University of Houston

MUG FILE PROJECT REPORT NUMBER UHMUG-13

FORTRAN Subroutines for the Pattern Recognition Algorithm Designed to Find "Look-Alikes" in a Mug File

K. Sumney

This project was supported by Grant Number 76-NI-99-012 awarded by the Law Enforcement Assistance Administration, U. S. Department of Justice, under the Omnibus Crime Control and Safe Streets Act of 1968, as amended. Points of view or opinions stated in this document are those of the authors and do not necessarily represent the official position or policies of the U. S. Department of Justice.

MUG FILE PROJECT REPORTS

UHMUG-1 Summary report for a Research Project *"A Man-Computer System for Solution of the Mug File Problem"*.
B. T. Rhodes, K. R. Laughery, G. M. Batten, and J. D. Bargainer.

UHMUG-2 *An Analysis of Procedures for Generating Facial Images*
K. R. Laughery, G. C. Duval, and R. H. Fowler.

UHMUG-3 *Factors Affecting Facial Recognition*
K. R. Laughery and R. H. Fowler

UHMUG-4 *The Minolta Montage Synthesizer as a Facial Image Generating Device*
F. H. Duncan and K. R. Laughery

UHMUG-5 *An Analysis of Strategies in Remembering and Generating Faces*
G. C. Duval

UHMUG-6 Data Base No. 1 - *Sketches and Identi-Kit Composites*

UHMUG-7 Data Base No. 2 - *Transcripts of Artist/Technician and Witness Interaction*

UHMUG-8 Data Base No. 3 - *Adjective Descriptors Used in Generating Sketches and Identi-Kit Composites*

UHMUG-9 Data Base No. 4 - *Miscellaneous Data from Sketch and Identi-Kit Generation*

UHMUG-10 *Support Hardware for Image Analysis Techniques Applied to the Mug File Program*
J. D. Bargainer

UHMUG-11 *Forgery Application of a Pattern Recognition Algorithm for Facial Images*
B. T. Rhodes and K. Prasertchuang

UHMUG-12 *An Evaluation of the UHMFS Facial Image Pattern Recognition Algorithms*
B. T. Rhodes and C. R. Walters

UHMUG-13 *FORTRAN Subroutines for the Pattern Recognition Algorithm Designed to Find "Look-Alikes" in a Mug File*
K. Sumney

UHMUG-14 *A Computer Simulation of the Minolta Montage Synthesizer*
G. W. Batten and T. Wiederhold

UHMUG-15 *The UHMFS Computer Software*
G. W. Batten, A. Karachievala and H. H. Nguyen

UHMUG-16 *Miscellaneous Computer Software for the Mug File Project*
G. W. Batten

University of Houston

MUG FILE PROJECT REPORT NUMBER UHMUG-13

FORTRAN Subroutines for the Pattern Recognition Algorithm Designed to Find "Look-Alikes" in a Mug File

K. Sumney

This project was supported by Grant Number 76-NI-99-012 awarded by the Law Enforcement Assistance Administration, U. S. Department of Justice, under the Omnibus Crime Control and Safe Streets Act of 1968, as amended. Points of view or opinions stated in this document are those of the authors and do not necessarily represent the official position or policies of the U. S. Department of Justice.

MUG FILE PROJECT REPORTS

UHMUG-1 Summary report for a Research Project *"A Man-Computer System for Solution of the Mug File Problem"*.
B. T. Rhodes, K. R. Laughery, G. M. Batten, and J. D. Bargainer.

UHMUG-2 *An Analysis of Procedures for Generating Facial Images*
K. R. Laughery, G. C. Duval, and R. H. Fowler.

UHMUG-3 *Factors Affecting Facial Recognition*
K. R. Laughery and R. H. Fowler

UHMUG-4 *The Minolta Montage Synthesizer as a Facial Image Generating Device*
F. H. Duncan and K. R. Laughery

UHMUG-5 *An Analysis of Strategies in Remembering and Generating Faces*
G. C. Duval

UHMUG-6 Data Base No. 1 - *Sketches and Identi-Kit Composites*

UHMUG-7 Data Base No. 2 - *Transcripts of Artist/Technician and Witness Interaction*

UHMUG-8 Data Base No. 3 - *Adjective Descriptors Used in Generating Sketches and Identi-Kit Composites*

UHMUG-9 Data Base No. 4 - *Miscellaneous Data from Sketch and Identi-Kit Generation*

UHMUG-10 *Support Hardware for Image Analysis Techniques Applied to the Mug File Program*
J. D. Bargainer

UHMUG-11 *Forgery Application of a Pattern Recognition Algorithm for Facial Images*
B. T. Rhodes and K. Prasertchuang

UHMUG-12 *An Evaluation of the UHMFS Facial Image Pattern Recognition Algorithms*
B. T. Rhodes and C. R. Walters

UHMUG-13 *FORTRAN Subroutines for the Pattern Recognition Algorithm Designed to Find "Look-Alikes" in a Mug File*
K. Sumney

UHMUG-14 *A Computer Simulation of the Minolta Montage Synthesizer*
G. W. Batten and T. Wiederhold

UHMUG-15 *The UHMFS Computer Software*
G. W. Batten, A. Karachievala and H. H. Nguyen

UHMUG-16 *Miscellaneous Computer Software for the Mug File Project*
G. W. Batten

UHMUG-13

FORTRAN SUBROUTINES FOR THE PATTERN RECOGNITION ALGORITHM DESIGNED TO FIND "LOOK-ALIKES" IN A MUG FILE

Kimball Sumney
Electrical Engineering Department

University of Houston
Central Campus
Houston, Texas

This project was supported by Grant Number 76-NI-99-012 awarded by the Law Enforcement Assistance Administration, U. S. Department of Justice, under the Omnibus Crime Control and Safe Streets Act of 1968, as amended. Points of view or opinions stated in this document are those of the authors and do not necessarily represent the official position or policies of the U. S. Department of Justice.

ABSTRACT

The purpose of this report is to describe a set of subroutines which are a FORTRAN IV coding of the pattern recognition algorithm designed by Dr. James R. Townes of the Electrical Engineering Department at the University of Houston. These subroutines have been incorporated into the UHMFS system which combines the capabilities of this pattern recognition algorithm to find "look-alikes" with the sorting capabilities of the Oakland, Calif. Police Department's CRIME system. This documentation is provided for law enforcement agencies who already have a system similiar to Oakland's CRIME system and want to combine the pattern recognition capability with it.

The algorithm is subdivided into three sections. These sections are realized via FORTRAN subroutines, with a drive subroutine to control execution of the sections. The basic operation of the algorithm is discussed first, followed by a detailed examination of each available subroutine.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS AND TABLES

FOREWARD

This report was prepared by Kimball Sumney, a graduate student working with Jim Townes. It describes a set of FORTRAN subroutines which can be used to implement this version of the Townes's "look-alike" algorithm. The coefficients in the regression equations were determined by a jack-kniffing procedure using a data base of sketch artist drawings and Identi-kit composites generated earlier in this research program.

Readers who are interested in the total mug file program or details of the pattern recognition work will need to consult other reports. This report is for those who want to implement the algorithm on the computer.

Ben T. Rhodes, Jr.
Project Director
June, 1977

SUMMARY

The algorithm requires nine facial measurements as primary input. Figure 1 shows where the measurements are taken from the face. Each step of the algorithm performs a transformation on these measurements or ratios of these measurements. The measurements can be taken from a sketch or composite representation of a subject developed from a witness description; or from a photograph of the subject. The unit of measurement used in determining the distances is immaterial as long as the same unit is used for all nine measurements.

In the first step of the algorithm, the measurements, which are listed in Table 1, are paired to form eight ratios. Assuming that the measurements are ordered according to the list in Table 1, (the order of the measurements shown in Table 1 is the order assumed by all the various components of the algorithm.) Table 2 and Table 3 depict the two different sets of ratios that are used by the look-alike algorithm depending on whether the image supplied is a sketch or a composite. Ratios formed from photographic measurements are standardized by dividing by the respective standard deviation. Ratios formed from a sketch or composite representation are not standardized.

The second step of the algorithm modifies the ratios generated from a sketch or composite representation. Ratios formed from photographic measurements are not processed by this phase of the algorithm. The first operation is a sixth order linear regression on each ratio. Then, an eighth order multi-linear regression is used to further modify the ratios. The value of the regression

coefficients differs depending on whether a sketch or composite is used.

In the final step of the algorithm the Euclidean distance between the selected ratios of the image supplied and the mug shot is calculated and sorted in ascending order of similarity (shortest distance) between the subject and the mug file photographs.

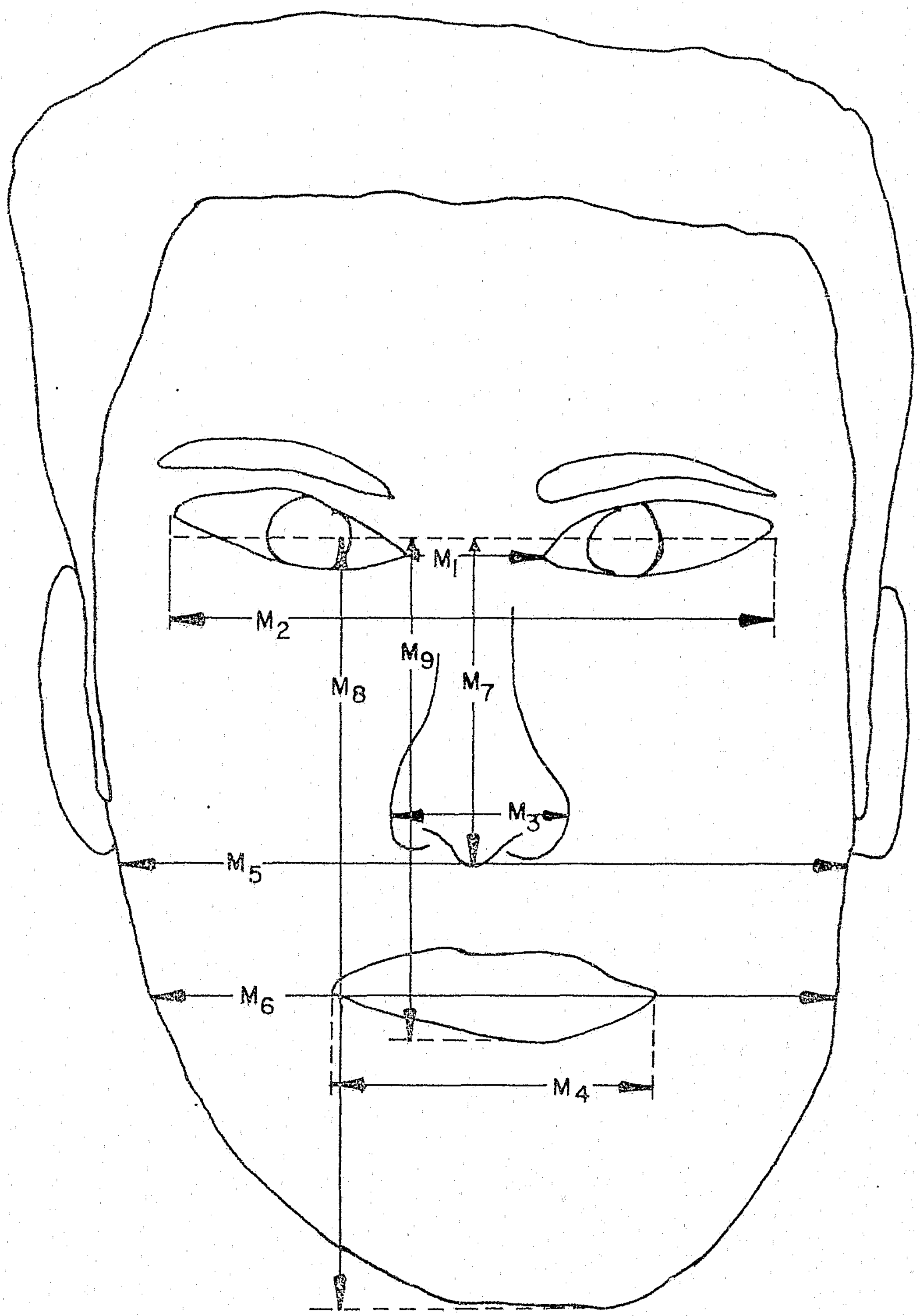


Figure 1

TABLE 1: MEASUREMENTS USED

Measurement Number	Definition of Measurement
1.	Internal Biocular Distance
2.	External Biocular Distance
3.	Nose Width
4.	Mouth Width
5.	Distance Across Face Measured Directly Under Nose
6.	Distance Across Face Measured Across Mouth
7.	Nose Length from Tip of Nose to Midline of Eyes
8.	Distance from Chin to Eyes
9.	Distance from Lower Lip to Eyes

TABLE 2: SKETCH RATIOS

Ratio Number	Measurements Used
1.	2/8
2.*	5/8
3.*	3/8
4.	5/2
5.	6/2
6.	6/8
7.	3/6
8.	9/3

*Not used when comparing two sets of ratios.

TABLE 3: COMPOSITE RATIOS

Ratio Number	Measurements Used
1.*	8/3
2.*	9/8
3.	5/3
4.	6/3
5.*	2/8
6.	2/9
7.*	3/1
8.	1/8

*Not used when comparing two sets of ratios.

DESCRIPTION OF SUBROUTINES

As previously mentioned, each step of the algorithm is realized by a FORTRAN subroutine. In this section, the available subroutines will be discussed in detail. The explanation of each subroutine is accompanied by a listing which includes extensive comment statements.

The first step of the algorithm is implemented by subroutine RATI1. This subroutine takes nine measurements as input and pairs them to form eight ratios according to the guides specified by the parameters called TYPE. Table 4 lists the allowed values of TYPE and the associated meaning. The first step of RATI1 is to check the assignments made to TYPE. If an invalid assignment is made, ERFLG is set to one (1) and control returns to the calling routine. The second step sets up dummy arrays to hold the guides and coefficients which are to be used. The ratios are then calculated using the dummy array for guidance. (Note: The denominator measurement is first checked to insure that an attempt to divide by zero (0) is not made.) Finally, if the measurements were from a photograph, the ratios are standardized and control is returned to the calling routine. If an error is detected by RATI1, ERFLG is set to one (1) and control immediately returns to the calling routine after an error message is printed. The results in the event of an error are not valid.

Subroutine RATI2 provides another means of implimenting the first of the algorithm. This subroutine uses two sets of nine measurements as input and returns two sets of eight ratios formed according to the value of TYPE. The operations performed by RATI2 are the same as the operations of RATI1, except that two sets of measurements and ratios are involved. The first step in RATI2 is to check the assignments made to TYPE. If an invalid assignment was made, ERFLG is set to one and control returns to the originating routine. The next step loads the dummy arrays used during processing with the proper coefficients. After checking the denominator measurement to insure a divide by zero is not attempted, the ratios are calculated according to the guides in the dummy arrays. If TYPE (2) is one or three, the ratios in RATO1 are standardized. If TYPE (2) equals two or three, the ratios in RATO2 are standardized. As in RATI1, the results produced by RATI2 are not valid in the event of an error which is indicated by ERFLG.

The second step of the algorithm is realized in ALGOR. The subroutine takes the ratios in RATIO (see Figure 6 and Figure 7 for the flowchart and listing of ALGOR) and modifies the ratios by a set of linear and multi-linear regression equations. The values taken on by the coefficients of the equations vary depending upon the mode of representation. Ratios produced from photographic measurements should not be modified by the regression equations. The first step executed by ALGOR is to check the values assigned to TYPE. If an invalid assignment was made (TYPE (2) equal one or three), ERFLG is set to one and control is returned to the originating routine. Next, the ratios are checked to insure that they are neither equal to zero nor negative. The third step replaces the ratios by a new value calculated from the sixth order linear regression. ($R_{i(new)} = L_{1i} + L_{2i} \cdot R_{i(old)} + L_{3i} \cdot R^2_{i(old)} \cdots + L_{7i} \cdot R^6_{i(old)}$, where L_{ji} are coefficients of the

TABLE 4: TYPE

Location	Value Assigned	Definition
Type (1)	Ø	Use Sketch Guides And Coefficients
	1	Use Composite Guides And Coefficients
Type (2)	Ø	Neither Set of Ratios Should be Standardized
	1	The First Set of Ratios (Ratio 1) Should Be Standardized
	2	The Second Set of Ratios (Ratio 2) Should be Standardized
	3	Both Sets of Ratios (Ratio 1, Ratio 2) Should Be Standardized

TABLE 5: CG (CONTROL GROUP)

Location	Value Assigned	Definition
CG (1)	Ø	Do Not Use RATI1
	1	Do Use RATI1
CG (2)	Ø	Do Not Use RATI2
	1	Do Use RATI2
CG (3)	Ø	Do Not Use ALGOR
	1	Do Use ALGOR
CG (4)	Ø	Do Not Use COMP
	1	Do Use COMP

regression equation and i=1,2,...,8.) The final step is to further modify the ratios by an eighth order multi-linear regression. ($R_{i(final)}=M_{i1}\cdot R_{1(new)}+M_{i2}\cdot R_{2(new)}+\ldots+M_{i8}\cdot R_{8(new)}$ where M_{ij} are the coefficients of the multi-linear regression and i=1,2,...,8). The results in the event of an error are invalid as in RATI1 and RATI2.

The final step of the algorithm, comparison of two sets of ratios, is performed by COMP. COMP takes the ratios in RATO1 and RATO2 and computes the Euclidean distance between specified ratios according to the type of representation. As in the other subroutines, COMP first checks TYPE for valid assignments. If an error exists, ERFLG is set to one and control returns to the calling routine. The next action is to load the proper guides for processing into the dummy array, then the squared Euclidean distance is calculated from the selected ratios. Finally, the Euclidean distance is computed and returned to the calling routine in ECLUD. As before, the results in the event of an error are invalid.

To facilitate usage of the subroutines, a "driver" subroutine is supplied with a control logic such that the user can use any of the routines with a single call statement. The subroutine, ALRIT, uses values assigned to CG as control flags indicating whether or not a certain subroutine is to be used. Table 5 shows the allowable values which may be taken by CG and the action controlled by the variables in CG. The first action taken by ALRIT is to check TYPE and CG for valid assignments. If an error is found, ERFLG is set to one and control returns to the calling routine. Next, CG (1) is checked to see if RATI1 should be used. ERFLG is then checked to see if an erreo returned should be made. The next step checks CG (2) to see if RATI2 should be executed, followed by a check of errors. Step four checks CG (3) to see if ALGOR should be executed and also checks for the occurence of an error. Finally, step five will compute the Euclidean distance between the ratios in RATO1 and RATO2 if CG (4) so indicates. ALRITH does not generate any written output and is primarily used to free the user from having to remember the calling sequences for each subroutine. Since the amount of storage required by ALRITH is small, its use is highly recommended.

LISTINGS OF SUBROUTINES

```
** RAT1 **                          FILE: 0476Q$Q$Q$*DUMP              TIME: 11:24:18

       SUBROUTINE RATI1(TYPE,MEAS,RATIO,ERFLG)
C
C     PURPOSE:  RATI1 CALCULATES EIGHT RATIOS FROM NINE
C               MEASUREMENTS ACCORDING TO THE SPECIFIED
C               REPRESENTATION.
C
C     FORTRAN CALLING SEQUENCE:
C
C              CALL RATI1(TYPE,MEAS,RATIO,ERFLG)
C
C     EXTERNAL VARIABLES:
C
C               TYPE IS INTEGER ARRAY DIMENSIONED TWO (2).  TYPE
C                    SPECIFIES WHICH COEFFICIENTS ARE TO BE USED
C                    AND THE REPRESENTATION FROM WHICH THE
C                    MEASUREMENTS WERE TAKEN.
C
C               MEAS IS AN INTEGER ARRAY DIMENSIONED NINE (9).
C                    MEAS1 CONTAINS A SET OF MEASUREMENTS.
C
C              RATIO IS A REAL ARRAY DIMENSIONED EIGHT (8).
C                    RATO1 CONTAINS RATIOS FORMED FROM THE
C                    MEASUREMENTS IN MEAS1.
C
C              ERFLG IS AN INTEGER VARIABLE.ERFLG RETURNS AN
C                    ERROR INDICATOR.
C
C     INTERNAL VARIABLES:
C
C                 IM IS AN INTEGER ARRAY DIMENSIONED NINE(9).
C                    IM HOLDS GUIDES DURING PROCESSING.
C
C              IMAGS IS AN INTEGER ARRAY DIMENSIONED TWENTY-FOUR (24).
C                    IMAGS HOLDS SKETCH GUIDES.
C
C              IMAGC IS AN INTEGER ARRAY DIMENSIONED TWENTY-FOUR (24).
C                    IMAGC HOLDS THE COMPOSITE GUIDES.
C
C                 SD IS A REAL ARRAY DIMENSIONED EIGHT (8).
C                    SD HOLDS THE STANDARD DEVIATION
C                    COEFFICIENTS.
C
C                SDS IS A REAL ARRAY DIMENSIONED EIGHT (8).
C                    SDS HOLDS THE SKETCH COEFFICIENTS.
C
C                SDC IS A REAL ARRAY DIMENSIONED EIGHT (8).
C                    SDC HOLD THE COMPOSIT COEFICIENTS.
C
C                  I IS AN INTEGER LOOP COUNTER.
C
C              NUMMS IS AN INTEGER VARIABLE USED TO
C                    LOCATE THE NUMERATOR MEASUREMENT.
C
C              DENMS IS AN INTEGER VARIABLE USED TO
C                    LOCATE THE DENOMINATOR MEASUREMENT.
C
C     CALLED BY:  ALRITH
```

```
      C
      C     SUBROUTINE CALLED:  NONE
      C
            INTEGER TYPE,ERFLG,DENMS
            DIMENSION TYPE(2),MEAS(9),RATIO(8),IM(24),IMAGS(24),IMAGC(24),
           *          SD(8),SDS(8),SDC(8)
      C
      C       SKETCH GUIDES
      C
            DATA IMAGS/ 2, 5, 3, 5, 6, 6, 3, 9, 8, 8, 8, 2, 2, 8, 6, 3,
           *            1, 0, 0, 1, 1, 1, 1, 1/
      C
      C       SKETCH STANDARD DEVIATION TERMS        (1/S.D.)
      C
            DATA SDS/ 18.7641     , 14.2058     , 29.4947     , 9.17068     ,
           *          8.52845     , 13.1528     , 30.9854     , 4.01649     /
      C
      C       COMPOSITE GUIDES
      C
            DATA IMAGC/ 8, 9, 5, 6, 2, 2, 3, 1, 3, 8, 3, 3, 8, 9, 1, 8,
           *            0, 0, 1, 1, 0, 1, 0, 1/
      C
      C       COMPOSITE STANDARD DEVIATION TERMS        (1/S.D.)
      C
            DATA SDC/ 2.91713     , 27.1110     , 2.89199     , 2.89263     ,
           *          18.7618     , 11.5399     , 7.88930     , 35.8410     /
      C
      C       CHECK DATA TYPE
      C
      C
      C       ERROR ENTRY?
      C
            IF (TYPE(2).LT.0.OR.TYPE(2).GT.3) GO TO 12
            IF (TYPE(1).EQ.1) GO TO 4
            IF (TYPE(1).NE.0) GO TO 12
      C
      C      ALL CLEAR, SKETCH DATA
      C
            DO 2 I=1,8
               IM(I)=IMAGS(I)
               IM(I+8)=IMAGS(I+8)
               IM(I+16)=IMAGS(I+16)
               SD(I)=SDS(I)
          2 CONTINUE
      C
      C       SKETCH COEFFICIENTS READY, START PROCESSING
      C
            GO TO 8
      C
      C       COMPOSITE DATA
      C
          4 DO 6 I=1,8
               IM(I)=IMAGC(I)
               IM(I+8)=IMAGC(I+8)
               IM(I+16)=IMAGC(I+16)
               SD(I)=SDC(I)
          6 CONTINUE
      C
```

```
C       COMPOSITE COEFFICIENTS READY, START PROCESSING.
C
      8 DO 10 I=1,8
C
C       CALCULATE RATIOS TO BE USED BY THE ALGORITHM.
C
         NUMMS=IM(I)
         DENMS=IM(I+8)
C
C       CHECK FOR DIVIDE BY ZERO
C
         IF (MEAS(DENMS).EQ.0) GO TO 12
         RATIO(I)=FLOAT(MEAS(NUMMS))/FLOAT(MEAS(DENMS))
     10 CONTINUE
        ERFLG=0
C
C       MEASUREMENTS FROM A PICTURE?
C
        IF (TYPE(2).EO.0.OR.TYPE(2).EQ.2) RETURN
C
C       PICTURE MEASUREMENTS          STANDARDIZE
C
        DO 11 I=1,8
         RATIO(I)=RATIO(I)*SD(I)
     11 CONTINUE
C          FINISHED, RETURN
        RETURN
C
C       ERROR SECTION:  BAD DATA OR ERROR ENTRY.
C
     12 ERFLG=1
        WRITE(6,100)
    100 FORMAT(' ERROR IN RATI1, WRONG TYPE OR DIVID BY ZERO')
        RETURN
        END
```

```
      SUBROUTINE RATI2(TYPE,MEAS1,MEAS2,RATO1,RATO2,ERFLG)
C
C     PURPOSE:  RATI2 CALCULATES TWO SETS OF EIGHT RATIOS
C               FROM TWO SETS OF NINE MEASUREMENTS.
C
C     FORTRAN CALLING SEQUENCE:
C
C         CALL RATI2(TYPE,MEAS1,MEAS2,RATO1,RATO2,ERFLG)
C
C     EXTERNAL VARIABLES:
C
C              TYPE IS INTEGER ARRAY DIMENSIONED TWO (2).  TYPE
C                     SPECIFIES WHICH COEFFICIENTS ARE TO BE USED
C                     AND THE REPRESENTATION FROM WHICH THE
C                     MEASUREMENTS WERE TAKEN.
C
C             MEAS1 IS AN INTEGER ARRAY DIMENSIONED NINE (9).
C                     MEAS1 CONTAINS A SET OF MEASUREMENTS.
C
C             MEAS2 IS AN INTEGER ARRAY DIMENSIONED NINE (9).
C                     MEAS2 CONTAINS A SET OF MEASUREMENTS.
C
C             RATO1 IS A REAL ARRAY DIMENSIONED EIGHT (8).
C                     RATO1 CONTAINS RATIOS FORMED FROM THE
C                     MEASUREMENTS IN MEAS1.
C
C             RATO2 IS A REAL ARRAY DIMENSIONED EIGHT (8).
C                     RATO2 CONTAINS RATIOS FORMED FROM THE
C                     MEASUREMENTS IN MEAS2.
C
C             ERFLG IS AN INTEGER VARIABLE.  ERFLG RETURNS AN
C                     ERROR INDICATOR.
C
C     INTERNAL VARIABLES:
C
C                IM IS AN INTEGER ARRAY DIMENSIONED TWENTY-FOUR (24).
C                     IM HOLDS GUIDES DURING PROCESSING.
C
C             IMAGS IS AN INTEGER ARRAY DIMENSIONED TWENTY-FOUR (24).
C                     IMAGS HOLDS SKETCH GUIDES.
C
C             IMAGC IS AN INTEGER ARRAY DIMENSIONED TWENTY-FOUR (24).
C                     IMAGC HOLDS THE COMPOSITE GUIDES.
C
C                SD IS A REAL ARRAY DIMENSIONED EIGHT (8).
C                     SD HOLDS THE STANDARD DEVIATION
C                     COEFFICIENTS.
C
C               SDS IS A REAL ARRAY DIMENSIONED EIGHT (8).
C                     SDS HOLDS THE SKETCH COEFFICIENTS.
C
C               SDC IS A REAL ARRAY DIMENSIONED EIGHT (8).
C                     SDC HOLD THE COMPOSIT COEFICIENTS.
C
C                 I IS AN INTEGER LOOP COUNTER.
C
C             NUMMS IS AN INTEGER VARIABLE USED TO
```

```
      C                  LOCATE THE NUMERATOR MEASUREMENT.
      C
      C           DENMS IS AN INTEGER VARIABLE USED TO
      C                  LOCATE THE DENOMINATOR MEASUREMENT.
      C
      C    CALLED BY:  ALRITH
      C
      C    SUBROUTINE CALLED:  NONE
      C
      C
      C.
            INTEGER TYPE,ERFLG,DENMS
            DIMENSION MEAS1(9),MEAS2(9),RATO2(8),RATO1(8),
           *         IM(24),IMAGC(24),IMAGS(24),SD(8),SDS(8),SDC(8),TYPE(2)
      C
      C       SKETCH GUIDES
      C
            DATA IMAGS/ 2, 5, 3, 5, 6, 6, 3, 9, 8, 8, 8, 2, 2, 8, 6, 3,
           *          1, 0, 0, 1, 1, 1, 1, 1/
      C
      C       SKETCH STANDARD DEVIATION TERMS        (1/S.D.)
      C
            DATA SDS/ 18.7641     , 14.2058     , 29.4947     , 9.17068
           *         8.52845     , 13.1528     , 30.9854     , 4.01649
      C
      C       COMPOSITE GUIDES
      C
            DATA IMAGC/ 8, 9, 5, 6, 2, 2, 3, 1, 3, 8, 3, 3, 8, 9, 1, 8,
           *          0, 0, 1, 1, 0, 1, 0, 1/
      C
      C       COMPOSITE STANDARD DEVIATION TERMS        (1/S.D.)
      C
            DATA SDC/ 2.91713     , 27.1110     , 2.89199     , 2.89263
           *         18.7618     , 11.5399     , 7.88930     , 35.8410
      C
      C       CHECK DATA TYPE
      C
      C
      C       ERROR ENTRY?
      C
            IF (TYPE(2).LT.0.OR.TYPE(2).GT.3) GO TO 18
            IF (TYPE(1).EQ.1) GO TO 4
            IF (TYPE(1).NE.0) GO TO 18
      C
      C       ALL CLEAR, SKETCH DATA
      C
            DO 2 I=1,8
               IM(I)=IMAGS(I)
               IM(I+8)=IMAGS(I+8)
               IM(I+16)=IMAGS(I+16)
               SD(I)=SDS(I)
          2 CONTINUE
      C
      C       SKETCH COEFFICIENTS READY, START PROCESSING
      C
            GO TO 8
      C
      C       COMPOSITE DATA
      C
```

```
    4 DO 6 I=1,8
         IM(I)=IMAGC(I)
         IM(I+8)=IMAGC(I+8)
         IM(I+16)=IMAGC(I+16)
         SD(I)=SDC(I)
    6 CONTINUE
C
C       COMPOSITE COEFFICIENTS READY, START PROCESSING.
C
    8 DO 10 I=1,8
C
C       CALCULATE RATIOS TO BE USED BY THE ALGORITHM.
C
         NUMMS=IM(I)
         DENMS=IM(I+8)
C
C       CHECK FOR DIVIDE BY ZERO
C
         IF (MEAS1(DENMS).EQ.0.OR.MEAS2(DENMS).EQ.0) GO TO 18
         RATO1(I)=FLOAT(MEAS1(NUMMS))/FLOAT(MEAS1(DENMS))
         RATO2(I)=FLOAT(MEAS2(NUMMS))/FLOAT(MEAS2(DENMS))
   10 CONTINUE
      ERFLG=0
C
C       MEASUREMENTS FROM A PICTURE?
C
      IF (TYPE(2).EQ.0) RETURN
      IF (TYPE(2).EQ.2) GO TO 14
C
C       PICTURE MEASUREMENTS            STANDARDIZE
C
      DO 12 I=1,8
         RATO1(I)=RATO1(I)*SD(I)
   12 CONTINUE
C
C       MEASUREMENTS FROM A PICTURE?
C
      IF (TYPE(2).NE.3)RETURN
C
C       PICTURE MEASUREMENTS            STANDARDIZE
C
   14 DO 16 I=1,8
         RATO2(I)=RATO2(I)*SD(I)
   16 CONTINUE
      RETURN
C
C       ERROR SECTION:  BAD DATA OR ERROR ENTRY.
C
   18 WRITE(6,100)
  100 FORMAT(' ERROR IN RATI2, ERROR ENTRY OR DIVIDE BY ZERO')
      END
```

```
      SUBROUTINE ALGOR(TYPE,RATIO,ERFLG)
C
C    PURPOSE:  ALGOR MODIFIES THE EIGHT RATIOS WITH
C              A LINEAR AND MULTI-LINEAR REGRESSION.
C
C    FORTRAN CALLING SEQUENCE:
C
C            CALL ALGOR(TYPE,RATIO,ERFLG)
C
C    EXTERNAL VARIABLES:
C
C             TYPE IS INTEGER ARRAY DIMENSIONED TWO (2).  TYPE
C                    SPECIFIES WHICH COEFFICIENTS ARE TO BE USED
C                    AND THE REPRESENTATION FROM WHICH THE
C                    MEASUREMENTS WERE TAKEN.
C
C            RATIO IS A REAL ARRAY DIMENSIONED EIGHT (8).
C                    RATO1 CONTAINS RATIOS FORMED FROM THE
C                    MEASUREMENTS IN MEAS1.
C
C            ERFLG IS AN INTEGER VARIABLE.  ERFLG RETURNS AN
C                    ERROR INDICATOR.
C
C    INTERNAL VARIABLES:
C
C               RL IS A REAL MATRIX.  RL HOLDS THE
C                    LINEAR REGRESSION COEFFICIENTS DURING
C                    PROCESSING.
C
C              RLS IS A REAL MATRIX.  RLS HOLDS THE
C                    SKETCH LINEAR REGRESSON COEFFICIENTS.
C
C              RLC IS A REAL MATRIX.  RLC HOLDS THE
C                    COMPOSITE LINEAR REGRESSION COEFFICIENTS.
C
C               RM IS A REAL MATRIX.  RM HOLDS THE MULTI-
C                    LINEAR COEFFICIENTS DURING PROCESSING.
C
C              RMS IS A REAL MATRIX.  RMS HOLDS THE
C                    SKETCH MULTI-LINEAR REGRESSION COEFFICIENTS.
C
C              RMC IS A REAL MATRIX.  RMC HOLDS THE
C                    COMPOSITE MULTI-LINEAR REGRESSION COEFFICIENTS.
C
C                I IS AN INTEGER LOOP COUNTER.
C
C                J IS AN INTEGER LOOP COUNTER.
C
C               XS IS A REAL VARIABLE USED AS A SCRATCH PAD.
C
C                P IS A REAL VARIABLE USED AS AN ACCUMULATOR.
C
C               PP IS A REAL VARIABLE USED AS A SCRATCH PAD.
C
C    CALLED BY:  ALRITH
C
C    SUBROUTINE CALLED:  NONE
```

```
C
      INTEGER TYPE,ERFLG
      DIMENSION TYPE(2),RATIO(8),RLS(8,7),RLC(8,7),RL(8,7),
     *          RMC(8,8),RMS(8,8),RM(8,8)
C
C       COMPOSITE LINEAR REGRESSION COEFFICIENTS.
C
      DATA RLC/-20.1839    ,-50.8731     , 10.9042     ,-7.81214    ,
     *          47.1197    , 58.8774     , 42.0195     , 16.3542    ,
     *          18.0872    , 71.3119     , .526522     , 28.0595    ,
     *         -125.505    ,-121.663     ,-119.581     ,-34.9403    ,
     *          1.89041    , 244.481     ,-.570597     ,-15.0083    ,
     *          91.6108    , 55.3313     , 146.144     ,-90.1592    ,
     *         -2.40978    ,-63.6567     , .173184E-04, 2.50166    ,
     *          1.13657    , 47.4550     ,-63.8376     , 535.887    ,
     *         -.163229    ,-390.584     ,-.126802E-02, .238376    ,
     *          298.352    , 7.09083     ,-5.95198     ,-899.596    ,
     *          .235107    ,-391.128     , .227879E-01,-.908174E-01,
     *         -542.203    ,-55.5274     , 12.5733     , 6321.08    ,
     *         -.285519E-01, 685.110     ,-.387168E-02, .437872E-02,
     *          244.313    , 20.9255     ,-2.78424     ,-13560.7    /
C
C       COMPOSITE MULTI-LINEAR REGRESSION COEFFICIENTS.
C
      DATA RMC/ .529198    ,-.355838     , .319023     , .558774    ,
     *          .588605E-01, .512307     , .755500E-01,-.894074E-01,
     *          .353725    , 1.08002     , .495551     , .321378    ,
     *          .519868    ,-.406399E-01,-.167401     , .310677    ,
     *          .934724    , .530598     , .650117     ,-.156314    ,
     *         -.437643    ,-.712955     ,-.573118     ,-.327275    ,
     *         -.334122    ,-.426188     , .166093     , .734818    ,
     *         -.438503    ,-.158132     , .592683     ,-.282872    ,
     *         -1.15464    ,-.220241     ,-1.47354     ,-1.23729    ,
     *         -.121136    ,-.112166     , .668840     , .235158    ,
     *          .428591    , .174737     , .527447     , .411820    ,
     *          .968940    , 1.22731     , .421269     , .150658    ,
     *         -.243225    ,-.826593E-02,-.667482E-01,-.143687    ,
     *          .232987    , .461196     , .847655     ,-.156776    ,
     *          .411564    , .122960     , .465913     , .636512    ,
     *          .104718E-01,-.373879E-01,-1.07833     , .752621    /
C
C       SKETCH LINEAR REGRESSION COEFFICIENTS.
C
      DATA RLS/ 96.8582    , 20.9630     , 17.3854     , 19.9587    ,
     *          9.96903    , 100.049     , 7.89575     , 3.93403    ,
     *         -261.618    ,-11.4405     ,-43.6896     ,-.333246    ,
     *          4.81844    ,-362.294     , 26.7107     ,-2.39796    ,
     *          237.626    , 14.7046     ,-16.1286     , 11.2972    ,
     *         -4.05233    , 487.193     ,-196.483     , 3.09746    ,
     *         -37.9147    ,-19.6582     , 133.449     ,-43.5140    ,
     *         -2.75247    ,-153.555     , 612.559     , .562615    ,
     *         -114.[illegible]23    , 3.94350     , 759.514     , 27.2013    ,
     *          3.91200    ,-138.916     ,-796.034     ,-.382430    ,
     *          263.503    , 14.1881     ,-1658.64     , 2.92929    ,
     *         -.904459    , 78.7569     , 230.145     ,-.121914    ,
     *         -171.731    ,-7.00229     , 542.341     ,-3.83308    ,
     *          .956842E-01, 2.45484     , 432.705     , .394466E-01/
C
```

```
C       SKETCH MULTI-LINEAR REGRESSION COEFFICIENTS.
C
        DATA RMS/ .545320     ,-.769104     , .860350E-01,-.868150     ,
       *         -.630034     ,-.531921     , .589636     , .205858     ,
       *          .505103     , 2.04013     , .293708     , .996755     ,
       *          .849775     , 1.75392     ,-.979727     ,-.453718     ,
       *          .677834     , .762433     , .682690     ,-.148673E-01,
       *         -.116812     , .527229     , .318647     ,-.297295     ,
       *         -.600156     ,-.854297     ,-.918802     ,-.297629E-01,
       *          .370835E-02,-.702658     ,-.403589     , .839028     ,
       *          .383501     , 1.14285     , .805981     , .715550     ,
       *          .768625     , 1.17392     , .106209     ,-.539430     ,
       *         -.467021     ,-1.05596     ,-.107177     ,-.773427E-01,
       *         -.153898     ,-1.01945     , .795217     , .227306     ,
       *         -.144675     ,-.282420     , .212254     , .203583     ,
       *          .103255     ,-.331128     , .659594     , .134035     ,
       *          .419073     , .311782     ,-.393779E-01, .166628     ,
       *          .143000E-01, .120152     , .130118     , .689257     /
C
C       ERROR CHECKING:  BAD DATA?
C
        DO 1 I=1,8
           IF (RATIO(I).LE.0.00) GO TO 2
      1 CONTINUE
        GO TO 3
C
C       MISSING OR NEGATIVE DATA
C
      2 WRITE(6,101)
    101 FORMAT(' BAD INPUT DATA FOR ALGOR ')
        ERFLG=1
        RETURN
C
C       ERROR CHECKING:  WRONG TYPE?
C
      3 IF (TYPE(2).EQ.1.OR.TYPE(2).EQ.3) GO TO 4
        IF (TYPE(1).EQ.0) GO TO 6
        IF (TYPE(1).EQ.1) GO TO 12
C
C       ERROR ENTRY
C
      4 WRITE(6,100)
    100 FORMAT(' ERROR ENTRY:  WRONG TYPE ',I10)
        ERFLG=1
        RETURN
C
C       INITIALIZE COEFFICIENTS
C
C
C       SKETCH DATA
C
      6 DO 10 I=1,8
           DO 8 J=1,7
              RL(I,J)=RLS(I,J)
              RM(I,J)=RMS(I,J)
      8    CONTINUE
           RM(I,8)=RMS(I,8)
     10 CONTINUE
```

```
C
C       SKETCH COEFFICIENTS READY, START PROCESSING
C
      GO TO 18
C
C       COMPOSITE DATA
C
   12 DO 16 I=1,8
        DO 14 J=1,7
          RL(I,J)=RLC(I,J)
          RM(I,J)=RMC(I,J)
   14   CONTINUE
        RM(I,8)=RMC(I,8)
   16 CONTINUE
C
C       COMPOSITE COEFFICIENTS READY, START PROCESSING
C
C
C       LINEAR REGRESSION
C
   18 DO 25 I=1,8
        XS=RL(I,1)
        P=RATIO(I)
        PP=P
        DO 22 J=2,7
          XS=XS+RL(I,J)*PP
          PP=PP*P
   22   CONTINUE
        RATIO(I)=XS
   25 CONTINUE
C
C       MULTI-LINEAR REGRESSION
C
      DO 35 I=1,8
        P=RM(I,1)*RATIO(1)
        DO 30 J=2,8
          P=P+RM(I,J)*RATIO(J)
   30   CONTINUE
        RATIO(I)=P
   35 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE COMP(TYPE,RATO1,RATO2,ECLUD,ERFLG)
C
C
C    PURPOSE:  COMP COMPARES TWO SETS OF EIGHT (8) RATIOS
C              IN RATO1 AND RATO2 BY COMPUTING THE
C              ECLUDIAN DISTANCE BETWEEN THE RATIOS.
C
C    FORTRAN CALLING SEQUENCE:
C
C            CALL COMP(TYPE,RATO1,RATO2,ECLUD,ERFLG)
C
C    EXTERNAL VARIABLES:
C
C             TYPE IS INTEGER ARRAY DIMENSIONED TWO (2).  TYPE
C                    SPECIFIES WHICH COEFFICIENTS ARE TO BE USED
C                    AND THE REPRESENTATION FROM WHICH THE
C                    MEASUREMENTS WERE TAKEN.
C
C            RATO1 IS A REAL ARRAY DIMENSIONED EIGHT (8).
C                    RATO1 CONTAINS RATIOS FORMED FROM THE
C                    MEASUREMENTS IN MEAS1.
C
C            RATO2 IS A REAL ARRAY DIMENSIONED EIGHT (8).
C                    RATO2 CONTAINS RATIOS FORMED FROM THE
C                    MEASUREMENTAS IN MEAS2.
C
C            ECLUD IS A REAL VARIABLE.  ECLUD RETURNS THE
C                    ECLUDIAND DISTANCE BETWEEN RATO1 AND RATO2.
C
C            ERFLG IS AN INTEGER VARIABLE.ERFLG RETURNS AN
C                    ERROR INDICATOR.
C
C    INTERNAL VARIABLES:
C
C               IM IS AN INTEGER ARRAY. IM HOLDS GUIDES DURING
C                    PROCESSING OF THE RATIOS.
C
C            IMAGS IS AN INTEGER ARRAY.  IMAGS HOLDS THE
C                    SKETCH GUIDES.
C
C            IMAGC IS AN INTEGER ARRAY.  IMAGC HOLD THE
C                    COMPOSITE GUIDES.
C
C                I IS AN INTEGER LOOP COUNTER.
C
C    CALLED BY:  ALRITH
C
C    SUBROUTINES CALLED:  NONE
C
      INTEGER TYPE,ERFLG
      DIMENSION RATO1(8),RATO2(8),TYPE(2),IMAGS(24),IMAGC(24),IM(24)
C
C       SKETCH GUIDES.
C
      DATA IMAGS/ 2, 5, 3, 5, 6, 6, 3, 9, 8, 8, 8, 2, 2, 8, 6, 3,
     *            1, 0, 0, 1, 1, 1, 1, 1/
C
```

```
      C         COMPOSITE GUIDES.
      C
                DATA IMAGC/ 8, 9, 5, 6, 2, 2, 3, 1, 3, 8, 3, 3, 8, 9, 1, 8,
               *            0, 0, 1, 1, 0, 1, 0, 1/
      C
      C         CHECK FOR AN ERROR ENTRY.
      C
                IF (TYPE(1).EQ.0) GO TO 2
                IF (TYPE(1).EQ.1) GO TO 6
      C
      C         ERROR ENTRY.
      C
                WRITE(6,100)
            100 FORMAT('  COMP :  WRONG TYPE')
                ERFLG=1
                RETURN
      C
      C         ENTRY OK, LOAD SKETCH COEFFICIENTS.
      C
              2 DO 4 I=1,24
                   IM(I)=IMAGS(I)
              4 CONTINUE
      C
      C         SKETCH COEFFICIENTS READY, START PROCESSING.
      C
                GO TO 8
      C
      C         ENTRY OK, LOAD COMPOSITE GUIDES.
      C
              6 DO 7 I=1,24
                   IM(I)=IMAGC(I)
              7 CONTINUE
      C
      C         COMPOSITE COEFFICIENTS READY, START PROCESSING.
      C
              8 ECLUD=0.0
                DO 10 I=1,8
      C
      C         COMPARISONS ARE MADE USING PRE-DETERMINED GUIDES.
      C
                   IF (IM(I+16).EQ.0) GO TO 10
                   ECLUD=ECLUD+(RATO1(I)-RATO2(I))**2
             10 CONTINUE
                ECLUD=SQRT(ECLUD)
                RETURN
                END
```

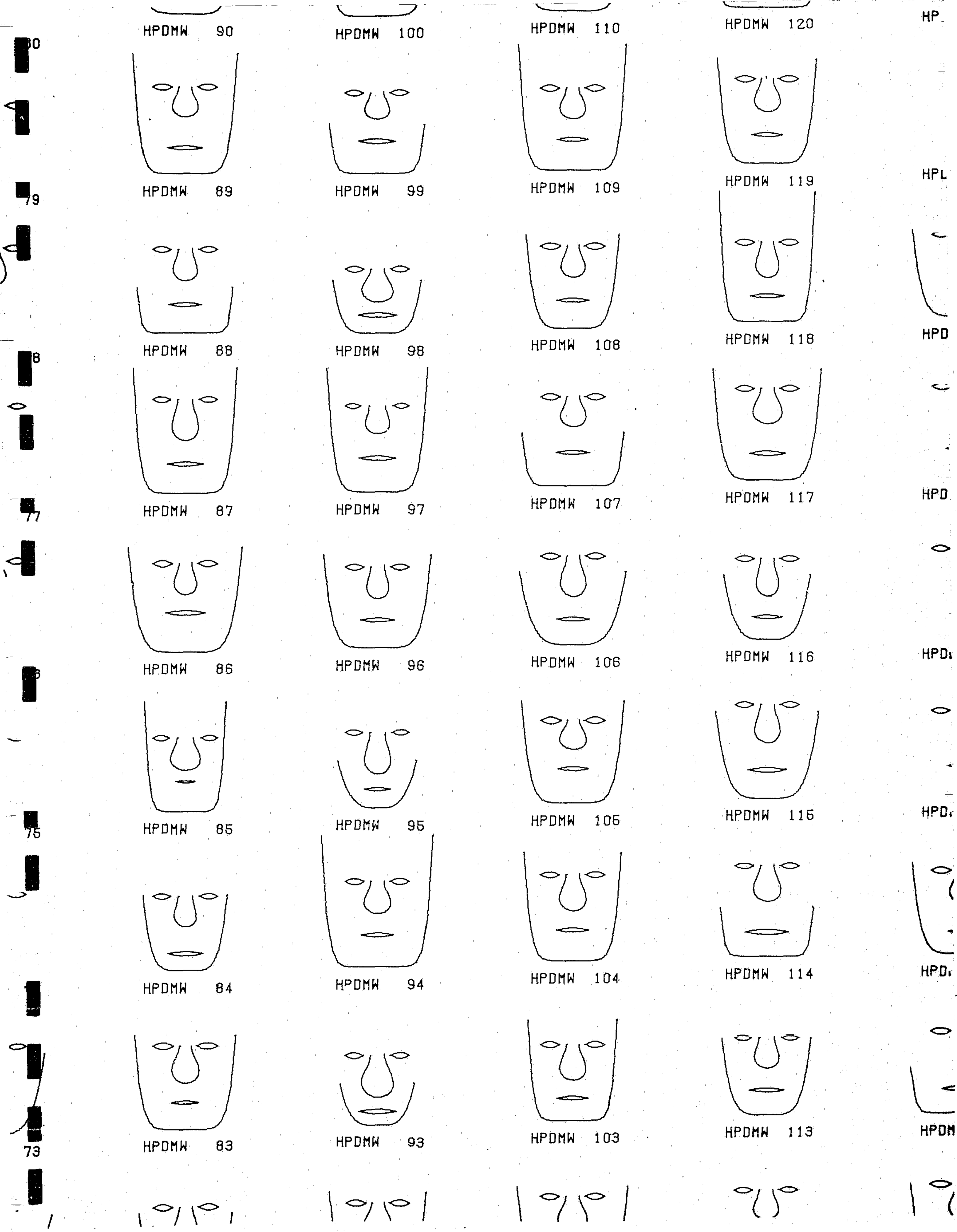
** ALR **   FILE: 04760$0$0$*DUMP   TIME: 11:24:16

```
      SUBROUTINE ALRIT(TYPE,CG,MEAS1,MEAS2,RATO1,RATO2,ECLUD,ERFLG)
C
C
C     PURPOSE:  ALRIT PROVIDES A CONTROL STUCTURE FROM WHICH
C               TO EXECUTE DIFFERENT MODULES OF THE LOOK ALIKE
C               ALGORITHM.
C
C
C     FORTRAN CALLLNG SEQUENCE:
C
C             CALL ALRIT(TYPE,CG,MEAS1,MEAS2,RATO1,RATO2,ECLUD,ERFLG)
C
C
C     EXTERNAL VARIABLES:
C
C              TYPE IS INTEGER ARRAY DIMENSIONED TWO (2).  TYPE
C                     SPECIFIES WHICH COEFFICIENTS ARE TO BE USED
C                     AND THE REPRESENTATION FROM WHICH THE
C                     MEASUREMENTS WERE TAKEN.
C
C                CG IS AN INTEGER ARRAY DIMENSIONED FOUR (4).  CG
C                     CONTAINS CONTROL FLAGS FOR SPECIFYING WHICH
C                     MODULES ARE TO BE USED.
C
C             MEAS1 IS AN INTEGER ARRAY DIMENSIONED NINE (9).
C                     MEAS1 CONTAINS A SET OF MEASUREMENTS.
C
C             MEAS2 IS AN INTEGER ARRAY DIMENSIONED NINE (9).
C                     MEAS2 CONTAINS A SET OF MEASUREMENTS.
C
C             RATO1 IS A REAL ARRAY DIMENSIONED EIGHT (8).
C                     RATO1 CONTAINS RATIOS FORMED FROM THE
C                     MEASUREMENTS IN MEAS1.
C
C             RATO2 IS A REAL ARRAY DIMENSIONED EIGHT (8).
C                     RATO2 CONTAINS RATIOS FORMED FROM THE
C                     MEASUREMENTAS IN MEAS2.
C
C             ECLUD IS A REAL VARIABLE.  ECLUD RETURNS THE
C                     ECLUDIAND DISTANCE BETWEEN RATO1 AND RATO2.
C
C             ERFLG IS AN INTEGER VARIABLE.  ERFLG RETURNS AN
C                     ERROR INDICATOR.
C
C     INTERNAL VARIABLES:
C
C             CKSUM IS AN INTEGER SCRATCH STORAGE.
C
C     CALLED BY:  N/A
C
C     SUBROUTINES CALLED:
C
C                1)  RATI1
C                2)  RATI2
C                3)  ALGOR
C                4)  COMP
C
```

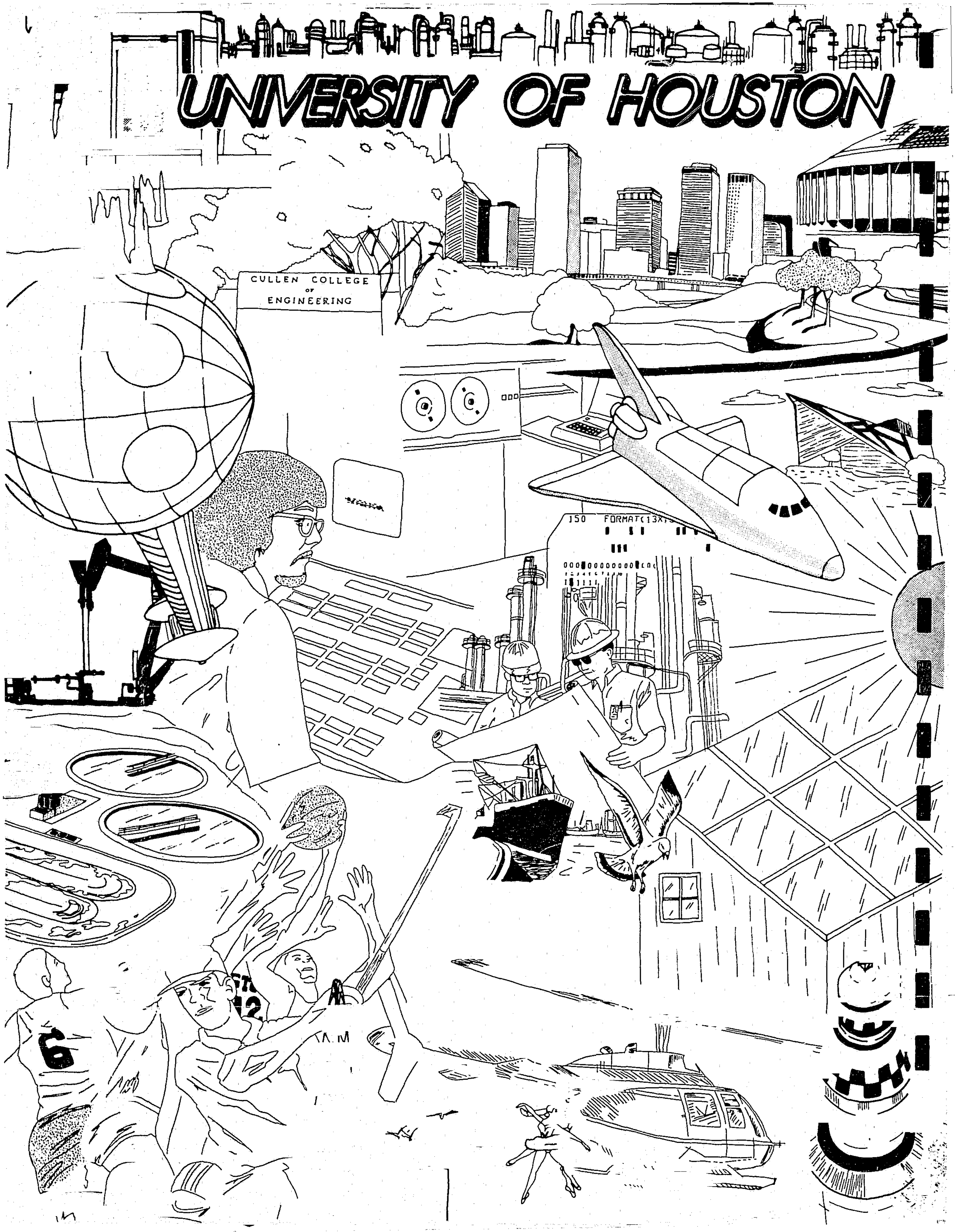
```
      INTEGER TYPE,CG,ERFLG,CKSUM
      DIMENSION CG(4),MEAS1(9),MEAS2(9),RATO1(8),RATO2(8),TYPE(2)
C
C       ASSUME ERROR ENTRY AND CHECK ASSUMPTION
C
      ERFLG=1
      CKSUM=CG(1)*8+CG(2)*4+CG(3)*2+CG(4)
      IF (CKSUM.EQ.0.OR.CKSUM.GE.12) RETURN
      IF (TYPE(1).LT.0.OR.TYPE(1).GT.1) RETURN
      IF (TYPE(2).LT.0.OR.TYPE(2).GT.3) RETURN
      IF (CG(3).EQ.1.AND.TYPE(2).EQ.1) RETURN
      IF (CG(3).EQ.1.AND.TYPE(2).EQ.3) RETURN
C
C       CONTROL GROUP INPUT OK, PROCEED
C
      ERFLG=0
C
C       FORM ONLY ONE RATIO SET?
C
      IF (CG(1).EQ.1) CALL RATI1(TYPE,MEAS1,RATO1,ERFLG)
C
C       ERROR RETURN FROM RATI1?
C
      IF (ERFLG.EQ.1) RETURN
C
C       NO ERROR, FORM TWO RATIO SETS?
C
      IF (CG(2).EQ.1) CALL RATI2(TYPE,MEAS1,MEAS2,RATO1,RATO2,ERFLG)
C
C       ERROR RETURN FROM RATI2?
C
      IF (ERFLG.EQ.1) RETURN
C
C       USE ALGORITHM ON RATO1?
C
      IF (CG(3).EQ.1) CALL ALGOR(TYPE,RATO1,ERFLG)
C
C       ERROR RETURN FROM ALGOR?
C
      IF (ERFLG.EQ.1) RETURN
C
C       COMPARE THE TWO SETS OF RATIOS?
C
      IF (CG(4).EQ.1) CALL COMP(TYPE,RATO1,RATO2,ECLUD,ERFLG)
C
C       FINISHED, EVEN IF AN ERROR OCCURRED
C
      RETURN
      END
```

** FORG2 ** FILE: 0476OOO$*DUMP TIME: 11:24:17

```
      SUBROUTINE SORT(NUM,ID,DIST)
      DIMENSION ID(700,2),DIST(700)
      N=NUM
      NUM1=NUM-1
    5 IF (DIST(N-1).LT.DIST(N)) GO TO 30
      N1=N-1
      DO 20 I=N1,NUM1
         IF (DIST(I).LT.DIST(I+1)) GO TO 30
         T=DIST(I)
         DIST(I)=DIST(I+1)
         DIST(I+1)=T
         DO 10 K=1,2
            J=ID(I,K)
            ID(I,K)=ID(I+1,K)
            ID(I+1,K)=J
   10    CONTINUE
   20 CONTINUE
   30 N=N-1
      IF (N.NE.1) GO TO 5
      RETURN
      END
```



HPDMW 90
HPDMW 89
HPDMW 88
HPDMW 87
HPDMW 86
HPDMW 85
HPDMW 84
HPDMW 83
HPDMW 100
HPDMW 99
HPDMW 98
HPDMW 97
HPDMW 96
HPDMW 95
HPDMW 94
HPDMW 93
HPDMW 110
HPDMW 109
HPDMW 108
HPDMW 107
HPDMW 106
HPDMW 105
HPDMW 104
HPDMW 103
HPDMW 120
HPDMW 119
HPDMW 118
HPDMW 117
HPDMW 116
HPDMW 115
HPDMW 114
HPDMW 113


UNIVERSITY OF HOUSTON
CULLEN COLLEGE OF ENGINEERING

END