")GO TO 40
      CALL NEWNAM (FIDIN2,FOUT2,ADD2,N3,N4)
      CALL NEWNAM(FIDIN1,FOUT3,ADD3,N1,N2)
      CALL NEWNAM(FIDIN1,FOUT4,ADD4,N1,N2)
C*
C*        CHANGE FILE NAMES
C*
      CHANGE (4,TITLE=FIDIN2)
C*    CHANGE (7, TITLE=FOUT2)
C*    CHANGE (8, TITLE=FOUT3)
C*    CHANGE (9, TITLE=FOUT4
      CHANGE (12, TITLE=FIDST2)
   40 CHANGE (3, TITLE=FIDIN1)
C*    CHANGE (6, TITLE=FOUT1)
      CHANGE (11, TITLE=FIDST1)
      WRITE(5,915)FIDIN1,FIDIN2,FIDST1,FIDST2
      DO 50 I=1,4
   50     FILENM(I)=FIDIN1(I)
          READ(3,997)CON1,CON2
          READ(3,997)CON3,CON4
          READ(3,997)CON5,CON6
          REWIND 3
C*
C*        SORT INPUT FILE(S) BY NON-CONTROL CONCEPT ID AND THEN
C*        WITHIN THAT BY CONTROL CONCEPT ID.
      CALL FILSRT(FIDIN1,FIDST1)
      IF (FIDIN2(1).NE." ")CALL FILSRT(FIDIN2,FIDST2)
      M1=M-6
C*        IF THERE ARE TWO INPUT FILES, SET KEY=1
      IF(FIDIN2(1).NE." ")KEY=1
C*
C*        FIRST INPUT FILE - CORRELATION OF RATINGS AGAINST PAIRS
C*        OF CONTROL CONCEPTS.
C*
C* 70 NUD=6
   70 REWIND 11
      REWIND 12
```

```
      NI1=11
      NI2=11
C*
      IF(NP.EQ.3) GO TO 300
   75 WRITE(10,908)DATE
      WRITE(10,*/)"WILCOXON MATCHED PAIRS TEST"
      WRITE(10,940)FILENM
C*    WRITE(NUD,906) FILENM
   80 WRITE(10,970)CON1,CON2,CON3,CON4,CON5,CON6
   89 IF (NP.NE.3 .AND. KEY.LE.2) WRITE(10,911)
      IF (NP.NE.3 .AND. KEY.GT.2) WRITE(10,910)
      DO 105 F=1,M1
         G=(-3)
         H=(-4)
         IX=0
C*
C*          READ IN CONCEPT NAME
C*
         READ(NI1,920)(CONCNM(JJ),JJ=1,15)
         BACKSPACE NI1
         DO 100 J=1,5,2
C*
C*          READ IN OSGOOD D VALUES
C*
          READ(NI1,990)(A(I),I=1,N)
          READ(NI2,990)(B(I),I=1,N)
C*
C*           FOR DEBUGGING PURPOSES ONLY, PRINT ARRAYS A AND B FOR THE
C*           FIRST CORRELATION FOR EACH CONCEPT.
C*
C*     IF(J.EQ.1.AND.F.LE.2)WRITE(15,995)(A(I),I=1,20),(B(I),I=1,20),
C*    *(CONCNM(I),I=1,15)
          IF (KEY.LT.3.OR.NP.EQ.1) CALL MP (N,A,B,K,T,Z,P,D,E,L,IE,KA)
          IF (KEY.GE.3.AND.NP.EQ.2) CALL UTEST  (A,B,N,KA,T,Z,P)
          IF (NP.NE.3) GO TO 90
C*
C*       PROCESS SPEARMAN RANK CORRELATION
C*
          CALL CORR(A,B,N,RS,F)
          IX=IX+1
          SPE(IX)=RS
          GO TO 95
C*
   90     IF (Z.GT.0) P=1.0 - P
          P= P*2.0
          BB=SN(KA)
C*
C*    IF THE WILCOXON TESTS ARE BEING DONE, GO TO 92 TO FIGURE THE SCALE
C*
          IF (KEY.LE.2) GO TO 92
          G=G+4
          S(G)=P
          S(G+1)=ABS(Z)
          S(G+2)=T
```

```
            S(G+3)=BB
            GO TO 95
C*          *
C*    COMPUTE SCALE, GO TO 93 IF CLOSER TO POSITIVE CONCEPT
C*    IF CLOSER TO NEGATIVE CONCEPT (BB="+" OR "="), FIND VALUE AND
C*    GO INPUT INTO PRINT FILE (S2).
C*
   92       H=H+5
            Q=50*ABS(Z)/CONV
            IF(BB.EQ."-")GO TO 93
            Q=50-Q
            GO TO 94
C*
C*    FIGURE VALUE IF CLOSER TO POSITIVE CONCEPT
C*
   93       Q=50+Q
C*
C*    ROUND VALUE AND TRUNCATE TO INTEGER. PUT EVERYTHING TO PRINTFILE
C*
   94       V=INT(Q+.5)
            S2(H)=P
            S2(H+1)=ABS(Z)
            S2(H+2)=T
            S2(H+3)=BB
            S2(H+4)=V
C*
   95       IF (KEY.LT.3)GO TO 100
            DO 97 I=1,N
               READ (NI1,990,END=100)
   97          READ(NI2,990)
  100       CONTINUE
C*
C*    IF THE WILCOXON TEST IS BEING PRINTED, THE FIRST WRITE IS USED.
C*    MANN-WHITNEY USES THE SECOND.
C*
         IF (NP.NE.3 .AND. KEY.LE.2) WRITE(10,981)F,(CONCNM(JJ),JJ=1,15),
     *(S2(LL),LL=1,15)
         IF (NP.NE.3 .AND. KEY.GT.2) WRITE(10,980)F,(CONCNM(JJ),JJ=1,15),
     *(S(LL),LL=1,12)
         IF (NP.EQ.3)WRITE(10,925)F,(CONCNM(JJ),JJ=1,15),(SPE(LL),LL=1,3)
C*        WRITE(NUD,904)F,(CONCNM(JJ),JJ=1,15),(S(LL),LL=1,12)
  105    CONTINUE
         IF (NP.NE.3)WRITE (10,950)
         IF (KEY.EQ.0)GO TO 120
         IF (NP.NE.3) GO TO 108
         IF(NP.EQ.3.AND.KEY.EQ.3) GO TO 400
C*
C*          IF TEST WAS CORRELATION,  RETURN FOR WILCOXON OR MANN-WHITNEY
C*
         IF (ENDFLG.EQ.1) GO TO 120
         ENDFLG = 1
         WRITE(5,*/)"DO YOU NOW WANT EITHER THE TEST FOR MATCHED PAIRS OR I
        *NDEPENDENT GROUPS?"
         READ(5,985)ANS
```

```
      IF (ANS.EQ."NO")GO TO 120
      WRITE(5,*/)"ENTER 1 FOR MATCHED PAIRS OR 2 FOR INDEPENDENT GROUPS"
      READ(5,/)NP
      WRITE(10,907)
      KEY = 1
      GO TO 70
  108 GO TO (200,300,400,110)KEY
  110 CONTINUE
      IF (ENDFLG.EQ.1) GO TO 120
         ENDFLG=1
      WRITE(5,*/)"DO YOU NOW WANT A SPEARMAN RANK CORRELATION FOR THE 2
     * INPUT FILES?"
      READ(5,985)ANS
      IF (ANS.EQ."NO") GO TO 120
         NP = 3
         KEY = 1
         GO TO 300
C* 120 CLOSE(6,DISP=CRUNCH)
C*    CLOSE(7,DISP=CRUNCH)
C*    CLOSE(8,DISP=CRUNCH)
C*    CLOSE(9,DISP=CRUNCH)
  120 CLOSE(11,DISP=DELETE)
      CLOSE(12,DISP=DELETE)
      CLOSE(14,DISP=CRUNCH)
      CLOSE(2,DISP=CRUNCH)
      STOP
C*
C*       2 INPUT FILES - FIRST AGAINST SECOND COMPARISON
C*       PROCESS SECOND FILE - EACH CONCEPT AGAINST CONTROL PAIRS
C*
  200 KEY=KEY+1
      DO 210 I=1,4
  210    FILENM(I)=FIDIN2(I)
C*    NUD=7
      NI1=12
      NI2=12
      WRITE(10,907)
C*
C*    IF MATCHED PAIRS TEST, SKIP TO 92 TO COMPUTE SCALE
C*
      GO TO 75
C*
C*       PROCESS FIRST AND SECOND FILES - EACH CONCEPT AGAINST POSITIVE
C*          CONTROL CONCEPTS - GOOD,STRONG,ACTIVE
C*
  300 REWIND 11
      REWIND 12
      NI1=11
      NI2=12
      IF(NP.EQ.3) KEY=KEY+1
      KEY=KEY+1
C*    NUD=8
      WRITE(10,907)
      WRITE(10,908)DATE
```

```
      IF (NP.EQ.1) WRITE(10,*/)"WILCOXON MATCHED PAIRS TEST"
      IF (NP.EQ.2) WRITE(10,*/)"MANN-WHITNEY INDEPENDENT GROUPS TEST"
      IF(NP.EQ.3) WRITE(10,*/)"SPEARMAN RANK CORRELATION COEFFICIENT"
      WRITE(10,900) FIDIN1,FIDIN2
C*    WRITE(NUD,906) FIDIN1,FIDIN2
      WRITE(10,901)CON1,CON3,CON5
      GO TO 89
C*
C*        PROCESS FIRST AND SECOND FILES - EACH CONCEPT AGAINST NEGATIVE
C*           CONTROL CONCEPTS - BAD,WEAK,PASSIVE
C*
  400 KEY=KEY+1
      REWIND 11
      REWIND 12
C*     NUD=9
      WRITE(10,907)
      WRITE(10,908)DATE
      DO 410 I=1,N
         READ(NI1,990)
  410    READ(NI2,990)
      IF (NP.EQ.1) WRITE(10,*/)"WILCOXON MATCHED PAIRS TEST"
      IF (NP.EQ.2) WRITE(10,*/)"MANN-WHITNEY INDEPENDENT GROUPS TEST"
      IF (NP.EQ.3) WRITE(10,*/)"SPEARMAN RANK CORRELATION COEFFICIENT"
      WRITE(10,902)FIDIN1,FIDIN2
C*    WRITE(NUD ,906)FIDIN1,FIDIN2
      WRITE(10,903)CON2,CON4,CON6
      GO TO 89
  900 FORMAT(1X,"ANALYSIS OF RESPONSES - POSITIVE - ",4A6," VS ",4A6///)
  901 FORMAT(36X,2A6,A3,19X,2A6,A3,19X,2A6,A3//)
  902 FORMAT(1X,"ANALYSIS OF RESPONSES - NEGATIVE - ",4A6," VS ",4A6///)
  903 FORMAT(36X,2A6,A3,19X,2A6,A3,19X,2A6,A3//)
  904 FORMAT(I3,1X,15A1,3(F7.5,F7.5,F4.0,1X, A1,1X ))
  905 FORMAT(/)
  906 FORMAT(4A6,"   VS   ", 4A6)
  907 FORMAT("1")
  908 FORMAT(1X,2A6)
  910 FORMAT(5X,"CONCEPT",10X,3(12X, "P",8X,"Z",5X,"T",2X,"SIGN")//)
  911 FORMAT(5X,"CONCEPT",10X,3(7X,"P",8X,"Z",6X,"T",2X,"SIGN",2X,"SC")
     */)
  915 FORMAT(1X,"YOUR FILES ARE: ",2(4A6,1X)/5X,2(5A6,1X)/)
  916 FORMAT(F7.5)
  920 FORMAT(22X,15A1)
  925 FORMAT(1X,I3,". ",15A1,13X,F8.3,26X,F8.3,26X,F8.3)
  930 FORMAT(1X,"A(",I1,")=",F10.5,3X,"B(",I1,")=",F10.5)
  940 FORMAT(1X,"ANALYSIS OF RESPONSES USING FILE, ",4A6///)
  950 FORMAT(///" NOTE A NEGATIVE SIGN MEANS THAT THE COMPARED CONCEPT
     -IS CLOSER TO THE FIRST OF THE TWO COMPARISON CONCEPTS."//
     -" P = PROBABILITY OF BEING LESS THAN (TWO-TAILED TEST)"//
     -" Z = STANDARD SCORE"//" T = SUM OF RANKS + OR - DIFFERENCES
     -WHICHEVER IS SMALLER")
  960 FORMAT(2(I3,3X),7A1)
  970 FORMAT(28X,2A6,A3,"VS",2A6,A3,2X,2A6,A3,"VS",2A6,A3,2X,2A6,
     1A3,"VS",2A6,A3//)
  980 FORMAT(1X,I3,". ",15A1, 9X, 3(F7.5,2X, F7.5,1X,F5.0,2X, A1, 9X ))
```

```
  981 FORMAT(1X,I3,". ",15A1,1X,3(4X,F7.5,2X,F7.5,1X,F5.0,2X,A1,2X,I3))
  985 FORMAT(A2)
  990 FORMAT(9X,F10.5)
  995 FORMAT(4(1X,10F10.5,5X/),1X,15A1)
  997 FORMAT(2A6,A3,1X,2A6,A3)
      END
      SUBROUTINE NEWNAM(FILIN,FILOUT,FILADD,JIN,KIN)
C*
C*        CREATE NEW FILE NAME BY APPENDING FILADD.
C*
      DIMENSION FILIN(1),FILOUT(1)
C*
C*        FIND LAST NON-BLANK WORD OF FILIN AND LOAD ALL NON-BLANK WORDS
C*        INTO FILOUT.
C*
      IF(JIN.EQ.0)GO TO 90
      J=JIN
      K=KIN
      DO 80 I=1,J
   80    FILOUT(I)=FILIN(I)
      GO TO 320
   90 DO 100 I1 =1,4,1
         I=5-I1
  100    IF(FILIN(I).NE." ")GO TO 200
  200 DO 250 K=1,I
  250    FILOUT(K)=FILIN(K)
C*     WRITE(5,900)(FILIN(II),II=1,4),(FILOUT(II),II=1,5)
C*
C*        APPEND FILADD TO FILOUT STARTING AT FIRST BLANK CHARACTER.
C*
      DO 300 J1 =I1,4,1
         J=5-J1
         DO 300 K=7,47,8
            IF((CONCAT(" ", FILOUT(J),47,K,8)).NE." ")GO TO 310
  300       CONTINUE
  310 K=K-8
      IF(K.GE.7)GO TO 315
      K=47
      J=J+1
  315 JIN=J
      KIN=K
      FILIN(J)=CONCAT(FILIN(J),".",K,47,8)
  320 K2=47
      K1=K+1
  330 FILOUT(J)=CONCAT(FILOUT(J),FILADD,K,K2,K1)
C*     WRITE(5,900)(FILIN(II),II=1,4),(FILOUT(II),II=1,5),FILADD
      K2=K2-K1
      IF(K2.LT.0)RETURN
      K=47
      K1=48-K1
      J=J+1
      GO TO 330
  900 FORMAT(1X,"FILEIN IS ",4A6,"  FILEOUT IS ",5A6/1X,
     *"FILADD IS", A6)
```

```
      END
      SUBROUTINE MP    (N,A,B,K,T,Z,P,D,E,L,IE,KA)
      DIMENSION A(1),B(1),D(1),E(1),L(1)
      IE=0
      K=N
C*          FIND DIFFERENCES OF MATCHED PAIRS
      BIG=0.0
      DO 55 I=1,N
      DIF=A(I)-B(I)
      IF (DIF) 10, 20, 30
C*        DIFFERENCE HAS A NEGATIVE SIGN (-)
   10 L(I)=1
       GO TO 40
C*        DIFFERENCE IS ZERO (0)
   20 L(I)=2
      K=K-1
      GO TO 40
C*        DIFFERENCE HAS A POSITIVE SIGN(+)
   30  L(I)=3
   40  DIF=ABS(DIF)
      IF(BIG-DIF) 45, 50, 50
   45 BIG=DIF
   50 D(I)=DIF
   55 CONTINUE
      IF(K) 57,57,59
   57 IE=1
      T=0.0
      Z=-4.31E68
      P=0
      GO TO 100
C*    STORE A LARGE VALUE IN PLACE OF 0 DIFFERENCE IN ORDER TO
C*    ASSIGN A LARGE RANK (LARGER THAN K), SO THAT ABSOLUTE VALUES
C*    OF SIGNED DIFFERENCES WILL BE PROPERLY RANKED
   59 BIG=BIG*2.0
      DO 65 I=1,N
      IF(L(I)-2) 65, 60, 65
   60 D(I)=BIG
   65 CONTINUE
      CALL RANK (D,E,N)
C*        FIND SUMS OF (+) DIFFERENCES AND (-) DIFFERENCES
      SUMP=0.0
      SUMM=0.0
      DO 80 I=1,N
      IF(L(I)-2) 70, 80, 75
   70 SUMM=SUMM+E(I)
      GO TO 80
   75 SUMP=SUMP+E(I)
   80 CONTINUE
C*        SET 1 = SMALLER SUM
      IF(SUMP-SUMM)  85, 85, 90
   85 T=SUMP
      KA=1
      GO TO 95
   90 T=SUMM
```

```
      KA=2
C*       COMPUTE MEAN, STANDARD DEVIATION, AND Z
   95 FK=K
C*    WRITE(15,97)T,FK,SUMM,SUMP,(L(I),I=1,N),(D(I),I=1,N),(E(I),I=1,N)
   97 FORMAT(1X,"T=",F4.1,3X,"FK=",F4.0,3X,"SUMM=",F6.2,3X,"SUMP=",
     *F6.2,3X,/1X, "L(I)=",15(I1,2X)/1X, "D(I)=",15(F8.3),3X/1X,
     *"E(I)= ", 15(F7.2))
      U=FK*(FK+1.0)/4.0
      S=SQRT((FK*(FK+1.0)*(2.0*FK+1.0))/24.0)
      Z=ABS((T-U)/S)
C*       COMPUTE THE PROBABILITY OF A VALUE AS EXTREME AS Z
C*     WRITE(2,*/)U,S,Z
   98 FORMAT(1X,"U=",F10.5,3X,"S=",F10.5,3X,"Z=",F10.5)
      CALL NDTR (Z,P,BIG)
  100 RETURN
      END
      SUBROUTINE NDTR(X,P,D)
C*
C*
C*    SUBROUTINE NDTR
C*
C*    PURPOSE
C*       COMPUTES Y=P(X)=PROBABILITY THAT THE RANDOM VARIABLE U,
C*       DISTRIBUTED NORMALLY (0,1), IS LESS THAN OR EQUAL TO X.
C*       F(X), THE ORDINATE OF THE NORMAL DENSITY AT X, IS ALSO
C*       COMPUTED.
C*
C*      USAGE
C*       CALL NDTR(X,P,D)
C*
C*    DESCRIPTION OF PARAMETERS
C*       X--INPUT SCALAR FOR WHICH P(X) IS COMPUTED
C*       P--OUTPUT PROBABILITY.
C*       D--OUTPUT DENSITY
C*
C*    REMARKS
C*       MAXIMUM ERROR IS 0.0000007.
C*
C*    METHOD
C*       BASED ON APPROXIMATIONS IN C. HASTINGS, APPROXIMATIONS FOR
C*       DIGITAL COMPUTERS, PRINCETON U. PRESS, 1955.   SEE EQUATION
C*       26.2.17, HANDBOOK OF MATHEMATICAL FUNCTIONS, ABROMOWITZ &
C*       STEGUN, DOVER.
C*
C*
C*
C*
      AX=ABS(X)
      T=1.0/(1.0+.2316419*AX)
      D=0.3989423*EXP(-X*X/2.0)
      P=1.0-D*T*((((1.330274*T-1.821256)*T+1.781478)*T-
     1  0.3565638)*T+0.3193815)
C*     WRITE(2,*/)X,AX,T,D,P
      IF(X)1,2,2
```

```
    1 P=1.0-P
    2 RETURN
      END
C*       RANK A VECTOR OF VALUES
C*    USAGE
C*       CALL RANK(A,R,N)
C*    DESCRIPTION OF PARAMETERS
C*       A - INPUT VECTOR OF N VALUES
C*       R - OUTPUT VECTOR OF LENGTH N. SMALLEST VALUE IS RANKED 1,
C*           LARGEST IS RANKED N. TIES ARE ASSIGNED AVERAGE OF TIED
C*           RANKS
C*       N - NUMBER OF VALUES
C*    REMARKS
C*       NONE
C*    SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C*      /NONE
C*    METHOD
C*      VECTOR IS SEARCHED FOR SUCCESSIVELY LARGER ELEMENTS. IF TIES
C*      OCCUR, THEY ARE LOCATED AND THEIR RANK VALUE COMPUTED.
C*       FOR EXAMPLE, IF 2 VALUES ARE TIED FOR SIXTH RANK, THEY ARE
C*       ASSIGNED A RANK OF 6.5 (=(6+7)/2)
      SUBROUTINE RANK(A,R,N)
         DIMENSION A(1),R(1000)
C*
C*       INITIALIZATION
      DO 10 I=1,N
   10 R(I)=0.0
C*
C*       FIND RANK OF DATA
C*
      DO 100 I=1,N
C*
C*       TEST WHETHER DATA POINT IS ALREADY RANKED
C*
      IF(R(I)) 20, 20, 100
C*
C*       DATA POINT TO BE RANKED
C*
   20 SMALL=0.0
      EQUAL=0.0
      X=A(I)
      DO 50 J=1,N
      IF(A(J)-X) 30, 40, 50
C*       COUNT NUMBER OF DATA POINTS WHICH ARE SMALLER
C*
C*
   30 SMALL=SMALL+1.0
      GO TO 50
C*
C*       COUNT NUMBER OF DATA POINTS WHICH ARE EQUAL
C*
   40 EQUAL=EQUAL+1.0
         R(J)=-1.0
   50 CONTINUE
```

```
C*
C*       TEST FOR TIE
C*
      IF(EQUAL-1.0) 60, 60, 70
C*
C*       STORE RANK OF DATA POINT WHERE NO TIE
C*
   60 R(I)=SMALL+1.0
      GO TO 100
C*
C*       CALCULATE RANK OF TIED DATA POINTS
C*
   70 P=SMALL + (EQUAL + 1.0)*0.5
      DO 90 J=I,N
      IF(R(J)+1.0) 90, 80, 90
   80 R(J)=P
   90 CONTINUE
  100 CONTINUE
      RETURN
      END
      SUBROUTINE UTEST(A,B,N,SIGN,T,Z,P)
      DIMENSION A(1),B(1),C(1000),R(1000)
C*
C*************************************************************************
C*       RANK SCORES FROM BOTH GROUP TOGETHER IN ASCENDING ORDER AND
C*       ASSIGN TIED OBVERVATIONS AVERAGE OF TIED RANKS
C*
C*       DESCRIPTION OF PARAMETERS:
C*         A  - INPUT VECTOR FROM FIRST FILE
C*         B  - INPUT VECTOR FROM SECOND FILE
C*         C  - A CONCATINATED WITH B.  USED FOR RANKING
C*         N  - LENGTH OF VECTORS A AND B
C*         T  - SUM OF SMALLER OF RANKS OF A AND B (OUTPUT)
C*         P  - PROBABILITY FROM SUBROUTINE NDTR
C*         R  - OUTPUT VECTOR OF RANKS.  SMALLEST VALUE IS RANKED 1,
C*              LARGEST IS RANKED N.  TIES ARE ASSIGNED AVERAGE OF TIED
C*              RANKS.
C*          SIGN  - STATISTIC USED TO TEST HOMOGENEITY OF THE TWO GROUPS
C*              (OUTPUT).
C*          Z  - MEASURE OF SIGNIGICANCE OF U IN TERMS OF NORMAL DIS-
C*              TRIBUTION (OUTPUT).
C*
C*       REMARKS:
C*          Z IS SET TO ZERO IF N2 IS LESS THAN 20.
C*
C*       SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED:
C*          RANK
C*          TIE
C*
C*
C*       METHOD:
C*          DESCRIBED IN S. SIEGEL, 'NONPARAMETRIC STATISTICS FOR
C*          FOR THE BEHAVIORAL SCIENCES', MCGRAW-HILL, NEW YORK, 1956,
C*          CHAPTER 6.
```

```
C*
C*************************************************************************
C*
C*     CONCATINATE VECTORS A AND B INTO C
C*
       DO 10 I=1,N
   10    C(I)=A(I)
        ,N2=N*2
       DO 20 I=N+1,N2
         X=I-N
         C(I)=B(X)
   20  CONTINUE
C*
C*       RANK C
C*
       CALL RANK(C,R,N2)
C*          WRITE(2,*/)(C(I),R(I),I=1,N2)
       Z=0
C*
C*       SUM RANKS IN 'A' GROUP
C*
       RA=0
       DO 30 I=1,N
   30     RA=RA+R(I)
C*
C*         SUM RANKS IN 'B' GROUP
C*
       RB=0
       DO 40 I=N+1,N2
   40    RB=RB+R(I)
C*
C*         R2 EQUALS TOTAL SUMMATION OF THE RANK
C*
       R2=RA+RB
C*      WRITE(2,*/)RA,RB,R2
C*
C*       CALCULATE U
C*
       FNX=N**2
       FN=N2
       FN2=N
       UP=FNX+FN2*((FN2+1)/2)-RB
       U=FNX-UP
       IF(UP-U)50, 60, 70
   50  U=UP
C*
C*         TEST FOR N2 LESS THAN 20
C*
   60  IF(N2-20)110, 70, 70
C*
C*         COMPUTE STANDARD DEVIATION
C*
   70  KT=1
       CALL TIE (R,N2,KT,TS)
```

```
      IF (TS) 80, 90, 80
   80 S=SQRT((FNX/(FN*(FN-1)))*(((FN*FN*FN-FN)/12.0)-TS))
      GO TO 100
   90 S=SQRT(FNX*(FN+1)/12.0)
C*
C*       COMPUTE Z
C*
  100 Z=(U-FNX*0.5)/S
  110 CALL NDTR(Z,P,BIG)
C*
C*          COMPARE RANKS OF A AND B
C*
      IF (RA-RB)130,140,150
C*
C*          IF RB IS LARGER THAN RA
C*
  130  SIGN=1
      T=RA
      GO TO 160
C*
C*          IF THE SUMMATIONS ARE EQUAL
  140  SIGN=3
       T=RA
       GO TO 160
C*
C*          IF RA IS LARGER THAN RB
C*
  150  SIGN=2
       T=RB
  160 RETURN
      END
      SUBROUTINE TIE(R,N,KT,T)
C*
C*************************************************************************
C*    SUBROUTINE TIE
C*
C*    PURPOSE
C*       CALCULATE CORRECTION FACTOR DUE TO TIES
C*
C*    USAGE
C*       CALL TIE(R,N,KT,T)
C*
C*    DESCRIPTION OF PARAMETERS
C*       R  - INPUT VECTOR OF RANKS OF LENGTH N CONTAINING VALUES
C*            1 TO N
C*       N  - NUMBER OF RANKED VALUES
C*       KT  - INPUT CODE FOR CALCULATION OF CORRECTION FACTOR
C*                  1   SOLVE EQUATION 1
C*                  2   SOLVE EQUATION 2
C*       T  - CORRECTION FACTOR (OUTPUT)
C*                EQUATION 1   T=SUM(CT**3-CT)/12
C*                EQUATION 2   T=SUM(CT*(CT-1)/2)
C*              WHERE CT IS THE NUMBER OF OBSERVATIONS TIED FOR A
C*                    GIVEN RANK
```

```
C*     REMARKS
C*        NONE.
C*
C*     SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C*        NONE
C*
C*     METHOD
C*        VECTOR IS SEARCHED FOR SUCCESSIVELY LARGER RANKS. TIES ARE
C*        COUNTED AND CORRECTION FACTOR 1 OR 2 SUMMED.
C*
C*************************************************************************
C*
      DIMENSION R(1)
C*
C*        INITIALIZATION
C*
      T=0.0
      Y=0.0
    5 X=1.0E38
      IND=0
C*
C*        FIND NEXT LARGEST RANK
C*
      DO 30 I=1,N
      IF(R(I)-Y) 30,30,10
   10 IF(R(I)-X) 20,30,30
   20 X=R(I)
      IND=IND+1
   30 CONTINUE
C*
C*        IF ALL RANKS HAVE BEEN TESTED, RETURN
C*
      IF(IND) 90,90,40
   40 Y=X
      CT=0.0
C*
C*        COUNT TIES
C*
      DO 60 I=1,N
      IF(R(I)-X) 60,50,60
   50 CT=CT+1.0
   60 CONTINUE
C*
C*        CALCULATE CORRECTION FACTOR
C*
      IF(CT) 70,5,70
   70 IF(KT-1) 75,80,75
   75 T=T+CT*(CT-1.)/2.0
      GO TO 5
   80 T=T+(CT*CT*CT-CT)/12.0
      GO TO 5
   90 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE DATIT(DATE)
      REAL DATE(2)
      DATE(1)='  /  /'
      DATE(2)='
      B=TIME(15)
      DATE(1)=CONCAT(DATE(1),B,47,47,16)
      DATE(1)=CONCAT(DATE(1),B,23,31,16)
      DATE(2)=CONCAT(DATE(2),B,47,15,16)
      RETURN
      END
      SUBROUTINE CORR(A,B,N,RS,F)
C*************************************************************************
C*                                                                       *
C*                                                                       *
C*     SUBROUTINE CORR                                                   *
C*                                                                       *
C*                                                                       *
C*     PURPOSE - TO COMPUTE THE SPEARMAN RANK CORRELATION COEFFICIENT    *
C*               FROM THE OSGOOD DEES CALCULATED BY THE PROGRAM          *
C*               STAT/OSGOOD.E PROGRAM, SEMDIF.                          *
C*     USAGE -  CALL CORR(A,B,M,RS)                                      *
C*     DESCRIPTION OF PARAMETERS                                         *
C*          A   - INPUT VECTOR OF M OBSERVATIONS FOR FIRST VARIABLE      *
C*          B   - INPUT VECTOR OF M OBSERVATIONS FOR SECOND VARIABLE     *
C*          N   - NUMBER OF CASES                                        *
C*          RS  - SPEARMAN RANK CORRELATION COEFFICIENT                  *
C*     THE SUBROUTINE USES THE IBM SUBROUTINES RANK, TIE, AND SRANK.     *
C*                                                                       *
C*                                                                       *
C*************************************************************************
C*
C*
      DIMENSION A(1),B(1),R(1000)
C*
      NR=0
C*
C*          CALL SRANK AND PRINT OUTPUT
C*
      CALL SRANK (A,B,R,N,RS,T,NDF,NR,F)
      RETURN
      END
      SUBROUTINE SRANK(A,B,R,N,RS,T,NDF,NR,F)
C*************************************************************************
C*
C*    SUBROUTINE SRANK
C*
C*    PURPOSE
C*       TEST CORRELATION BETWEEN TWO VARIABLES BY MEANS OF SPEARMAN
C*       RANK CORRELATION COEFFICIENT
C*
C*    USAGE
C*       CALL SRANK(A,B,R,N,RS,T,NDF,NR)
C*
C*    DESCRIPTION OF PARAMETERS
```

```
C*       A    - INPUT VECTOR OF N OBSERVATIONS FOR FIRST VARIABLE
C*       B    - INPUT VECTOR OF N OBSERVATIONS FOR SECOND VARIABLE
C*       R    - OUTPUT VECTOR FOR RANKED DATA, LENGTH IS 2*N. SMALLEST
C*              OBSERVATION IS RANKED 1, LARGEST IS RANKED N. TIES
C*              ARE ASSIGNED AVERAGE OF TIED RANKS.
C*       N    - NUMBER OF OBSERVATIONS
C*       RS   - SPEARMAN RANK CORRELATION COEFFICIENT (OUTPUT)
C*       T    - TEST OF SIGNIFICANCE OF RS (OUTPUT)
C*       NDF  - NUMBER OF DEGREES OF FREEDOM (OUTPUT)
C*       NR   - CODE, 0 FOR UNRANKED DATA IN A AND B, 1 FOR RANKED
C*              DATA IN A AND B (INPUT)
C*
C*    REMARKS
C*       T IS SET TO ZERO IF N IS LESS THAN TEN
C*
C*    SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C*       RANK
C*       TIE
C*
C*    METHOD
C*       DESCRIBED IN S. SIEGEL, "NONPARAMETRIC STATISTICS FOR THE
C*       BEHAVIORAL SCIENCES", MCGRAW-HILL, NEW YORK, 1956,
C*       CHAPTER 9
C*************************************************************
C*
      DIMENSION A(1),B(1),R(1)
C*
      FN=FLOAT(N)
      FNNN=(FN**3)-FN
C*
C*       DETERMINE WHETHER DATA IS RANKED
C*
      IF(NR-1) 5, 10, 5
C*
C*       RANK DATA IN A AND B VECTORS AND ASSIGN TIED OBSERVATIONS
C*       AVERAGE OF TIED RANKS
C*
      NS=0
    5 CALL RRANK (A,R,N,RS,T,NDF,NS)
      NS=N
      CALL RRANK (B,R,N,RS,T,NDF,NS)
      GO TO 40
C*
C*       MOVE RANKED DATA TO R VECTOR
C*
   10 DO 20 I=1,N
   20 R(I)=A(I)
      DO 30 I=1,N
      J=I+N
   30 R(J)=B(I)
C*
C*       COMPUTE SUM OF SQUARES OF RANK DIFFERENCES
C*
   40 D=0.0
```

```
      DO 50 I=1,N
      J=I+N
   50 D=D+(R(I)-R(J))*(R(I)-R(J))
C*
C*       COMPUTE TIED SCORE INDEX
C*
      KT=1
      NS=0
      CALL TTIE (R,N,KT,TSA,NS)
      NS=N
      CALL TTIE (R,N,KT,TSB,NS)
C*
C*       COMPUTE SPEARMAN RANK CORRELATION COEFFICIENT
C*
      IF(TSA) 60,55,60
   55 IF(TSB) 60,57,60
   57 RS=1.0-6.0*D/FNNN
      GO TO 70
   60 X=FNNN/12.0-TSA
      Y=X+TSA-TSB
      RS=(X+Y-D)/(2.0*(SQRT(X*Y)))
C*
C*       COMPUTE T AND DEGREES OF FREEDOM IF N IS 10 OR LARGER
C*
      T=0.0
   70 IF(N-10) 80,75,75
   75 CONTINUE
      T=RS*SQRT(FLOAT(N-2)/(1.0-RS*RS))
   80 NDF=N-2
C*     IF (F.EQ.1)CALL DEBUG(A,B,R,N,RS,T,FNNN,X,Y,D,TSA,TSB)
      RETURN
      END
      SUBROUTINE DEBUG(A,B,R,N,RS,T,FNNN,X,Y,D,TSA,TSB)
      DIMENSION A(1),B(1),R(1)
      DO 100 I = 1,N
  100    WRITE (14,900 ) A(I),B(I),R(I),R(I+N)
      WRITE (14,*//)N,RS,T,FNNN,X,Y,D,TSA,TSB
      RETURN
  900 FORMAT(4F12.8)
      END
      SUBROUTINE TTIE(R,N,KT,T,NS)
C*
C*****************************************************************
C*    SUBROUTINE TIE
C*
C*    PURPOSE
C*       CALCULATE CORRECTION FACTOR DUE TO TIES
C*
C*    USAGE
C*       CALL TIE(R,N,KT,T)
C*
C*    DESCRIPTION OF PARAMETERS
C*       R  - INPUT VECTOR OF RANKS OF LENGTH N CONTAINING VALUES
C*            1 TO N
```

```
C*       N  - NUMBER OF RANKED VALUES
C*       KT  - INPUT CODE FOR CALCULATION OF CORRECTION FACTOR
C*                  1   SOLVE EQUATION 1
C*                  2   SOLVE EQUATION 2
C*       T  - CORRECTION FACTOR (OUTPUT)
C*                EQUATION 1   T=SUM(CT**3-CT)/12
C*                EQUATION 2   T=SUM(CT*(CT-1)/2)
C*              WHERE CT IS THE NUMBER OF OBSERVATIONS TIED FOR A
C*                    GIVEN RANK
C*    REMARKS
C*       NONE
C*
C*    SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C*       NONE
C*
C*    METHOD
C*       VECTOR IS SEARCHED FOR SUCCESSIVELY LARGER RANKS. TIES ARE
C*       COUNTED AND CORRECTION FACTOR 1 OR 2 SUMMED.
C*
C*************************************************************************
C*
      DIMENSION R(1)
C*
C*       INITIALIZATION
C*
      T=0.0
      Y=0.0
    5 X=1.0E38
      IND=0
C*
C*       FIND NEXT LARGEST RANK
C*
      DO 30 I=1,N
      IF(R(NS+I)-Y) 30,30,10
   10 IF(R(NS+I)-X) 20,30,30
   20 X=R(NS+I)
      IND=IND+1
   30 CONTINUE
C*
C*       IF ALL RANKS HAVE BEEN TESTED, RETURN
C*
      IF(IND) 90,90,40
   40 Y=X
      CT=0.0
C*
C*       COUNT TIES
C*
      DO 60 I=1,N
      IF(R(NS+I)-X) 60,50,60
   50 CT=CT+1.0
   60 CONTINUE
C*     IF (F.EQ.1) WRITE(14,*//)CT,X,T
C*
C*       CALCULATE CORRECTION FACTOR
```

```
C*
      IF(CT) 70,5,70
   70 IF(KT-1) 75,80,75
   75 T=T+CT*(CT-1.)/2.0
      GO TO 5
   80 T=T+(CT*CT*CT-CT)/12.0
      GO TO 5
   90 CONTINUE
      RETURN
      END
      SUBROUTINE RRANK(A,R,N,RS,T,NDF,NS)
C*
C*************************************************************************
C*       RANK A VECTOR OF VALUES
C*    USAGE
C*       CALL RANK(A,R,N)
C*    DESCRIPTION OF PARAMETERS
C*       A - INPUT VECTOR OF N VALUES
C*       R - OUTPUT VECTOR OF LENGTH N. SMALLEST VALJE IS RANKED 1,
C*           LARGEST IS RANKED N. TIES ARE ASSIGNED AVERAGE OF TIED
C*           RANKS
C*       N - NUMBER OF VALUES
C*    REMARKS
C*       NONE
C*    SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C*       NONE
C*    METHOD
C*       VECTOR IS SEARCHED FOR SUCCESSIVELY LARGER ELEMENTS. IF TIES
C*       OCCUR, THEY ARE LOCATED AND THEIR RANK VALUE COMPUTED.
C*       FOR EXAMPLE, IF 2 VALUES ARE TIED FOR SIXTH RANK, THEY ARE
C*       ASSIGNED A RANK OF 6.5 (=(6+7)/2)
C*************************************************************************
C*
      DIMENSION A(1),R(1)
C*
C*     INITIALIZATION
      DO 10 I=1,N
   10 R(NS+I)=0.0
C*
C*       FIND RANK OF DATA
C*
      DO 100 I=1,N
C*
C*       TEST WHETHER DATA POINT IS ALREADY RANKED
C*
      IF(R(NS+I)) 20, 20, 100
C*
C*       DATA POINT TO BE RANKED
C*
   20 SMALL=0.0
      EQUAL=0.0
      X=A(I)
      DO 50 J=1,N
      IF(A(J)-X) 30, 40, 50
```

```
C*.       COUNT NUMBER OF DATA POINTS WHICH ARE SMALLER
C*
C*
   30 SMALL=SMALL+1.0
      GO TO 50
C*
C*        COUNT NUMBER OF DATA POINTS WHICH ARE EQUAL
C*
   40 EQUAL=EQUAL+1.0
      R(NS+J)=-1.0
   50 CONTINUE
C*
C*        TEST FOR TIE
C*
      IF(EQUAL-1.0) 60, 60, 70
C*
C*        STORE RANK OF DATA POINT WHERE NO TIE
C*
   60 R(NS+I)=SMALL+1.0
      GO TO 100
C*
C*        CALCULATE RANK OF TIED DATA POINTS
C*
   70 P=SMALL + (EQUAL + 1.0)*0.5
      DO 90 J=I,N
      IF(R(NS+J)+1.0) 90, 80, 90
   80 R(NS+J)=P
   90 CONTINUE
  100 CONTINUE
      RETURN
      END
      SUBROUTINE TABLE
C*************************************************************************
C*                                                                       *
C*                                                                       *
C*    SUBROUTINE TABLE                                                   *
C*                                                                       *
C*         PURPOSE - THIS SUBROUTINE WRITES THE TABLE OF                 *
C*                   CONVERSION FACTORS THAT THE USER CAN                *
C*                   CHOOSE FROM WHEN RUNNING SIGTEST.                   *
C*                                                                       *
C*************************************************************************
      DIMENSION FACTOR(10)
      DATA FACTOR/3.4077,3.91993,4.37237,4.78214,5.15943,5.51093,
     *5.84131,6.14501,0,0/
C*                  WRITE THE HEADING
      WRITE(5,99)
      WRITE(5,100)
      WRITE(5,101)
      WRITE(5,102)
C*                  WRITE THE BODY OF THE TABLE
      J=0
      DO 10 I=15,50,5
        J=J+1
```

```
      WRITE(5,110)I,FACTOR(J)
   10 CONTINUE
C*
      WRITE(5,99)
C*
   99 FORMAT("0.....")
  100 FORMAT("0        TABLE OF CONVERSION FACTORS        ")
  101 FORMAT("0    NO. OF CASES                FACTOR")
  102 FORMAT("     ------------                ------")
  110 FORMAT(1X,8X,I2,19X,F7.5)
      RETURN
      END
```

```
#FILE (LAWSONTWO)STAT/SEMDIF/SORT ON PACK
50 $SET$LIST$LEVEL 3
100    IDENTIFICATION  DIVISION.
200   PROGRAM-ID. FILSRT.
300 *REMARKS.  GENERAL OUTLINE OF SORT .  B. METIVIER.
400    ENVIRONMENT DIVISION.
500   CONFIGURATION SECTION.
600   INPUT-OUTPUT SECTION.
700   FILE-CONTROL.
800        SELECT DISKIN ASSIGN TO
900        000020*  001200
1000         DISK
1100          .
1200        SELECT DISKOUT ASSIGN TO
1300        000020*  001200
1400        DISK
1500          .
1600      SELECT SORTF  ASSIGN TO SORT DISK.
1700   DATA DIVISION.
1800   FILE SECTION.
1900    FD DISKIN
2000        BLOCK CONTAINS  60 RECORDS
2300        VA OF ID IN-FILE.
2600   01 IN-REC.
2700        02 FIL  PIC X OCCURS 42.
3000   SD SORTF.
3100   01 S-REC.
3111        02  FIELD02 PIC X(0003).
3112        02  FIELD01 PIC X(0003).
3120        02  FIELD03 PIC X(0003).
3140        02 FILLER   PIC X(0033).
3200   FD DISKOUT
3300        BLOCK CONTAINS  60 RECORDS
3600        VA OF ID OUT-FILE SAVE-FACTOR 001.
3900   01 D-REC.
3950      02 FIL   PIC X OCCURS 42.
4200   WORKING-STORAGE SECTION.
4230   01 IDIN COMP WITH LOWER-BOUNDS.
4240      02 DUMDUM PIC 9(11) OCCURS 4.
4250   01 IN-FILE REDEFINES IDIN   PIC X(24).
4260   01 IDOUT  COMP WITH LOWER-BOUNDS.
4270      02 DUMDAT PIC 9(11) OCCURS 5.
4280   01 OUT-FILE REDEFINES IDOUT  PIC X(30).
4300   PROCEDURE DIVISION USING IDIN, IDOUT.
4400   START-START SECTION.
4500   SORT-PAR.
4700      SORT SORTF ON
4711         ASCENDING KEY FIELD01
4712         ASCENDING KEY FIELD02, FIELD03
4800          USING DISKIN
4900             GIVING DISKOUT.
4950       EXIT PROCEDURE.
#
```

```
$SET LIST
FILE   5(KIND="REMOTE",MYUSE="IO")
FILE   6(FILE="PLOT/PRT.",KIND="PRINTER")
FILE   7(FILE="PLOT/INPUT.",KIND="DISK",FILETYPE=7)
C*
C*****************************************************************************
C*                                                                           *
C*                                                                           *
C*      PROGRAM: STAT/SEMDIF/PLOT                                            *
C*                                                                           *
C*      PURPOSE: THIS PROGRAM PLOTS THE E,P,A VALUES FOR THE                 *
C*         MEANS DOUBLED AS CALCULATED BY THE PROGRAM STAT/                  *
C*         SEMDIF.                                                           *
C*                                                                           *
C*      PROCEDURE: THE INPUT FILE FOR THIS PROGRAM IS FORMED                 *
C*         BY RUNNING STAT/SEMDIF AND USING THE RESULTING                    *
C*         FILE WITH A SUFFIX OF "/OUTPUT" AS AS INPUT FILE.                 *
C*         THE FORMAT OF THE INPUT FILE IS AS FOLLOWS:                       *
C*               1            1-3      NUMBER OF CONCEPTS                  *
C*               2            1-3      CONCEPT IDENT NUMBER                *
C*                            4-10     MEAN  OF E                          *
C*                           11-18     MEAN  OF P                          *
C*                           19-25     MEAN  OF A                          *
C*                           26-30     SPACE                               *
C*                           31-41     CONCEPT NAME - ALPHA                *
C*          3 - (N+I)                  SAME AS RECORD 2 - ONE FOR          *
C*                                     EACH CONCEPT                        *
C*               (N+2)        1-3      CONCEPT IDENT NUMBER                *
C*                            4-10     STAND. ERROR OF MEAN OF E           *
C*                           11-18       "      "   "    "   "  P          *
C*                           19-25       "      "   "    "   "  A          *
C*       (N+3) - 2*N+1                 SAME AS RECORD N+2 - ONE FOR        *
C*                                     EACH CONCEPT                        *
C*      (2*N)+2 - END         1-90     OSGOOD DEE VALUES FOR CONCEPTS      *
C*                                                                           *
C*      THIS PROGRAM USES RECORDS 1-N+1                                      *
C*                                                                           *
C*      SINCE THE FIRST SET OF DATA IN THE FILE IS JUST THE MEANS            *
C*      AND NOT THE MEANS DOUBLED,  THE E,P,A VALUES ARE READ INTO           *
C*      ARRAYS "E","P",AND "A" AND ARE THEN DOUBLED.                         *
C*                                                                           *
C*      WHEN PLOTTING A SET OF E,P,A VALUES,  THE "P" VALUE                  *
C*      IS PLOTTED ON THE HORIZONTAL AXIS, THE "E" VALUE IS                  *
C*      PLOTTED ON THE VERTICAL AXIS, AND THE POINT ITSELF IS                *
C*      CODED ALPHABETICALLY ACCORDING TO WHAT THE "A" VALUE IS              *
C*      (THE TABLE OF VALUES FOR THE "A" IS PRODUCED BELOW THE               *
C*      GRAPH ITSELF).  THE SUBROUTINE "AVAR" CONVERTS THE "A"               *
C*      TO ITS PROPER CODE.  THE SUBROUTINE "CODE" PRINTS OUT THE            *
C*      TABLE OF VALUES FOR THE "A" VARIABLE.                                *
C*                                                                           *
C*      THE GRAPH IS SCALED AS FOLLOWS:                                      *
C*          HORIZONTAL AXIS - EACH "-" IS 1/6 OF 1                           *
C*          VERTICAL AXIS   - EACH "!" IS 1/5 OF 1                           *
C*      EACH AXIS IS SCALED FROM "1" TO "14"                                 *
```

```
C*                                                                     *
C*      WHEN THE GRAPH IS SENT TO THE PRINTER, THE ACTUAL E,P,A        *
C*      VALUE ALONG WITH THE CORRESPONDING CONCEPT NAME ARE PRINTED    *
C*      ON THE RIGHT-HAND SIDE OF THE AXIS.                            *
C*                                                                     *
C*                                                                     *
C*      WRITTEN BY MARCY L. METIVIER FOR E.D.LAWSON AT THE             *
C*      STATE UNIVERSITY COLLEGE OF NEW YORK AT FREDONIA,              *
C*      NEW YORK 14063.  COMPLETED AUGUST, 1983.                       *
C*                                                                     *
C***********************************************************************
C*
C*
      CHARACTER*1 AXIS(70,84), KEY, ALPHA(13), TITLE*27, CONCNM(109)*15
      DIMENSION E(109), P(109), A(109)
      INTEGER XAXIS, YAXIS, SCALE(14)
      DATA ALPHA/"A","B","C","D","E","F","G","H","I","J","K","L","M"/
C*
C*          INITIALIZE AXIS TO CONTAIN ALL BLANKS
C*
      DO 20 I=1,70
         DO 20 J=1,84
            AXIS(I,J)=" "
            CONCNM(J)=" "
   20 CONTINUE
C*
      WRITE(5,*)"ENTER NAME OF INPUT FILE.  NO MORE THAN 27 CHARACTERS
     *PLEASE.  (MUST  BE A /OUTPUT FILE...)"
      READ(5,900)TITLE
C*
      IF (TITLE(27:).NE." ") THEN
          TITLE(27:)="."
          GO TO 90
      ELSE
          DO 50 I=22,1,-1
   50        IF (TITLE(I:I).NE." ") GO TO 60
   60     I=I+1
          TITLE(I:I)="."
      END IF
C*
C*            WRITE FILENAME ON PRINTOUT
C*
      WRITE(6,960)TITLE
C*
C*             OPEN INPUT FILE
C*
   90 OPEN(7,FILE=TITLE)
      READ(7,905)N
C*
C*             READ IN MEANS E,P,A VALUES AND DOUBLE THEM
C*
      DO 100 I=1,N
         READ(7,910)E(I),P(I),A(I),CONCNM(I)
         E(I)=E(I)*2
```

```
          P(I)=P(I)*2
          A(I)=A(I)*2
          XAXIS=NINT((P(I)-1)*6)
          YAXIS=NINT((E(I)-1)*5)
C*
C*             FIND CORRECT "CODE" FOR THE A VALUE AND "PLOT" IT
C*             ON THE "GRAPH".
C*
          CALL AVAR(A(I),KEY,ALPHA)
          AXIS(YAXIS,XAXIS)=KEY
  100 CONTINUE
C*
C*             PRINT "GRAPH"
C*
       L=0
       K=66
C*
       DO 200 I=14,2,-1
          L=L+1
          K=K-1
C*             WRITE ACTUAL E,P,A VALUES UNTIL THERE ARE NO MORE
          IF (L.LE.N) THEN
            WRITE(6,920)I,(AXIS(K,J),J=2,84),E(L),P(L),A(L),CONCNM(L),L
          ELSE
            WRITE(6,920)I,(AXIS(K,J),J=2,84)
            END IF
C*
          SCALE(I)=I
C*
          DO 150 LL=1,4
             L=L+1
             K=K-1
C*
             IF (L.LE.N) THEN
               WRITE(6,930)(AXIS(K,J),J=2,84),E(L),P(L),A(L),CONCNM(L),L
             ELSE
               WRITE(6,930)(AXIS(K,J),J=2,84)
               END IF
C*
   150    CONTINUE
   200 CONTINUE
       WRITE(6,940)
       WRITE(6,950)(SCALE(I),I=2,14)
       CALL CODE(ALPHA)
       WRITE(6,970)
       STOP
C*
   900 FORMAT(A27)
   905 FORMAT(I3)
   910 FORMAT(4X,3F7.3,5X,A15)
   920 FORMAT(1X,I2," +",83A1,3F7.3,3X,A15,1X,I3)
   930 FORMAT(4X,"!",83A1,3F7.3,3X,A15,1X,I3)
   940 FORMAT(2X,"1 ",13("-----+"),"   P")
   950 FORMAT("0",3X,13(4X,I2))
```

```
  960 FORMAT("1","   E",12X,"E VS. P VS. A PLOT FOR ",A27,22X,"E",8X,
     *"P",8X,"A")
  970 FORMAT("0NOTE:  THE PROGRAM CAN ONLY PLOT ONE CONCEPT IN A SINGLE
     *SPACE."/" THEREFORE, IF THERE ARE TWO OR MORE CONCEPTS IN THE SAME
     * PLACE ONLY ONE LETTER WILL APPEAR.")
      END
C*
C*****************************************************************
C*                                                               *
C*            SUBROUTINE AVAR                                    *
C*                                                               *
C*     USAGE:    AVAR(VALUE,KEY,ALPHA)                           *
C*                                                               *
C*     PURPOSE:  THIS SUBROUTINE FINDS THE CORRECT ALPHABETIC    *
C*               CHARACTER FOR VALUE AND RETURNS IT IN KEY       *
C*                                                               *
C*     VARIABLES:                                                *
C*               ALPHA: CHARACTER ARRAY CONTAINING ALPHABETIC    *
C*                      CHARACTERS FROM "A" TO "M"               *
C*               POS  : CONVERSION FACTOR FOR VALUE              *
C*               VALUE: REAL NUMBER TO BE CONVERTED TO ALPHA     *
C*               KEY  : CONTAINS THE ALPHABETIC CHARACTER        *
C*                      VALUE IS CONVERTED TO                    *
C*                                                               *
C*****************************************************************
C*
      SUBROUTINE AVAR(VALUE,KEY,ALPHA)
      CHARACTER*1 ALPHA(13), KEY
      INTEGER POS
C*
C*
      POS=INT(VALUE)-1
      KEY=ALPHA(POS)
      RETURN
      END
C*
C*****************************************************************
C*                                                               *
C*          SUBROUTINE CODE                                      *
C*                                                               *
C*       USAGE:      CALL CODE(ALPHA)                            *
C*       PURPOSE:    TO WRITE THE CODE FOR THE "A" VARIABLE      *
C*                   IN STAT/SEMDIF/PLOT                         *
C*                                                               *
C*       ALPHA IS A CHARACTER*1 ARRAY CONTAINING THE LETTERS     *
C*       FROM A TO M.                                            *
C*                                                               *
C*                                                               *
C*****************************************************************
C*
      SUBROUTINE CODE(ALPHA)
      CHARACTER*1 ALPHA(13)
C*
      WRITE(6,900)
```

```
      DO 100 I=1,6
         A=I+1
         K=A+6
         WRITE(6,910)A,A,ALPHA(I),K,K,ALPHA(I+6)
  100 CONTINUE
      WRITE(6,920)ALPHA(13)
C*
  900 FORMAT(/"0THE KEYS FOR THE 3RD VARIABLE (THE 'A' VARIABLE ARE..."
     */)
  910 FORMAT(2(4X,I2,".00"," - ",I2,".99"," = ",A1))
  920 FORMAT(18X,"14.00 = ",A1)
      RETURN
      END
```

```
 $SET LIST
FILE  4(KIND="REMOTE",MYUSE="OUT")
FILE  5(KIND="REMOTE",MYUSE="IN")
FILE  6(KIND="PRINTER")
FILE  10(FILE="FORMS/INPUT.",KIND="DISK",FILETYPE=7)
C*****************************************************************
C*                                                               *
C*                                                               *
C*     TITLE : SEMDIF/FORMS/PRG                                  *
C*                                                               *
C*     AUTHOR : MARCY METIVIER                                   *
C*                                                               *
C*       THIS PROGRAM WRITES OUT SURVEY FORMS FOR THE            *
C*     SEMDIF PROGRAMS.                                          *
C*                                                               *
C*       THE PROGRAM FIRST ASKS FOR THE NUMBER OF COPIES         *
C*     TO BE MADE.  THEN THE NUMBER OF CONCEPTS IS ASKED         *
C*     (INUM) AND A TITLE FOR THE PRINTOUTS IS ENTERED.          *
C*     THE USER THEN HAS A CHOICE OF ENTERING THE CONCEPTS       *
C*     FROM AN EXTERNAL FILE OR FROM THE TERMINAL.               *
C*     A MAXIMUM OF 109 CONCEPTS ARE ALLOWED.                    *
C*                                                               *
C*       ONCE THE CONCEPTS ARE READ IN INTO THE ARRAY CONCNM     *
C*     THE SUBROUTINE RAN IS CALLED WHICH PUTS THE CONCEPT       *
C*     NAMES IN RANDOM ORDER INTO THE ARRAY RANDNM.              *
C*     THE FORMS ARE THEN PRINTED OUT WITH 6 "CARDS" TO A        *
C*     PAGE.  IF MORE THAN ONE COPY IS REQUESTED, THEN RAN       *
C*     IS CALLED AGAIN SO THAT THE CONCEPTS WILL BE IN A         *
C*     DIFFERENT ORDER FOR EACH COPY.  EACH COPY WILL BE         *
C*     ASSIGNED A UNIQUE "CASE" NUMBER WHICH IS PRINTED          *
C*     AFTER THE TITLE ON EACH PAGE.                             *
C*                                                               *
C*                                                               *
C*                                                               *
C*                                                               *
C*****************************************************************
C*
C*
C*
      DIMENSION NUMBER(109)
      CHARACTER CONCNM(109)*15, RANDNM(109)*15, CAT1(9)*9, CAT2(9)*9
      CHARACTER TITLE*18, HEAD*25
      DATA CAT1/"KIND","WEAK","FAST","COLD","LARGE","DISHONEST","HAPPY",
     *     "DELICATE","SHARP"/
      DATA CAT2/"CRUEL","STRONG","SLOW","HOT","SMALL","HONEST","SAD",
     *     "RUGGED","DULL"/
C*
C*     INITIALIZE THE ARRAY OF CONCEPTS TO CONTAIN ALL BLANKS
C*
      DO 20 I=1,109
   20       RANDNM(I)="               "
C*
C*
      WRITE(4,*)"PLEASE ENTER THE NUMBER OF COPIES TO BE MADE."
```

```
      READ(5,990)ICOPY
      WRITE(4,*)"PLEASE ENTER THE NUMBER OF CONCEPTS TO BE USED."
      READ(5,990)INUM
      WRITE(4,*)"PLEASE ENTER THE TITLE FOR THE PRINTOUTS."
      READ(5,945)HEAD
      WRITE(4,*)"DO YOU WANT THE CONCEPTS TO BE ENTERED FROM A FILE OR
     *FROM THE          TERMINAL? (1=FILE, 2=TERMINAL) "
      READ (5,925)IWHAT
      IF (IWHAT.EQ.2) THEN
         WRITE(4,995)
         DO 50 I=1,INUM
            WRITE(4,980)I
            READ(5,901)CONCNM(I)
   50    CONTINUE
      ELSE
         WRITE(4,*)"ENTER NAME OF INPUT FILE (LESS THAN 18 CHARACTERS)."
         READ(5,935)TITLE
         IF (TITLE(18:).NE." ") THEN
            TITLE(18:)="."
            GO TO 65
         ELSE
            DO 55 I=17,1,-1
   55          IF (TITLE(I:I).NE." ") GO TO 60
   60       I=I+1
           TITLE(I:I)="."
         END IF
C*
C*  OPEN INPUT FILE AND READ CONCEPTS
C*
   65    OPEN (10, FILE=TITLE)
         DO 67 I=1,INUM
   67    READ(10,900)CONCNM(I)
      END IF
C*
      IXNUM=1
C*
C*  NOW WRITE OUT THE SHEETS
C*
      DO 800 I=1,ICOPY
         CALL RAN(CONCNM,RANDNM,NUMBER,INUM,IXNUM)
         WRITE(6,905)
           WRITE(6,913)
C*
        J=1
   70  IF (J.LE.INUM) THEN
         ICOUNT=1
         WRITE(6,920)
         WRITE(6,910)HEAD,I, HEAD,I
         WRITE(6,915)
C*
  100     IF (ICOUNT.LE.3.AND.J.LE.INUM) THEN
            JJ=J+1
            WRITE(6,930)NUMBER(J),RANDNM(J),NUMBER(JJ),RANDNM(JJ)
            WRITE(6,940)
```

```
C*
                DO 200 K=1,9
                   WRITE(6,950)CAT1(K),CAT2(K),CAT1(K),CAT2(K)
                   WRITE(6,940)
  200           CONTINUE
C*
  300           WRITE(6,920)
                ICOUNT=ICOUNT+1
                J=J+2
                GO TO 100
              END IF
C*
              WRITE(6,970)
            GO TO 70
          END IF
C*
  800 CONTINUE
      STOP
C*
C*
  900 FORMAT(4X,A15)
  901 FORMAT(A15)
  905 FORMAT("1",2(55("-"),5X))
  910 FORMAT("0",2("SEMANTIC DIFFERENTIAL ",A25,2X,I3,8X))
  913 FORMAT("0SEX  M  F    DATE_____   INSTRUCTOR_____
     *HR_____")
  915 FORMAT("0",2("PLEASE CIRCLE THE BEST ASSOCIATION FOR CONCEPT.",
     *13X))
  920 FORMAT("0",2(55("*"),5X))
  925 FORMAT(I2)
  930 FORMAT(/"0",2(6X,I3,".  ",A15,35X))
  935 FORMAT(A18)
  940 FORMAT(1X,2(55("-"),5X))
  945 FORMAT(A25)
  950 FORMAT(1X,2("!",2X,A9,3X,"1   2   3   4   5   6   7",5X,A9,"!",
     *5X))
  970 FORMAT("1")
  980 FORMAT(" PLEASE ENTER CONCEPT NO.",I3,"...")
  990 FORMAT(I3)
  995 FORMAT(" EACH CONCEPT NAME MUST BE LESS THAN 16 CHARACTERS.")
      END
C*
C*
C*
C*****************************************************************
C*                                                               *
C*                                                               *
C*    SUBROUTINE RAN                                             *
C*                                                               *
C*      THIS SUBROUTINE RANDOMIZES THE PRINTOUT ORDER OF THE     *
C*   CONCEPTS STORED IN CONCNM.                                  *
C*                                                               *
C*****************************************************************
C*
```

```
C*
      SUBROUTINE RAN(CONCNM, RANDNM, NUMBER, INUM, IXNUM)
      DIMENSION NUMBER(109), LIST(109)
      CHARACTER CONCNM(109)*15, RANDNM(109)*15
C*
C* SET THE NUMBER ARRAY
C*
      DO 10 I=1,109
   10    NUMBER(I)=0
C*
C* INITIALIZE LIST TO CONTAIN ALL ZEROS (0)
C*
      DO 50 I=1,109
   50    LIST(I)=0
C*
      DO 100 II=1,INUM
   70    CALL RNDINT(IXNUM,JNUM,1,INUM)
         IF (LIST(JNUM).EQ.1) GO TO 70
         LIST(JNUM)=1
         NUMBER(II)=JNUM
         RANDNM(II)=CONCNM(JNUM)
  100  CONTINUE
C*
      RETURN
      END
C*
C*
C*
C**************************************************************
C*                                                            *
C*                                                            *
C*          SUBROUTINE RNDINT                                 *
C*                                                            *
C*    SUBROUTINE TO GENERATE THE RANDOM NUMBER                *
C*       IX : THE RANDOM INTRINSIC ARGUMENT                   *
C*       J  : THE RANDOM NUMBER RETURNED BY THE RANDOM INTRINSIC *
C*       K  : THE INCLUSIVE MINIMUM VALUE OF THE RANGE        *
C*       L  : THE INCLUSIVE MAXIMUM VALUE OF THE RANGE        *
C*                                                            *
C**************************************************************
C*
C*
C*
      SUBROUTINE RNDINT(IX,J,K,L)
      Y=RANDOM(IX)
      Q1=FLOAT(L-K+1)
      Q2=FLOAT(K)
      J=INT(Q1*Y+Q2)
      RETURN
      END
```

```
$SET LIST
FILE  7(KIND="DISK",TITLE="DCKIN1",FILETYPE=7)
FILE  8(KIND="DISK",TITLE="DCKIN2",FILETYPE=7)
FILE  9(KIND="PRINTER")
C*****************************************************************
C*                                                               *
C*        PROGRAM:  SEMDIF/TTEST                                 *
C*                                                               *
C*        THIS PROGRAM TEST THE SIGNIFICANCE OF THE DIFFERENCES  *
C*    BETWEEN THE MEANS OF TWO POPULATIONS.  BY MEANS OF A T-TEST,*
C*    THE INPUT DATA IS STORED IN 2 FILES, CREATED BY THE PROGRAM *
C*    STAT/SEMDIF WITH THE SUFFIX "/OUTPUT".                     *
C*                                                               *
C*                                                               *
C*        THE INPUT FILE HAS THE FOLLOWING FORMAT:               *
C*             RECORD        COLS      CONTENT                  *
C*               1            1-3      NUMBER OF CONCEPTS       *
C*               2            1-3      CONCEPT IDENT NUMBER     *
C*                            4-10     MEAN  OF E               *
C*                           11-18     MEAN  OF P               *
C*                           19-25     MEAN  OF A               *
C*                           26-30     SPACE                    *
C*                           31-41     CONCEPT NAME - ALPHA     *
C*          3 - (N+1)                  SAME AS RECORD 2 - ONE FOR*
C*                                     EACH CONCEPT             *
C*            (N+2)           1-3      CONCEPT IDENT NUMBER     *
C*                            4-10     STAND. ERROR OF MEAN OF E*
C*                           11-18      "      "   "   "  P     *
C*                           19-25      "      "   "   "  A     *
C*       (N+3) - 2*N+1                 SAME AS RECORD N+2 - ONE FOR*
C*                                     EACH CONCEPT             *
C*    (2*N)+2 - END           1-90     OSGOOD DEE VALUES FOR CONCEPTS*
C*                                                               *
C*    THIS PROGRAM ONLY USES THE RECORDS 1-(2*N)+1               *
C*                                                               *
C*    PROGRAM WRITTEN BY M.L. METIVIER FOR E.D.LAWSON AT STATE   *
C*    UNIVERSITY COLLEGE, FREDONIA, NEW YORK, 14063.             *
C*    COMPLETED AUGUST, 1983                                     *
C*                                                               *
C*****************************************************************
      CHARACTER*28 TITLE1,TITLE2
      CHARACTER*8 WHEN
      DIMENSION ARRAY1(109,3),ARRAY2(109,3),ARRAY3(109,3),ARRAY4(109,3),
     *          ARRAY5(3),BLANK(1,3)
C*
C*    PRINT DATE ON PRINTOUT
C*
      CALL DATIT(WHEN)
      WRITE(9,900)WHEN
C*
C*
   10 WRITE(6,930)
      READ(5,950)TITLE1
      WRITE(6,940)
```

```
      READ(5,950)TITLE2
C*
C*      PUT PERIOD AT END OF FIRST FILENAME
C*
      DO 200 I=1,27
         IF(TITLE1(I:I).EQ." ") THEN
            TITLE1(I:I)="."
            GO TO 210
         END IF
200   CONTINUE
      TITLE1(28:28)="."
210   OPEN (7,TITLE=TITLE1
C*      REPEAT FOR SECOND FILENAME
C*
      DO 250 I=1,27
         IF(TITLE2(I:I).EQ." ")THEN
            TITLE2(I:I)="."
            GO TO 260
         END IF
250   CONTINUE
      TITLE2(28:28)="."
260   OPEN (8,TITLE=TITLE2)
      READ(7,920)INUM
      READ(8,920)INUM
C*
C*      NOW READ BOTH PARTS OF THE FILES INTO ARRAYS 1-4
C*
      DO 300 I=1,INUM
         READ (7,800)(ARRAY1(I,J),J=1,3)
         READ (8,800)(ARRAY2(I,J),J=1,3)
300   CONTINUE
      DO 310 I=1,INUM
         READ (7,800)(ARRAY3(I,J),J=1,3)
         READ (8,800)(ARRAY4(I,J),J=1,3)
310   CONTINUE
C*
C*      NOW CALCULATE AND PRINT THE RESULTS OF THE T-TEST
C*
      WRITE(9,910)TITLE1,TITLE2
      DO 400 I=1,INUM
         DO 350 J=1,3
            ARRAY5(J)=ABS(ARRAY1(I,J)-ARRAY2(I,J))/SQRT(ARRAY3(I,J)**
     *                2+ARRAY4(I,J)**2)
350      CONTINUE
         WRITE(9,850)I,(ARRAY5(J),J=1,3)
400   CONTINUE
      STOP
C*
  800 FORMAT(4X,3F7.3)
  850 FORMAT("0",I3,3X,F7.5,3X,F7.5,3X,F7.5)
  900 FORMAT(1X,A8//)
  910 FORMAT(" T-TEST RESULTS FOR FILES ",A28," AND ",A28)
  920 FORMAT(I3)
```

```
930 FORMAT(" ENTER 1ST FILENAME-TO MORE THAN 27 CHARACTERS PLEASE...")
940 FORMAT(" ENTER 2ND FILENAME-TO MORE THAN 27 CHARACTERS PLEASE...")
950 FORMAT(A27)
    END
C*
C*
    SUBROUTINE DATIT(WHEN)
    CHARACTER*8 WHEN
    CHARACTER*6 B
    DATA WHEN/"  /  /  "/
    B=DATE("MMDDYY")
    WHEN(1:2)=B(1:2)
    WHEN(4:5)=B(3:4)
    WHEN(7:8)=B(5:6)
    RETURN
    END
```

```
$READ" FREE JOB LIST
FILE  7(KIND="DISK",TITLE="D.MIN1",FILETYPE=7)
FILE  8(KIND="DISK",TITLE="D.MIN2",FILETYPE=7)
FILE  9(KIND="PRINTER")
C*****************************************************************
C*                                                               *
C*                                                               *
C*                  PROGRAM:  STAT/SEMDIF/CORR                   *
C*                                                               *
C*                                                               *
C*     THIS PROGRAM INPUTS TWO FILES EACH CONTAINING THE OSGOOD D *
C*     VALUES BETWEEN EACH AND EVERY CONCEPT.  THESE DISTANCES WERE *
C*     CALCULATED BY THE  SUBROUTINE, SDGPD,IN THE PROGRAM, SEMDIF. *
C*     SEMDIF OUTPUTS THIS DATA IN A PRINTED REPORT.  TO OBTAIN THE *
C*     DATA, SEMDIF WAS RUN AND THE RESULTING /OUTPUT FILE IS USED *
C*     AS THE INPUT FILE.                                        *
C*     THIS PROGRAM THEN COMPUTES THE SPEARMAN RANK CORRELATION  *
C*     COEFFICIENT FOR THE TWO (2) GROUPS OF DATA.  IT CAN HANDLE *
C*     A DISTANCE MATRIX FOR UP TO 109 CONCEPTS INCLUSIVE.  EACH TIME *
C*     THE PROGRAM FINISHED PROCESSING 2 FILES, IT RETURNS FOR 2 MORE *
C*     FILES UNTIL THE USER HAS FINISHED HIS/HER JOB.            *
C*                                                               *
C*     THE FOLLOWING IS THE FORMAT OF THE INPUT FILES:           *
C*              RECORD        COLS      CONTENT                  *
C*                1             1-3     NUMBER OF CONCEPTS       *
C*                2             1-3     CONCEPT IDENT NUMBER     *
C*                              4-10    MEAN  OF E               *
C*                             11-28    MEAN  OF P               *
C*                             29-25    MEAN  OF A               *
C*                             26-30    SPACE                    *
C*                             31-41    CONCEPT NAME - ALPHA     *
C*          3 - (N+1)                   SAME AS RECORD 2 - ONE FOR *
C*                                      EACH CONCEPT             *
C*             (N+2)            1-3     CONCEPT IDENT NUMBER     *
C*                              4-10    STAND. ERROR OF MEAN OF E *
C*                             11-28     "      "   "   "   "  P *
C*                             29-25     "      "   "   "   "  A *
C*       (N+3) - 2*N+1                  SAME AS RECORD N+2 - ONE FOR *
C*                                      EACH CONCEPT             *
C*     (2*N)+2 - END             1-90    OSGOOD, DEE VALUES FOR CONCEPTS *
C*                                                               *
C*       THIS PROGRAM ONLY USES RECORDS (2*N)+2 - END.           *
C*                                                               *
C*     THE RESULTS OF THIS SUBROUTINE PLUS THE FIRST TEN VALUES  *
C*     THAT THIS PROGRAM USES FROM EACH OF THE INPUT FILES ARE SENT *
C*     TO THE PRINTER.                                           *
C*                                                               *
C*     THIS SUBROUTINE USES THE IBM SUBROUTINES RANK, TIE, AND SRANK. *
C*                                                               *
C*     PROGRAM WRITTEN BY M.L.METIVIER FOR S.D.LAWSON AT STATE   *
C*     UNIVERSITY COLLEGE, FREDONIA, NEW YORK, 14063.  COMPLETED *
C*     AUGUST 29, 1983.                                          *
C*                                                               *
C*                                                               *
```

```
C*****************************************************************
C*
C*
C*
      CHARACTER*1 K
      CHARACTER*2 WHEN
      CHARACTER*29 TITLE1,TITLE2
      DIMENSION A(5386),B(5386),R(11772),BLANK(1,3)
C*
C*       PRINT DATE ON TOP OF PRINTOUT
C*
      CALL DATE(WHEN)
      WRITE(9,920)WHEN
C*
   10 WRITE(6,*)"ENTER 1ST FILENAME PLEASE-NO MORE THAN 28 CHARACTERS.."
      READ(5,900)TITLE1
      WRITE(6,*)"ENTER 2ND FILENAME PLEASE-NO MORE THAN 28 CHARACTERS.."
      READ(5,900)TITLE2
C*
C*          PUT PERIOD AT END OF FIRST FILENAME
C*
      DO 50 I=1,28
         IF(TITLE1(I:I).EQ." ")THEN
            TITLE1(I:I)="."
            GO TO 60
         END IF
   50 CONTINUE
      TITLE1(29:29)="."
   60 OPEN (7,TITLE=TITLE1)
C*
C*            RE EAT FOR SECOND FILENAME
C*
      DO 100 I=1,28
         IF(TITLE2(I:I).EQ." ")THEN
            TITLE2(I:I)="."
            GO TO 110
         END IF
  100 CONTINUE
      TITLE2(29:29)="."
  110 OPEN (8,TITLE=TITLE2)
      REWIND 7
      REWIND 8
C*
C*
C*       N = N3 = NUMBER OF CONCEPTS TO BE PROCESSED
C*       N1= N4 = NUMBER OF 1ST CASE
C*       N2= N5 = NUMBER OF LAST CASE
C*
C*
      READ(7,930)K
      READ(8,930)N
      L=(N*(N-1))/2
      NN=0
C*
```

```
C*          SKIP OVER 1ST PART OF FILE
C*
      KK=K*2
      DO 120 I=1,KK
         READ(7,940)(BLANK(1,J),J=1,3)
         READ(8,940)(BLANK(1,J),J=1,3)
  120 CONTINUE
C*
C*          READ IN INPUT DATA
C*
      READ(7,*)(A(I),I=1,N)
      READ(8,*)(B(I),I=1,N)
C*
C*           CALL SRANK AND PRINT OUTPUT
C*
      WRITE(9,910)TITLE1,TITLE2
      CALL SRANK (A,B,R,N,RS,T,NDF,NR)
      WRITE(9,960) RS,T,NDF
C*              REPEAT WITH MORE FILES?
      WRITE(6,*)"DO YOU WISH TO USE 2 MORE FILES?"
      READ(5,950)X
      IF (X.EQ."Y") GO TO 10
      CLOSE(7)
      CLOSE(8)
      STOP
  900 FORMAT(A20)
  910 FORMAT(//,1X,A20," VS ",A20)
  920 FORMAT(1X,A8//)
  930 FORMAT(I5)
  940 FORMAT(4X,3F7.3)
  950 FORMAT(A1)
  960 FORMAT ("0 SPEARMAN RANK CORRELATION COEFFICIENT=",F5.2/
     *" ,'I NIFICANCE=",F8.3/" NUMBER OF DEGREES OF FREEDOM=",I5)
      END
      SUBROUTINE SRANK(A,B,R,N,RS,T,NDF,NR)
C*************************************************************
C*
C*     SUBROUTINE SRANK
C*
C*     PURPOSE
C*        TEST CORRELATION BETWEEN TWO VARIABLES BY MEANS OF SPEARMAN
C*        RANK CORRELATION COEFFICIENT
C*
C*     USAGE
C*        CALL SRANK(A,B,R,N,RS,T,NDF,NR)
C*
C*     DESCRIPTION OF PARAMETERS
C*        A    - INPUT VECTOR OF N OBSERVATIONS FOR FIRST VARIABLE
C*        B    - INPUT VECTOR OF N OBSERVATIONS FOR SECOND VARIABLE
C*        R    - OUTPUT VECTOR FOR RANKED DATA, LENGTH IS 2*N. SMALLEST
C*               OBSERVATION IS RANKED 1, LARGEST IS RANKED N. TIES
C*               ARE ASSIGNED AVERAGE OF TIED RANKS.
C*        N    - NUMBER OF OBSERVATIONS
C*        RS   - SPEARMAN RANK CORRELATION COEFFICIENT (OUTPUT)
```

```
C*          T   - TEST OF SIGNIFICANCE OF RJ (OUTPUT)
C*          DF  - NUMBER OF DEGREES OF FREEDOM (OUTPUT)
C*          KR  - CODE, 0 FOR UNRANKED DATA IN A AND B, 1 FOR RANKED
C*                DATA IN A AND B (INPUT)
C*
C*     REMARKS
C*          T IS SET TO ZERO IF N IS LESS THAN TEN
C*
C*     SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C*          RANK
C*          TIE
C*
C*     METHOD
C*          DESCRIBED IN S. SIEGEL, "NONPARAMETRIC STATISTICS FOR THE
C*          BEHAVIORAL SCIENCES", MCGRAW-HILL, NEW YORK, 1956,
C*          CHAPTER 9
C*********************************************************************
C*
      DIMENSION A(1),B(1),R(1)
C*
      FN=FLOAT(N)
      FACT=(FN**3)-FN
C*
C*          DETERMINE WHETHER DATA IS RANKED
C*
      IF(KR-1) 5, 10, 5
C*
C*          RANK DATA IN A AND B VECTORS AND ASSIGN TIED OBSERVATIONS
C*          AVERAGE OF TIED RANKS
C*
    5 NS=0
      CALL RANK (A,R,N,RS,T,NDF,NS)
      NS=N
      CALL RANK (B,R,N,RS,T,NDF,NS)
      GO TO 40
C*
C*          MOVE RANKED DATA TO R VECTOR
C*
   10 DO 20 I=1,N
   20 R(I)=A(I)
      DO 30 I=1,N
      J=I+N
   30 R(J)=B(I)
C*
C*          COMPUTE SUM OF SQUARED OF RANK DIFFERENCES
C*
   40 D=0.0
      DO 50 I=1,N
      J=I+N
   50 D=D+(R(I)-R(J))*(R(I)-R(J))
C*
C*          COMPUTE TIED SCORE INDEX
C*
      KT=1
```

```
      M=0
      CALL TIE (R,N,KT,TSA,M)
      M=M
      CALL TIE (R,N,KT,TSB,M)
C*
C*       COMPUTE SPEARMAN RANK CORRELATION COEFFICIENT
C*
      IF(TSA) 60,55,60
   55 IF(TSB) 60,57,60
   57 RS=1.0-6.0*D/FNNN
      GO TO 70
   60 X=FNNN/12.0-TSA
      Y=FNNN/12.0-TSB
      RS=(X+Y-D)/(2.0*(SQRT(X*Y)))
      WRITE(9,900)FNNN,X,Y,D
C*
C*       COMPUTE T AND DEGREES OF FREEDOM IF N IS 10 OR LARGER
C*
      T=0.0
   70 IF(N-10) 80,75,75
   75 CONTINUE
      T=RS*SQRT(FLOAT(N-2)/(1.0-RS*RS))
   80 NDF=N-2
      RETURN
  900 FORMAT("0FNNN=",G10.4,"  X=",G10.4,"  Y=",G10.4,"  D=",G10.4)
      END
      SUBROUTINE TIE(R,N,KT,T,NS)
C*
C*************************************************************************
C*     SUBROUTINE TIE
C*
C*     PURPOSE
C*        CALCULATE CORRECTION FACTOR DUE TO TIES
C*
C*     USAGE
C*        CALL TIE(R,N,KT,T)
C*
C*     DESCRIPTION OF PARAMETERS
C*        R  - INPUT VECTOR OF RANKS OF LENGTH N CONTAINING VALUES
C*             1 TO N
C*        N  - NUMBER OF RANKED VALUES
C*        KT - INPUT CODE FOR CALCULATION OF CORRECTION FACTOR
C*                   1   SOLVE EQUATION 1
C*                   2   SOLVE EQUATION 2
C*        T  - CORRECTION FACTOR (OUTPUT)
C*                EQUATION 1   T=SUM(CT**3-CT)/12
C*                EQUATION 2   T=SUM(CT*(CT-1)/2)
C*              WHERE CT IS THE NUMBER OF OBSERVATIONS TIED FOR A
C*                    GIVEN RANK
C*     REMARKS
C*        NONE
C*
C*     SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C*        NONE
```

```
C*
C*      RANK
C*         VECTOR IS REPLACED BY SUCCESSIVELY LARGER RANKS. TIES ARE
C*         COUNTED AND CORRECTION FACTOR 1 OR 2 SUMMED.
C*
C*****************************************************************
C*
      DIMENSION R(1)
C*
C*         INITIALIZATION
C*
      T=0.0
      Y=0.0
    5 X=1.0E38
      IND=0
C*
C*         FIND NEXT LARGEST RANK
C*
      DO 30 I=1,N
      IF(R(NS+I)-Y) 30,30,10
   10 IF(R(NS+I)-X) 20,30,30
   20 X=R(NS+I)
      IND=IND+1
   30 CONTINUE
C*
C*         IF ALL RANKS HAVE BEEN TESTED, RETURN
C*
      IF(IND) 90,90,40
   40 Y=X
      CT=0.0
C*
C*         COUNT TIES
C*
      DO 60 I=1,N
      IF(R(NS+I)-X) 60,50,60
   50 CT=CT+1.0
   60 CONTINUE
C*
C*         CALCULATE CORRECTION FACTOR
C*
      IF(CT) 70,5,70
   70 IF(CT-1) 75,80,75
   75 T=T+CT*(CT-1.)/2.0
      GO TO 5
   80 T=T+(CT*CT*CT-CT)/12.0
      GO TO 5
   90 CONTINUE
      RETURN
      END
      SUBROUTINE RANK(A,R,N,NS,T,NDF,MS)
C
C*****************************************************************
C*         RANK A VECTOR OF VALUES
C*     USAGE
```

```
C*       CALL RANK(A,R,N)
C*    DESCRIPTION OF PARAMETERS
C*       A - INPUT VECTOR OF N VALUES
C*       R - OUTPUT VECTOR OF LENGTH N. SMALLEST VALUE IS RANKED 1,
C*           LARGEST IS RANKED N. TIES ARE ASSIGNED AVERAGE OF TIED
C*           RANKS
C*       N - NUMBER OF VALUES
C*    REMARKS
C*       NONE
C*    SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C*       NONE
C*    METHOD
C*       VECTOR IS SEARCHED FOR SUCCESSIVELY LARGER ELEMENTS. IF TIES
C*       OCCUR, THEY ARE LOCATED AND THEIR RANK VALUE COMPUTED.
C*       FOR EXAMPLE, IF 2 VALUES ARE TIED FOR SIXTH RANK, THEY ARE
C*       ASSIGNED A RANK OF 6.5 (=(6+7)/2)
C*************************************************************
C*
      DIMENSION A(1),R(1)
C*
C*       INITIALIZATION
      DO 10 I=1,N
   10 R(N+I)=0.0
C*
C*       FIND RANK OF DATA
C*
      DO 100 I=1,N
C*
C*       TEST WHETHER DATA POINT IS ALREADY RANKED
C*
      IF(R(N+I)) 20, 30, 100
C*
C*       DATA POINT TO BE RANKED
C*
   20 SMALL=0.0
      EQUAL=0.0
      X=A(I)
      DO 50 J=1,N
      IF(A(J)-X) 30, 40, 50
C*       COUNT NUMBER OF DATA POINTS WHICH ARE SMALLER
C*
C*
   30 SMALL=SMALL+1.0
      GO TO 50
C*
C*       COUNT NUMBER OF DATA POINTS WHICH ARE EQUAL
C*
   40 EQUAL=EQUAL+1.0
      R(N+J)=-1.0
   50 CONTINUE
C*
C*       TEST FOR TIE
C*
      IF(EQUAL-1.0) 60, 60, 70
```

```
C*
C*        ASSIGN RANK TO DATA POINT WHERE NO TIE
C*
   60 R(NS+I)=SMALL+1.0
      GO TO 100
C*
C*        CALCULATE RANK OF TIED DATA POINTS
C*
   70 P=SMALL + (EQUAL + 1.0)*0.5
      DO 90 J=1,K
      IF(R(NS+J)+1.0) 90, 80, 90
   80 R(NS+J)=P
   90 CONTINUE
  100 CONTINUE
      RETURN
      END
      SUBROUTINE DATIT(WHEN)
      CHARACTER*8 WHEN
      CHARACTER*6 B
      DATA WHEN/"  /  /  "/
      B=DATE("MMDDYY")
      WHEN(1:2)=B(1:2)
      WHEN(4:5)=B(3:4)
      WHEN(7:8)=B(5:6)
      RETURN
      END
```

INSTRUCTIONS FOR RUNNING SEMDIF PROGRAMS.

1. STAT/SEMDIF

"This program can perform the computations for:
1. Means and standad deviations on subscales
2. Means and standard deviations on evaluation, potency, and activity factors (EPA scores)
3. Osgood D values for all concepts
4. The correlation between distance measures obtained from EPA scores and Osgood's D
5. Tests of significance of distances obtained with Osgood D.

Modifications: This program has been modified so that the means, the standard errors of the means, and the Osgood dees are saved in a file with the suffix "/OUTPUT". This file is for use with the programs: STAT/SEMDIF/CORR, STAT/SEMDIF/TTEST, and STAT/SEMDIF/PLOT."

To execute this program, type:
RUN $FREDLIB/STAT/SEMDIF

Enter the file name when requested by the program.

2. To get the Mann-Whitney or Wilcoxon results, two programs must be run as follows:

A. STAT/SEMDIF/OSGOOD

"This program computes the Osgood D (distance) values for each rated concept and other concepts on the subscales in preparation for use with the Wilcoxon or other rank statistical test."

To execute this program, type:
RUN $FREDLIB/STAT/SEMDIF/OSGOOD

Enter the name of the input file when requested by the program. This program creates the input file for the next program. It is stored under the name, filename/SD where filename is the name that was entered as the name of the input file.(ex. AA/MALE/SD)

If you want to compare 2 files for either the Mann-Whitney or the Wilcoxon tests, you must first run the original files through $FREDLIB/STAT/SEMDIF/OSGOOD to create an SD file for each of the original files.

B. STAT/SEMDIF/SIGTEST

"This program takes data input from program STAT/SEMDIF/OSGOOD and makes a series of comparisons. This program tests the

significance of the distance (converted into ranks) as a two-tailed test between one concept and two other concepts. Thus where 101=concept GOOD and 102=concept BAD we are interested in learning whether another concept such as self=07 is significantly closer to the concept GOOD or to BAD.

The Ident numbers of the control concepts are the last 6 numbers of the concept list (i.e. if there are 106 concepts, the control or reference concepts are 101-106). The program will handle up to 109 concepts including the last six. The names of the control concepts are contained in the first 3 records. Additional control or reference concepts can be listed by moving additional concepts to the last 6 positions.

The program is also set up to use two input files and compare the distance between one concept and the control concept in the first file and the identical concept and control concept in the second file. Initially it processes the first file as described above and then does the same with the second file. Then it takes from each file the distances between each concept and the positive controls (i.e. GOOD,STRONG,ACTIVE) and compares like concepts as described above. Finally it does the same for the distances between the concepts and the negative controls (i.e. BAD, WEAK, PASSIVE).

UPDATE (1982): This program has been modified to now include 3 tests from which the user can choose, the Wilcoxon test, the Mann Whitney test and the Spearman rank correlation for the individual against the control concepts. These tests can also be performed with two (2) input files.
If the user has two (2) input files and chooses either the Wilcoxon or Mann-Whitney tests, the program will ask if the user would like the files to be also ranked. This is asked after the original test is completed.

If the user requests the Mann-Whitney test, the Wilcox test is first done on the separate files. If the Spearman rank is requested, after it is done the user has the choice of ending the program, of performing wither the Wilcoxon or the Mann-Whitney tests.

The Wilcoxon (matched pairs procedure) is used with one group when differneces are tested between a concept and 2 refernece points such as GOOD and BAD. The Wilcoxon can be used with two groups, for example a pre- and post test also. Here testing would be to see whether a concept was closer to GOOD on one testing or the other. The Mann-Whitney is used with independent groups.

The Spearman correlation determines the degree of association between two sets of Dee measurements, as GOOD vs. concept in one group vs. a second.

UPDATE (1983): Another column has been added to the printout in

order to show P on a scale from 0 to 100. This scale is a somewhat modified percentile score. The probability levels are shown in the output under "P" and the scaled values under "SC". This scale is used only with the Wilcoxon matched pairs test."

To execute this program, type:
RUN $FREDLIB/STAT/SEMDIF/SIGTEST

Input the following information requested by the program.

NUMBER OF CONCEPTS - the total number of concepts for the current study including the control concepts.

NUMBER OF CASES IN EACH FILE - Both files must have the same number of cases. Example if both cases have 35 cases, enter 35

CONVERGION FACTOR - the corresponding factor to the number of cases entered as given in the table.

NUMBER OF INPUT FILES - '1' if you are only comparing a single set of concepts against the control concepts.

'2' if you are comparing 2 sets of of concepts against first the positive control concepts and then the negative concepts.

NAME(S) OF INPUT FILE(S) - the filename/SD files created by the OSGOOD program which was run first. If number of files was 1, only one file name will be requested. If number of files was 2, 2 file names will be requested, one at a time.

3. To get the Spearman rank correlations run the two programs as follows:

A. Run $FREDLIB/STAT/SEMDIF/OSGOOD as stated above.

B. Run $FREDLIB/STAT/SEMDIF/SIGTEST

When you are asked: "Enter 1 if matched pairs,etc.", enter 3. The program will come back with the same file names. When the correlations are calculated, the program will then ask you to enter 1 if you wish to do the Wilcox test on the same files, 2 if you wish to run the Mann-Whitney test on the same files , or 3 if you wish to end the program.

4. STAT/SEMDIF/CORR

"This program inputs two files each containing the Osgood D values between each and every concept. These distances were calculated by the program, STAT/SEMDIF. STAT/SEMDIF outputs this data in a printed report and stores it in a disk file. The name of this file is the name of the input file with a suffix of "/OUTPUT" added. This is the file to be used as the input file for STAT/SEMDIF/CORR.

STAT/SEMDIF/CORR computes the Spearman Rank Correlation for the two (2) groups of data. It can handle a distance matrix for up to 100 concepts inclusive. Each time the program finishes processing 2 files, it returns for 2 more files until the user has finished his/her job."

A. Run STAT/SEMDIF as described in section I above for each of the files which you want to correlate.

NOTE: The print code in the first line of the input file for STAT/SEMDIF must be 02 (not 01).

You will use the resulting /OUTPUT files as input files for STAT/SEMDIF/CORR.

B. Start the program by typing:
RUN $PRELLIB/STAT/SEMDIF/CORR

C. Enter the file names when asked for. Remember to use the /OUTPUT files!! (i.e. $DCLASS.SAMPLE/OUTPUT)

D. The program will ask you if you wish to use 2 more files. If you do, type 'Y'. If not, type 'N'.

5. STAT/SEMDIF/TTEST

"This program tests the significance of the differences between the means of two populations by means of a t-test. The input data is stored in 2 files, created by the program STAT/SEMDIF with the suffix "/OUTPUT"."

A. This program is run with the same files as STAT/SEMDIF/CORR. You must first run STAT/SEMDIF as above to get the /OUTPUT files.
NOTE: The print code in the first line of the input file for STAT/SEMDIF must be 02 (not 01).

B. Start the program by typing:
RUN $PRELLIB/STAT/SEMDIF/TTEST

The program will ask you to enter the names of the files. Remember to use the /OUTPUT files that you got from running

STAT/SEMDIF (i.e. PDCLASSJAMILY/OUTPUT).

6. STAT/SEMDIF/PLOT

"This program plots the E,P,A values for the means doubled as calculated by the program STAT/SEMDIF. The input file for this program is formed by running STAT/SEMDIF and using the resulting file with a suffix of "/OUTPUT".
When plotting a set of E,P,A values, the "P" value is plotted on the horizontal axis, the "E" value is plotted on the vertical axis, and the point itself is coded alphabetically according to what the "A" value is (the table of values for the "A" is produced below the graph itself).

The graph is scaled as follows:
Horizontal Axis - each "-" is 1/6 of 1
Vertical Axis - each "!" is 1/5 of 1
Each axis is scaled from "1" to "14"

When the graph is sent to the printer, the actual E,P,A values along with the corresponding concept name are printed on the right-hand side of the axis."

A. This program is run with the same file as SEMDIF/CORR and SEMDIF/TTEST. However, you only use one (1) file per plot! You must first run $FREDLIB/STAT/SEMDIF to get the /OUTPUT file.

B. Now, RUN $FREDLIB/STAT/SEMDIF/PLOT and enter the file name when it is asked for.

7. STAT/SEMDIF/FORMS

"This program prints the Semantic Differential Scale forms for the SEMDIF programs."

A. To run the forms program, you must be in LAWSON2.
Type R STAT/SEMDIF/FORMS.
If you want over 10 copies of the forms, go to step 4 below.

B. The program will first ask for the number of copies to be made.
Then the number of concepts is asked.
Next a title for the printouts is entered. The title must be less than 25 characters long.

C. The last questions deal with how the concepts are entered.

The program will first ask if you want to have the concepts entered from a file or from the terminal. Enter 1 if you want to use a file. Each record in the file must have the first 4 spaces as the concept number and then the concept must be less than 15 characters long. The concepts must be numbered sequentially starting with concept 1.

If you want to manually enter the concepts from the terminal, enter 2. The program will give you the concept numbers in sequential order and you just enter the concept names.

b. To print many forms, use the following procedure:

a. Create input file as described in step 3 above and save.

b. [illegible]

```
 2000 BEGIN JOB SENDIF/SURVEY/FORMS;
 4000 CLASS = 4;
 6000 FETCH = "DON'T LABEL OUTPUT";
 8000 RUN OBJECT/STAT/SENDIF/FORMS;
 9000 FILE FILE4(KIND=DISK,MAXRECSIZE=14,TITLE=SCRATCH,
                 PROTECTION=TEMPORARY);
 9500 FILE FILE5(KIND=READER);
10000 DATA FILE5
12000 XXX                                  NUMBER OF COPIES
14000 YYY                                  NUMBER OF CONCEPTS
15000 ZZZ                                   TITLE FOR PRINTOUT
16000 1
17000 WWWWWWWWWWWWWWW                      NAME OF INPUT FILE
18000 END JOB;
```

Replace 'XXX' in line 12000 with the number of copies desired.
Replace 'YYY' in line 14000 with the number of concepts.
Replace the Z's in line 15000 with the title you want on the printout.
Replace the W's in line 17000 with the name of the file created in step 4.a. above.

c. Type: :START

d. Type: REMOVE