DANIEL F. MERRIAM, Editor

FORTRAN II PROGRAMS FOR 8 METHODS OF CLUSTER ANALYSIS (CLUSTAN I)

By

DAVID WISHART
University of St. Andrews

COMPUTER CONTRIBUTION 38
State Geological Survey
The University of Kansas, Lawrence
1969

in cooperation with the
American Association of Petroleum Geologists
Tulsa, Oklahoma

Editor's Remarks

Publication of this program "FORTRAN II programs for 8 methods of cluster analysis (CLUSTAN I)" by David Wishart represents another milestone in the COMPUTER CONTRIBUTION Series. It will be the last FORTRAN II program published, although about 20 percent of the orders now are for programs in FORTRAN II. We feel by this time most installations will have available FORTRAN IV compabilities.

This program is probably one of the best documented ones we have published. It is appropriate too, in that it shows what can be done with a small, slow computer when necessary. The system has been converted for use on the IBM System 360/Model 44 and that version will be available also. For a limited time, the Geological Survey will make available on magnetic tape this system of programs for $60.00 (U.S.). If punched cards are required, an extra $10.00 is needed to cover handling charges.

An up-to-date list of publications may be obtained by writing Editor, Computer Contributions, Kansas Geological Survey, University of Kansas, Lawrence, Kansas 66044 U.S.A.

COMPUTER CONTRIBUTIONS

Kansas Geological Survey
University of Kansas
Lawrence, Kansas

Computer Contribution

(continued on inside back cover)

* Active Member, ° Associate Member, + Junior Member, American Association of Petroleum Geologists.

FORTRAN II PROGRAMS FOR 8 METHODS OF CLUSTER ANALYSIS (CLUSTAN I)

by

David Wishart

INTRODUCTION

The intensity of study of numerical classification methods during the last decade is probably due to 3 factors: (a) the difficulty (perhaps impossibility) of defining a general model for a wide range of applications and all types of data, (b) the theoretical problems arising from nonstandard data distributions (the mere existence of a classification problem practically implies heterogeneous data), and (c) the advent of widespread computational facilities. In the absence of a generally accepted theory of classification, considerable emphasis has been placed on intuitive rules and empirically justified procedures. From such weak theses, it is inevitable that at any indecisive stage there will be recourse to the original trial-and-error techniques. Consequently, researchers who discover new classification problems, or measure new data sets for existing subjects, will often only be satisfied completely by the familiar many-method comparitive exercise. Such experiments require computational facilities and ready made programs due to the enormous number of calculations required by these procedures - CLUSTAN I has been developed to meet these demands.

Following the decision to prepare a comprehensive suite of classification programs, it is immediately apparent that there are certain generalizations which can be made to simplify the design of the system. For example, most clustering methods require the initial computation of a similarity matrix - hence we can confine this calculation to one general routine (CORREL), and use smaller programs for each of the individual methods. In order to allow for the addition to the system of those methods which do not use a similarity matrix, a separate initial routine (FILE) should be used to control all data input, transformations and evaluation of background statistics. Finally, because the results obtained from the clustering programs can be expressed in a standard form, a single routine (RESULT) can be designed to control all necessary cluster interpretation functions. In general, an ideal clustering program library should be flexible sufficiently to be used with all types of data, allow for the introduction of additional clustering programs, handle reasonably sized problems and be defined suitably to have general machine accessibility.

The present set of programs (CLUSTAN I) was developed to meet these considerations as a design system from which could evolve a more sophisticated version - CLUSTAN II. Although it contains some major omissions, such as the provision for 'missing' data, CLUSTAN I is reasonably flexible, and can classify by the following 8 standard methods:

1. Nearest neighbor
2. Farthest neighbor
3. Group average
4. Centroid
5. Median
6. Ward's error sum
7. Lance-Williams flexible
8. Mode analysis

The configuration 1 - 8 currently is being extended to include the following additional methods:

9. Information analysis
10. k - dendrogram ultrametric
11. Association analysis
12. Divisive and agglomerative group analysis.

The present interest in CLUSTAN I lies mainly in testing its ease of implementation, and extent of usage. Such experience which is gained from the use of the system will contribute, together with suggestions concerning its improvement, toward the definition and development of CLUSTAN II in 1970.

GENERAL CONCEPTS OF CLUSTERING

The actual process of clustering is extremely straightforward. M variables are measured for each of N objects (samples) and the similarity between any two objects, between two groups of objects or between an object and a group is defined for each method in terms of the types of variables which are used. There are four common variable types: numeric, binary, ordered multistate and unordered multistate. Almost all similarity measures are defined for data in either the numeric or binary modes, and consequently multistate characters or mixed variable types must first be transformed (such transformations are suggested later). Next, the objects are clustered systematically according to the computational rules of an appropriate method. The analysis objective, common to all methods, is to arrive at a grouping of the N objects which is 'meaningful' in the context of the data subject. It is the interpretation of 'meaningful' which is difficult and differs with discipline and problem.

Most similarity measures and clustering methods so far proposed have been defined from intuitive reasoning, and justified by the empirical results of 'known' data. Sometime groupings are required for which the overall variable distribution within each class is minimized; clustering processes designed for this purpose are referred to as 'minimum-variance' methods - techniques 2, 6 and 7 of the CLUSTAN I configuration are notably of this type. Alternatively, the resultant classes may be required to be 'natural' or 'distinct', for which some degree of internal variation is allowed; we call such techniques 'natural class' methods, of which nearest neighbor and mode analysis are good examples (for a discussion of 'natural class' and 'minimum-variance' methods see Forgey, 1964, 1965; and Wishart, 1968b, 1968c). In absence of any sound theoretical basis for the subject, perhaps the most practical approach to clustering is to experiment with several methods until it becomes apparent which methods are most likely to succeed with a particular problem. The longterm view of such studies is that, if a satisfactory theory of classification can be formulated, then the weight of substantial empirical evidence will be a major factor in its implementation.

We can state the general result, obtained from clustering N objects as follows:

(i) There shall be K groups of objects (N > K > 1) such that each group contains at least one object.

(ii) Each object may be assigned to no group (if the object is 'unclassifiable' in terms of the method), one group only (if the method forms disjoint clusters), or more than one group (if overlapping clusters are permitted).

For the purposes of CLUSTAN I, the rare incidence of overlapping class methods will be avoided so that the structure of disjoint clusters can be specified by one array, called the classification array ICLA. This is usually punched on cards by the clustering programs, but may be prepared by hand if object groupings from some other source are to be examined. The format is:

Card 1 Identification title (Format 20A4)
Cards 2,3,... Classification array (ICLA(I), I=1, N) (Format 26I3)

The identification title should specify the method name, and sufficient information to associate the grouping with a particular analysis. The classification array takes the values:

ICLA(I) = 0, if object I is unclassified.
ICLA(I) = k, if object I belongs to cluster k.

The code k should be used consistently throughout ICLA for every object belonging to cluster k, but the codes for the K clusters are not necessarily sequential from 1 to K (see, for example, the classification arrays punched by HIERAR for the Lance-Williams flexible method of the trial data schedule).

PROGRAM DESCRIPTIONS AND SEQUENCES

CLUSTAN I contains five programs which communicate via a disk-stored data file (the modification or simulation of this data file is straightforward). The general functions of these programs, in their normal operating sequence, are:

1. FILE - Loads the input data to disk
2. CORREL - Computes the similarity matrix
   either
3. MODE - Clusters by mode analysis (method 8)
   or
4. HIERAR - Clusters by methods 1 - 7
5. RESULT - Prints data file values and cluster interpretation statistics.

Programs FILE, CORREL and RESULT represent the basic software, and any new clustering program (CLUSTER), which is written subsequently for the system will function in the sequence:

FILE CORREL CLUSTER

if a similarity matrix is required for CLUSTER, or:

FILE CLUSTER

otherwise. The particular facilities available with these programs are as follows:

1. FILE
   (a) reads data in binary and/or numeric form.
   (b) computes a principal components analysis for up to 35 numeric variables (optional).
   (c) computes product-moment correlations for up to 35 numeric variables (optional).
   (d) computes numeric variable means and standard deviations.
   (e) transforms numeric data to standard scores (optional).

Notes: because FILE initializes a new data record, no previously stored data are assumed in the disk file.

2. CORREL
   (a) evaluates either a numeric or binary similarity coefficient matrix.
   (b) can mask some of the filed data from the computation of similarity coefficients by means of the masking facility.
   (c) provides 19 standard similarity coefficients and a means of defining new measures.
   (d) searches for the k objects which are most similar to each of the N objects (the k-linkage lists).

Notes: CORREL can follow any program - hence the existing similarity matrix can be replaced by a new one without re-executing FILE.

3. MODE
   (a) refers to the similarity matrix and and k-linkage lists, and cannot, therefore, follow FILE or HIERAR.
   (b) makes no change to the disk-stored file. Object groupings are punched on cards with the format previously described.
4. HIERAR
   (a) refers to the similarity matrix, and cannot, therefore, follow FILE or HIERAR.

(b) destroys the similarity matrix and k-linkage lists on exit.
(c) Object groupings are punched on cards.

5. RESULT (a) prints all filed data blocks on user request.
(b) evaluates cluster diagnostics for any classification arrays which are punched in the standard format.

Notes: RESULT makes no change in the data file, and can follow any program. Error messages are given if data selected for output are not filed.

The overall dimensions of the input data are:
N(number of objects/samples) - maximum 250
MN(number of numeric variables) - maximum 200
MB(number of binary variables) - maximum 400

These maxima have been determined by the core size of the IBM 1620 II for which CLUSTAN I was developed. They can, however, be extended easily, and instructions for such modifications are given in the system operating manual (Wishart, 1968a).

General Data Considerations

We consider variables of the following four types:

1. Numeric (quantitative; e.g. height; percentage)
2. Ordered multistate (semiquantitative; e.g. many, some, few, none)
3. Unordered multistate (qualitative; e.g. color - white, red, brown, black)
4. Binary (qualitative; e.g. presence/absence; yes/no)

For the purpose of evaluating similarities, data should be expressed only in numeric and binary modes. In general, a classification data matrix takes the form of either b_{ij} - binary mode, or x_{ij} - numeric mode, where

$b_{ij} = 1$, if binary variable j is 'present' for case i

or $b_{ij} = 0$, if binary variable j is 'absent' for case i

and x_{ij} is the value of numeric variable j for case i.

Table 1a. - Typical binary input matrix for 4 cases with 10 binary variables (1 = present, 0 = absent).

	Binary variables									
CASES	1	2	3	4	5	6	7	8	9	10
1	1	0	0	1	0	1	0	0	0	0
2	0	0	1	0	0	1	1	0	1	1
3	1	0	0	0	1	0	0	0	1	0
4	1	1	0	0	0	1	1	1	1	1

Table 1b. - Examples of numeric data in raw and standard form for 5 numeric variables expressed by 4 cases.

	Numeric variables (raw data)				
CASES	1	2	3	4	5
1	25	0.8	104	10	0.06
2	32	0.73	102	2	0.01
3	15	0.18	98	5	0.04
4	43	0.25	107	7	0.02

	Numeric variables (standardized)				
CASES	1	2	3	4	5
1	-.37	1.12	0.38	1.37	1.43
2	.32	.87	-.23	-1.37	-1.17
3	-1.35	-1.12	-1.45	- .34	.39
4	1.40	-.87	1.30	.34	-.65

Important: No missing values are permitted. Table 1a contains an example of a binary data matrix for 4 cases and 10 variables; in practice, the zeros are not stored because it is usually more economical to list, for each case, the codes of those binary variables which are 'present'. Thus the input data for case 1 of Table 1a would be

1 4 6

and for case 2

3 6 7 9 10

and so on. Table 1b shows a numeric data matrix of 5 numeric variables measured for 4 cases; one matrix contains the raw data, whereas the other shows the results of 'standardization' or the 'standard scores'. A measure of the similarity between two cases, in terms of numeric data, is the Euclidean distance coefficient d^2_{ik} defined by

$$d^2_{ik} = \sum_{j=1}^{M} (x_{ij} - x_{kj})^2 .$$

In this example, the computation of the coefficients from the raw data introduces an obvious bias towards variable 3 due to its magnitude. Hence, in order to eliminate such bias the similarities should be evaluated from the matrix of standard scores x^*_{ij}, defined by

$$x^*_{ij} = (x_{ij} - \bar{X}_j)/S_j ,$$

where $\bar{X}_j$, S_j are the mean and standard deviation for variable j.

Table 2. - Example of mixed-mode data. Variable types are: A - binary, B - unordered multistate, C - numeric, D - ordered multistate.

	CASES				
Variables	1	2	3	4	
A. Sex	1	0	1	1	Male = 1; Female = 0
B. Hair color	3	2	4	1	White =1; Red =2; Brown = 3; Black = 4.
C. Age	25	32	15	43	Years
D. Salary	1	3	2	3	<$4000 =1; $4000 - $6000 = 2; >$6000 = 3.

The most satisfactory data for classification purposes would be either all binary or all numeric. However, problems inevitably occur for which the available data are a mixture of the four variable types original mentioned, and it is, therefore, necessary to transform data from one mode to another. Table 2 contains an example of a mixed-mode data matrix for 4 people with 4 variables. Variable A can be treated as binary where Male = 1 and Female = 0; B is an unordered multistate character for which it can be said that, in this instance, the relationship (White-Red) is just as strong as (White-Black), a correspondence which is not reflected if the codes 1 - 4 are treated as numeric variable values; although variable C (age) is not truly continuous, the number of possible scores on this character is sufficiently large for it to be treated as a numeric variable; variable D (salary) is an instance of an ordered multistate character. In this situation, the relationship (1-2) is stronger than (1-3), hence the term 'ordered', and any transformation of this variable should take the ordering into account.

In order to find some basis for the comparison of these people, transformations must be defined from each of the four variable types into the two basic data modes: binary and numeric. Furthermore, the resulting conversions must minimize the amount of bias which is introduced towards any one of the original variables. The following five transformations are by no means ideal, but do provide a simple means to resolve the mixed-mode problem.

Table 3a.- Suggested transformation from binary data to numeric. Resulting numeric variable should be standardized subsequently.

Binary			Numeric
'present'	1	T_{BN} →	1.0
'absent'	0		0.0

Table 3b. - Example of transformation from unordered multistate variables to binary.

Unordered multistate			Binary 1	2	3	4
White	1	T_{UB} →	1	0	0	0
Red	2		0	1	0	0
Brown	3		0	0	1	0
Black	4		0	0	0	1

Table 3c. - Transformations from ordered multistate to binary (T_{OB}), and numeric to binary ($T_{NO} + T_{OB}$).

Numeric		Zones	Ordered multistate		Binary 1	2	3
15	T_{NO} →	10 - 20	1	T_{OB} →	0	0	0
25		20 - 30	2		1	0	0
32		30 - 40	3		1	1	0
43		40 - 50	4		1	1	1

Table 3d. - Transformation from ordered multistate to numeric T_{ON}. Choice of substitute codes is critical, even if numeric variable is standardized.

Ordered multistate		Numeric
1	T_{ON} →	1.0
2		2.0
3		3.0

1. T_{BN} (binary to numeric).-The binary codes (1,0) are converted directly to the numeric values (1.0,0.0) as shown in Table 3a. A vector of such scores, if used with other numeric variables such as 'Age', should be standardized before similarities are evaluated. Hence, the actual codings of 'present' and 'absent' are arbitrary, because standardization will always yield the same standard scores.

2. T_{UB} (unordered multistate to binary).-If there are p states for the multistate character, p binary characters should be created, one for each state. A situation exhibiting state q would score 0 for each of these p binary characters with the exception of character q, for which the score should be 1 for 'present'. Table 3b illustrates the use of T_{UB} with the variable 'color of hair' in Table 2.

3. T_{OB} (ordered multistate to binary).-For an ordered multistate character having p states, (p-1) binary characters are created. The conversion differs from T_{UB} as demonstrated in Table 3c. In this exam-

ple, the binary characters 1, 2, 3 may be regarded as being attributes > 20, > 30, > 40 respectively. The resulting coding illustrates the means by which strong relationships (1-2) and weak relationships (1-4) are preserved.

4. T_{NO} (numeric to ordered multistate).-Table 3 also shows the path, via transformations T_{NO} and T_{OB}, by which a numeric variable is converted to binary. The range of the numeric variable is first 'zoned' into p subranges by transformation T_{NO} to construct an ordered multistate character. T_{OB} is used then to convert this character to binary form. There is no reason why the zones of the numeric variable should have equal length - for a skew or modal distribution, the zones would probably be best chosen if their frequences were roughly uniform.

5. T_{ON} (ordered multistate to numeric).-This transformation, illustrated in Table 3d, is weakest if generalized. Depending on the relationships between the p states, numeric values are chosen and replace the coded states accordingly. In the example, the values 1.0, 2.0 and 3.0 replace states 1, 2 and 3 respectively. The selection of these substitute codes is crucial, even if the variable is standardized subsequently, because their differences from the mean are reflected proportionately by the standard scores. Every attempt should be made to associate the inserted codes with the means of the ordered intervals which they represent. For example, a percentage variable which has been zoned 0-30, 30-50, 50-70, 70-100 could have substitute codes 15, 40, 60 and 85 if the distribution is known to be rectangular, or 22, 43, 57 and 78 if it is a normally distributed variate.

Table 4. - Summary of transformations from all variable types to binary and numeric.

FROM	TO Binary	Numeric
Binary	-	T_{BN}
Numeric	$T_{NO} + T_{OB}$	-
Ordered multistate	T_{OB}	T_{ON}
Unordered multistate	T_{UB}	$T_{UB} + T_{BN}$

Table 5. - Complete transformations to numeric and binary form for data of Table 2. Resulting binary matrix has 10 variables and is given in Table 1. Resulting numeric matrix shows raw scores which should be standardized before similarities are computed.

Original Variables		Binary Conversion: Cases 1	2	3	4	T Type	Numeric Conversion: Cases 1	2	3	4	T Type
A. Sex		1	0	1	1	-	1.0	0.0	1.0	1.0	T_{BN}
B. Hair Color	(1)	0	0	0	1	T_{UB}	0.0	0.0	0.0	1.0	T_{UB}
	(2)	0	1	0	0		0.0	1.0	0.0	0.0	+
	(3)	1	0	0	0		1.0	0.0	0.0	0.0	T_{BN}
	(4)	0	0	1	0		0.0	0.0	1.0	0.0	
C. Age	>20	1	1	0	1	T_{NO}	25.0	32.0	15.0	43.0	
	>30	0	1	0	1	+					
	>40	0	0	0	1	T_{OB}					
D. Salary	>1	0	1	1	1	T_{UB}	1.0	3.0	2.0	3.0	T_{ON}
	>2	0	1	0	1						

The transformation from unordered multistate to numeric (T_{UN}) is obtained by combining T_{UB} and T_{BN}. Table 4 contains a summary of the conversions from each variable state to the basic data modes in terms of the transformations discussed, and in Table 5, the mixed-mode data of Table 2 is shown after conversion to both binary and numeric modes.

Program FILE (data input)

The data input device is a user subroutine READ which is rewritten normally for each data deck. The general form of this subroutine is:

```
      SUBROUTINE READ (NUM,X,MN,LIST,LENG)
      DIMENSION X(1), LIST(1)
      READ 5, (X(I),I=1,MN),(LIST(J),J=1,LENG)
5     FORMAT ( ———— )
      RETURN
      END
```

One call of the subroutine is used to read the data for one sample: the value of NUM is the sample number as set by the main program, and it is available in the subroutine for error detection; MN (the number of numeric variables) is specified by the FILE parameter card. The remaining variables in the parameter list of READ should be read or calculated by the subroutines as follows:

X contains the MN numeric variable values for sample NUM. (LIST(I),I=1,LENG) specifies the codes of the LENG binary variables which are 'present' for sample NUM.

The value of LENG must be at least 1, and should be set by READ-LENG is most suitably calculated (see, for example, the version of READ used for the trial data set). FORMAT statement 5 is defined according to the format of the input data for each problem - thus general input format is allowed. If data are presented in one mode only, the input statement should be modified to:

either READ 5,(X(I),I=1,MN) (for numeric data only:MB=0)

or READ 5,(LIST(J),J=1,LENG) (for binary data only:MN=0)

(For an example of a numeric only version of subroutine READ, see cards 30810 - 30920).
It is recommended that READ also should contain PRINT statements to list the data as they are entered. This enables errors in the data cards or in READ to be detected at an earlier stage. Any transformations from the raw data into the binary or numeric modes should be completed previously in the data deck, or computed by suitable additional statements in subroutine READ.

The values of N, MN and MB must be specified on the FILE parameter card. If numeric data are presented, product-moment correlations can be computed for all pairs of numeric variables using parameter JCOR, and parameter NPCF can be used to select a principal components analysis. Both of these facilities are restricted to the special instance if $0 < MN \leq 35$ for storage reasons (blank parameter suppress their selection, and error messages inhibit their selection when MN >35). If a principal components analysis is selected, it is usual to request that eigenvectors be filed by inserting a nonblank character for JEIGV. However, suppresion of this option (by leaving JEIGV blank) can save up to 123 records of the disk file - if N is large, the calculation of a similarity matrix is liable to overlap errors because only 4300 records are reserved for the disk-data file. The numeric data matrix can be retained in either raw or standard form by the use of parameter ISTAND. Numeric similarity coefficients will be computed subsequently by CORREL from the raw data or standard scores, depending on the numeric data-file status determined by ISTAND. It is suggested that the normal medium should be standard scores, hence ISTAND should be nonblank.

Program CORREL (similarity coefficients)

Similarity coefficients can be computed from either binary or numeric data. The Euclidean distance coefficient, defined earlier, is an example of a dissimilarity coefficient which can be used with numeric data. The only other numeric similarity coefficient available with CORREL is the product-moment correlation, which is applied to the two vectors of variable scores corresponding to a pair of samples.

The remaining 17 similarity coefficients are all defined for binary data, where the matching information about two samples can be summarized in the notation of the 2 x 2 table given in Table 6: Cell A contains the number of binary variables which are 'present'

Table 6. - 2 x 2 binary conversion table from which all binary similarity coefficients are defined.

		sample i +	sample i -
sample k	+	A	B
	-	C	D

for both samples i and k; B is the number which are 'present' for k but 'absent' for i, and so on. Binary coefficients now are defined in terms of the numbers A, B, C, D; for example, the binary coefficient of Russel and Rao is defined as

$$S_{ik} \quad \frac{A}{A+B+C+D} = \frac{A}{MB}$$

Table 7. - List of similarity coefficients provided by COEF. Note that coefficients 20 - 25 are vacant, and code + 999.999 is inserted for any indeterminate value.

Coefficient Number	Form	Data
1	Euclidean distance: $\frac{1}{M}\Sigma(x_i-y_i)^2$	Numeric
2	$\frac{B+C}{M}$ (binary Euclidean distance)	Binary
3	Product-moment correlation: r_{ik}	Numeric
4	$\frac{A+D}{M}$	Binary
5	$\frac{A}{A+B+C}$	Binary
6	$\frac{2A}{2A+B+C}$	Binary
7	$\frac{2(A+D)}{2(A+D)+(B+C)}$	Binary
8	$\frac{A}{A+2(B+C)}$	Binary
9	$\frac{A+D}{A+D+2(B+C)}$	Binary
10	$\frac{A}{B+C}$	Binary
11	$\frac{A+D}{B+C}$	Binary
12	$\frac{(A+D)-(B+C)}{M}$	Binary
13	$\frac{A}{M}$	Binary
14	$\frac{1}{2}\left\{\frac{A}{A+C}+\frac{A}{A+B}\right\}$	Binary
15	$\frac{1}{4}\left\{\frac{A}{A+C}+\frac{A}{A+B}+\frac{D}{B+D}+\frac{D}{C+D}\right\}$	Binary
16	$\frac{A}{\sqrt{(A+C)(A+B)}}$	Binary
17	$\frac{AD}{\sqrt{(A+B)(A+C)(B+D)(C+D)}}$	Binary
18	$\frac{AD-BC}{\sqrt{(A+B)(A+C)(B+D)(C+D)}}$	Binary
19	$\frac{AD-BC}{AD+BC}$	Binary

CORREL provides a means of computing a similarity matrix, for one of 19 similarity coefficients, from either binary or numeric data. The particular coefficient which is used is selected from Table 7 by the CORREL parameter ICOEF, and the appropriate data set is chosen by means of the masking facility described below. All the coefficient computation is controlled by subroutine COEF (cards 12210 - 14030) which contains 6 empty program segments to permit the addition of other similarity coefficients to the list. Details of the necessary modifications required to introduce a new coefficient are given in the system operating manual (Wishart, 1968a).

If similarity coefficients are computed from all data presented - and this implies that the data are either all binary or all numeric - then the CORREL parameters MSKB and MSKN must both be blank, and no partial-mask cards are required.

If similarities are to be computed from some subset of the data matrix, then the masking facility should be used. The data medium can be chosen from one of the following:

1. Factor scores
2. Raw numeric data
3. Numeric standard scores
4. Binary data

Factor scores are selected by setting a nonzero value for NPC. The result is that numeric similarity coefficients are computed using the scores corresponding to the first NPC factors obtained from the principal components analysis of the numeric data (the principal components analysis also should have been selected by use of FILE parameter NPCF, and NPC $\leq$ NPCF). Binary or numeric similarity coefficients are selected according to the input data medium (if the input is all binary or all numeric), or according to the unmasked data mode which is determined by the masking facility through MSKB and MSKN (if mixed-mode data are presented). Assuming that numeric data only are presented for the computation of similarities (i.e. the binary data are masked completely, or absent), the choice between (2) raw numeric data or (3) numeric standard scores is determined by the status of the numeric data file through FILE parameter ISTAND. Similarly the binary data medium can be selected for the computation of similarities only if the numeric data are masked completely, or absent. The request for a binary similarity coefficient with parameter ICOEF, coupled with unmasked numeric data, or vice-verse, produces an error message and causes job termination.

The numeric or binary data modes can be masked, partially masked or unmasked by manipulation of the CORREL parameters MSKB and MSKN, and the partial-mask cards. The value of MSKN corresponds to the number of numeric variables which are to be masked, hence:

(i) If numeric data are not presented (i.e. MN=0) then MSKN should be zero and no partial-mask cards are used.

(ii) If numeric data are presented, and a subset of the numeric variables is used for the computation of similarities, then MSKN should be set equal to the number of numeric variables which are to be masked. The codes of those numeric variables which are masked should be punched on the partial-mask cards, and there should be precisely MSKN such codes, with 0 < MSKN < MN (the numeric variables are assumed to be coded from 1 to MN according to their input order).

(iii) If the entire numeric data matrix is masked (i.e. similarities are to be computed from the binary data), then MSKN should be set equal to MN. Because a total mask does not require the specification of masked variables, no partial-mask cards are used.

Similar options apply to the manipulation of binary data using MSKB. The general rules of masking can be stated as follows:

1. Some data must be unmasked (to permit computation)
2. When mixed-mode data are provided, a total mask must be used in one of the modes.
3. The masked variable codes punched on partial-mask cards only can refer to variables in one of the modes.

Table 8. - Summary of rules and options of CORREL masking facility using parameters MSKN, MSKB and partial-mask cards.

Data selected for the computation of similarities	MSKN	MSKB	Partial Mask Cards
All numeric (no binary given).	0	0	Not used
Some numeric (no binary given	0<MSKN<MN	0	List of masked numeric codes
All binary (no numeric given).	0	0	Not used
Some binary (no numeric given).	0	0<MSKB<MB	List of masked binary codes.
All numeric (binary given).	0	MB	Not used
Some numeric (binary given).	0<MSKN<MN	MB	List of masked numeric codes.
All binary (numeric given)	MN	0	Not used
Some binary (numeric given).	MN	0<MSKB<MB	List of masked binary codes.

Table 8 gives all the possible combinations of data selection which can be obtained from manipulation of MSKB, MSKN and the partial-mask cards. The masking facility effectively enables the samples to be classified in terms of one set of data, while additional or original data can be carried on the file to provide background statistics during the interpretive routine RESULT. Thus, for example, in a mixed-mode problem, the transformation of numeric data to binary form for classification does not exclude the possibility of computing means and standard deviations for the original numeric variables.

Program MODE

Mode analysis (Wishart, 1968b, 1968c) is a derivative of nearest neighbor (single linkage) which searches for 'natural' subgroupings of the data by estimating disjoint density surfaces in the sample distribution. A sample point i becomes 'dense' if the similarity coefficient threshold d reaches a level at which at least k other points j satisfy $S_{ij} \geq d$ - that is, at least k points are 'similar' to i at threshold d. The N samples are ordered according to their kth. highest similarity coefficient (derived directly from the k-linkage lists), and this ordering determines the sequence in which the points become dense. A hierarchy is defined from this ordering by selecting similarity coefficients sequentially by magnitude from the k-linkage list; thus at each cycle a new dense point is introduced. Each new dense point either initiates a new cluster nucleus, joins an existing cluster or fuses two or more clusters; cluster groupings are given prior to fusion cycles so that the output of classifications is restricted. The first cycle initiates a single cluster, and the last cycle causes the entire population to be fused into one cluster. Sometimes only one cluster is formed (indicating no natural subgrouping), but usually the analysis reaches the point at which a maximum number of clusters is isolated. This grouping may be considered most significant. However, certain spurious groupings are sometimes thrown out, particularly with binary data if a dense point may initiate a new cluster nucleus at introduction, and then fuse to an existing cluster at the next cycle. Thus another constraint is imposed on the output of groupings, namely, that at fusion, groupings are output only if at least two of the clusters fused possess more than MINFUS dense points - in this manner, the output of groupings is restricted to those cycles after which 'important' clusters are lost at fusion.

Two levels of classification are possible: for the complete classification level every nondense point is allocated to its cluster of 'best fit', so that the entire population is classified; at the nuclei level, only the dense points and each nondense point which is 'similar' to a dense point are classified - the remaining nondense points are unclassified. Hence, the optional nuclei classifications can be used to indicate the core of each cluster. The enclosure ratio is defined as

$$\text{enc} = \frac{1}{N} \times \text{(the number of samples classified at the nuclei level)}$$

and enc provides a measure of the 'resolution' of cluster centers. A high enclosure ratio suggests a stable separation of the population.

The MODE parameters now can be specified as follows:

KL - density level k, determined by the user

MINC - minimum number of clusters which are of interest (usually 1)

MINFUS - minimum cluster size criterion for output of groupings.

PERC - minimum enclosure ratio to terminate the analysis, provided that the number of clusters is not greater than MINC (usually, PERC = 0.8).

ION - nonblank to select nuclei classifications, otherwise blank.

Selection of the density level KL is not crucial, but dependent on N; the following rough guide to KL and MINFUS values has been determined empirically:

Number of samples (N)	KL	MINFUS
less than 30	1	1
30 - 90	2	2
greater than 90	3	3

It should be noted that a perfectly valid analysis may be obtained if a blank card is inserted in lieu of the MODE parameters (in this instance, the values KL=3, MINC = 1, MINFUS = 0, PERC = 0.8 and ION = blank are allocated by the program).

MODE requires access to the similarity matrix and k-linkage lists (computed for any similarity coefficient). At exit, these and all other sections of the data file remain unaltered. Perhaps the most logical coefficient to use in conjunction with mode analysis is the Euclidean distance (code 1 for numeric, 2 for binary data), but because the algorithm can be generalized for all similarity measures, no restriction has been imposed for the program.

Program HIERAR

The general hierarchical fusion process is defined as follows:

1. Every sample initially constitutes a single-element cluster.
2. At each fusion step, those two clusters which are 'most similar' as determined by a particular method, are combined.
3. The analysis terminates when the initial N clusters have been agglomerated, in (N-1) fusion steps, into a single cluster universe.

(For a full discussion of the hierarchical fusion process used for HIERAR, see Lance and Williams, 1967; Wishart, 1969; Ward's error sum method is given by Ward, 1963).

Parameter KTRAN specifies the particular clustering method which is to be used. Some methods can be used logically only with Euclidean distance coefficients, and in Table 9 they are listed together with their coefficient compatibility. An error message is given if a method selected for HIERAR is incompatible with the similarity coefficient matrix filed by CORREL.

KA and KB specify those groupings which are printed and punched on to cards (the punched deck then can be used as input to RESULT for detailed cluster diagnostics). Usually, KA=2 and KB=10, thus the output is confined to the last 9 groupings in the fusion

Table 9. – List of methods available with HIERAR, and their similarity coefficient compatibility.

KTRAN	METHOD	Compatible Similarity Coefficients
1	Nearest neighbor	all
2	Farthest neighbor	all
3	Group average	all
4	Centroid	1,2
5	Median	1,2
6	Ward's method	1,2
7	Flexible	1,2

sequence – from 10 clusters down to 2. The value of BETA, which need be set for method 7, should be negative: Lance and Williams (1967) suggest the value – 0.25.

Because HIERAR modifies the elements of the similarity matrix, this matrix is destroyed at exit. It is suggested therefore that a suitable device should be used to store a permanent version of the data file, and a fresh copy of this file should be transferred to the 'working region' of the programs after each execution of HIERAR.

The entire fusion process can be represented diagramatically by a dendrogram or 'linkage tree'. Each sample is located at a branch of the tree, and each fusion in the sequence is described by connecting two branches on the diagram. The connections should be drawn parallel to a similarity scale, and some writers have suggested that a significant grouping can be detected where two fusions are separated by a noticeable drop in the similarity level. Although HIERAR does not plot a dendrogram, subroutine DENDRO orders the sample codes in their dendrogram sequence providing a key to its construction (in the trial data schedule, the dendrogram for the flexible method has been drawn by hand). DENDRO also contains punch statements on comments which could be used to output the fusion data in a form suitable for the construction of the dendrogram by an offline plotting program (such a program will be developed for the later extension of CLUSTAN I).

Program RESULT

Almost all the output which can be obtained from RESULT is selected by punching any nonblank character in the appropriate parameter column of the input card. The exceptions are: the number of factor scores (INSCOR) and the number of k-linkage lists (INKLIS) which are to be listed – these values must be right-justified integers if selected, or blank otherwise.

The following formulae define the various statistics which can be filed, and printed by RESULT:

(1) $\bar{X}_j = \frac{1}{N} \sum_i x_{ij}$ (mean for numeric variable j)

(2) $S_j = \sqrt{\frac{1}{N} \sum_i (x_{ij} - \bar{X}_j)^2}$ (standard deviation for variable j)

(3) $r_{jp} = L_{jp} / \sqrt{L_j L_p}$ (product-moment correlation between variables j and p)

where $L_{jp} = N \sum_i x_{ij} x_{ip} - \sum_i x_{ij} \sum_i x_{ip}$

and $L_j = N \sum_i x_{ij}^2 - (\sum_i x_{ij})^2$

(4) The principal components analysis is obtained by computing eigenvectors and eigenvalues from the product-moment correlation matrix (the covariance matrix is not computed, and therefore cannot be used).

(5) Factor scores are computed from the product:
(eigenvector) x (standard scores)

(6) The binary variable frequency f_j is defined as the number of objects for which binary variable j is 'present'.

(7) $p_j = f_j \times 100/N$ (percentage occurrence for binary variable j)

(8) Under the heading 'CLUSTER DIAGNOSTICS', RESULT can evaluate the following statistics for each cluster C in a grouping of the objects:

NUMERIC: CLUSTER MEAN: $\bar{X}_{cj}$
STANDARD DEVIATION: S_{cj}
F-RATIO: $F_j = S_{cj}/S_j$
T-VALUE: $T_j = (\bar{X}_{cj} - \bar{X}_j)/S_j$

BINARY: CLUSTER FREQUENCY: f_{cj}
PERCENTAGE OCCURRENCE: p_{cj}
PERCENTAGE RATIO: $V_j = p_{cj}/p_j$

Each cluster grouping must be input to RESULT on cards using the standard format. Such classification arrays are punched by programs MODE and HIERAR: they also may be prepared by hand using other sources of classification. Although the F and T statistics correspond to strict statistical confidence tests, no test of significance may be performed unless the numeric variable distributions can be assumed normal. However, these values do give an indication of the 'diagnosticity' of certain variables.

DESCRIPTION OF DECK SETUP AND PARAMETERS

1. DECK SETUP: (Recompile subroutine read)

   (FILE execution)
   Card 1 - Title
   Card 2
   Data Deck

   (CORREL Execution)
   Card 1
   Partial-Mask card(s)

   (MODE Execution)
   Card 1

   (HIERAR Execution)
   Card 1

   (RESULT Execution)
   Card 1
   Classification arrays

2. FILE INPUT:

   Card 1 identifies the job. A data description title may be punched in columns 1-80. (Format 20A4).

   Card 2

Columns	Purpose
1-4	N - number of samples (I4)
5-8	MN - number of numeric variables (I4)
9-12	MB - number of binary variables (I4)
13	*ISTAND - standard scores selector (A1)
14	*JCOR - product-moment correlations selector (A1)
15-16	NPCF - number of factor scores computed and filed for a principal components analysis (I2)
17	*JEIGV - filing of eigenvectors selector (A1)

3. CORREL INPUT:

   Card 1

Columns	Purpose
1-2	ICOEF - code of similarity coefficient used (I2)
3-4	KMAX - number of k-linkage lists filed (I2)
5-6	NPC - number of factor scores used to compute similarities; insert zero if similarities are computed from the binary or numeric data (I2).
7-9	MSKB - number of binary variables masked from the computation of similarities (I3).
10-12	MSKN - number of numeric variables masked (I3).

   PARTIAL-MASK CARDS

   These cards are used if a partial mask is required (see Table 8). The masked variable codes associated with the partially masked data mode should be punched, with 20-codes-per-card, until exhausted (i.e. until MSKB or MSKN is exhausted). The format is (20I4), hence each code should be punched as an integer, right justified to every fourth column.

4. MODE INPUT:

   Card 1

Columns	Purpose
1-2	KL - density level (I2)
3-4	MINC - minimum number of clusters considered (I2)
5-6	MINFUS - minimum cluster size criterion for classification output (I2)
7-11	PERC - minimum enclosure ratio to terminate analysis (F5.3).
12	*ION - nuclei classifications selector (A1)

5. HIERAR INPUT:

   Card 1

Columns	Purpose
1	KTRAN - method selector: see Table 9 (I1)
2-4	KA - minimum number of clusters of interest (I3).
5-7	KB - maximum number of clusters of interest (I3)
8-12	BETA - applies to method 7 only: suggest -0.25 (F5.2).

6. RESULT INPUT:

   Card 1

Columns	Purpose (FORMAT (A1) unless otherwise stated).
1	*INRAW - raw numeric data listing and variables maxima and minima.
2	*INSTAN - numeric standard scores
3	*INMEAN - numeric means and standard deviations.
4	*INCOR - numeric product-moment correlations.
5	*INEIG - eigenvalues, percentage and cumulative variance.
6	*INEIGV - eigenvectors
7-8	INSCOR - number of factor scores listed (I2).
11	*IBRAW - raw binary data listing.
12	*IBFREQ - binary variable frequencies

13	*IBPERC - binary variable percentage occurrences.
21	*IMCOEF - similarity matrix
22	IMKLIS - number of k-linkage lists (I2)

CLUSTER DIAGNOSTICS

31	*ICARAY - classification array
32	*ICLIST - listing of sample numbers for each cluster
33	*ICMEAN - numeric means, standard deviations, F-ratios and T-values for each cluster
34	*ICFREQ - binary variable frequencies
35	*ICPERC - binary variable percentage occurrences
36	*ICRAT - binary percentage ratios

CLASSIFICATION ARRAYS:

After the data-file values have been listed, using parameters INRAW to IMKLIS, RESULT reads classification arrays (punched with the standard format) and interprets cluster diagnostics. The style of input is to read an array, print diagnostics, read next array, . . . etc., and the program finishes on a 'TRAP ERROR' when all classification arrays have been read. For systems lacking this facility it may be necessary to specify the number of arrays to be read by RESULT using an additional parameter on card 1.

NOTES: *Insert any nonblank character to select operations, leave blank to suppress.

Installation notes

1. Simulating the disk-data file: Each main program contains a DEFINE DISK (10,4300) statement (cards 40, 7050, 14740, 22280, 31040 and 37320) which specifies the number of real or integer words (10) which can be written on one disk record, and the maximum permitted length (4300 records) of the disk-data file. When changing the file length to q records, card 12-110 should be changed to N2=q in order to detect overlap errors. All disk input/output is controlled by subroutine DISKIO which uses the FORTRAN II FETCH and RECORD statements (these operations are described in cards 860-990). When simulating the disk-data file, it is important to preserve the 10-word-per-record operating principle because all mainline programs use this format to compute file positions (e.g. subroutine GET). Furthermore, it is also important to ensure that central elements of an array may be accessed as if the array is partitioned into 10-word blocks; for example, the statement

```
      RECORD   (15)   (X(I), I = 1, 25)
```

would allocate (X(I), I = 1, 10) to record 15, (X(I), I = 11, 20) to 16, and (X(I), I = 21, 25) to the first half of record 17 - the last 5 positions of record 17 are not used. In this instance, if the single element P = X(13) is required the sequence

```
      FETCH    (16)   (X(I), I = 11,13)
      P = X(13)
```

should work.

To simulate the disk file, it should be possible to merely remove the five DEFINE DISK statements and rewrite subroutine DISKIO. The replacement version of DISKIO may take the form of new disk input/output statements suitable for a FORTRAN IV compiler with a disk system, or machine code operations which improve the access time. Alternatively, for large core machines, DISKIO may have 'own' arrays with which the entire disk file is held in core, or 'own' arrays and a paging system for block input/output using magnetic tape. When using magnetic tape storage, efficiency will be lost if it is not possible to store the entire similarity matrix in core at one time. Although access to the similarity matrix by CORREL, MODE and HIERAR is sequential, the number of disk READ operations during one scan may be so small that REWIND time on magnetic tape could cause considerable delays. When the disk file is held in core using 'own' arrays in DISKIO, it is necessary to communicate the record from FILE to CORREL, and from CORREL to MODE, HIERAR and RESULT. It is therefore suggested that cards 470 and 8330 of the listing should be replaced by a WRITE MAGNETIC TAPE operation to load the data file on to magnetic tape, for permanent record, following the completion of programs FILE and CORREL. The DISKIO option ISEL = 1 (cards 740-780) which is used to read the file parameters at the start of every program, can now be changed to a READ MAGNETIC TAPE operation which reloads the data file into core.

The storage format for the disk-data file is as follows:

RECORD NUMBER	DATA
1 - 2	N, MB, MN, NPCF, NPC, ISTAND, IMASK, IDATA, ICOEF, ITYPE, KMAX: disk-file parameters.
3 - 5	LNDATA, LBDATA, LMEANS, LVARS, L-CORS, LEIGS, LEIGVS, LSCORS, LENGS, LFR-EQS, LNMASK, LBMA-SK, LMAT, LKLIST, L-NEXT; disk record pointers.
6 - 7	(TEXT (I), I = 20); data identification card.
LNDATA	Start of numeric data file; $(X_{IJ}, J = 1, MN)$
LBDATA	Start of Binary data file; $(LIST_{IJ}, J = 1, LENG(I))$.
LMEANS	Numeric variable means; (XM(J), J = 1, MN)

LVARS	Numeric variable variances; (VAR(J), J = 1, MN)
LCORS	Numeric variable product-moment correlations; (COR(J), J = 1, MN * (MN + 1)/2)
LEIGS	Eigenvalues; (EIG(J), J = 1, MN)
LEIGVS	Eigenvectors; (EIGV (J), J = 1, MN*MN)
LSCORS	Start of factor scores file; (F_{1J}, J=1,NPCF)
LENGS	Binary sample list lengths; (LENG(I), I=1, N)
LFREQS	Binary variable frequencies; (IFREQ(J), J=1, MB)
LNMASK	Numeric variable mask; (XMASK(J), J=1, MN)
LBMASK	Binary variable mask; (MASKB(J), J=1, MB)
LMAT	Start of similarity matrix; row 1 -S (2, 1)
LMAT + 1	Second row of similarity matrix; S (3, 1), S (3,2)
LKLIST	k-linkage lists; see cards 28380 - 28550
LNEXT	end of file - next free record.

The following parameters have not been described in the text:

ISTAND = 1	if numeric file contains raw data
ISTAND = 2	if numeric file contains standard scores
IMASK = 1	if no data mask is used
IMASK = 2	if a data mask is used
IDATA = 1	if binary coefficients are computed
IDATA = 2	if numeric coefficients are computed
ITYPE = 1	if coefficient is similarity type
ITYPE = 2	if coefficient is dissimilarity type
XMASK(J) = 1.0	if numeric variable J is unmasked
XMASK(J) = 0.0	if numeric variable J is masked
MASKB(J) = 1	if binary variable J is unmasked
MASKB(J) = 0	if binary variable J is masked

A zero value for any parameter (e.g. IDATA, ITYPE, IMASK) or for any record pointer indicates that the associated data operation has not been completed.

2. Subroutines: Each program has been designed in subroutine block form so that facilities such as 'LOCAL', 'CHAIN' or 'OVERLAY' may be used to conserve core space for arrays. A listing index to the subroutines and their calling programs is given in Table 10.

Two versions of ORDER are provided: ORDER1 (cards 23100 - 23380) is slow in execution but short; ORDER2 (cards 37910 - 38690) is a version of 'QUICKERSORT' which is longer in object form than ORDER1, but considerably faster and preferable if core space is not at a premium.

For 1620 II users, the FORTRAN II version of subroutine DISKIO (cards 580 - 1130) can be replaced by the SPS version (cards 38700 - 40160); this saves roughly two-thirds of the CORREL, MODE and HIERAR execution times.

3. Dimensions: The declared array dimensions can be modified to extend or reduce the limits of N, MN and MB (full instructions for such modifications are given in the system operating manual, Wishart 1968a). This is a straightforward procedure, provided that the use, in COMMON, of binary overlays to equivalence fixed and floating variables is understood (see comments 270 - 290): in fact, the present dimensions have been modified for the St. Andrews IBM 1620 II to extend N to 400 individuals.

4. New similarity coefficients: 6 empty slots are provided in subroutine COEF for the definition of new similarity coefficients. The procedure can be understood most easily by studying the way in which coefficients 1 - 19 are computed in COEF (the use of arrays XMASK and MASKB for the masking facility should be carefully considered).

Further developments

CLUSTAN I currently is being extended to incorporate the following classification programs:

9. Information analysis (Lance and Williams, 1966)
10. k-dendrogram ultrametric (Jardine and Sibson, 1968).
11. Association analysis (Lance and Williams, 1965).
12. Divisive and agglomerative group analysis (Crawford and Wishart, 1967, 1968).

In addition, supplementary routines will be developed to:

1. Transform all input data types to binary and numeric.
2. Plot principal components scatter planes.
3. Plot dendrograms from the output of DENDRO.

The major facilities omitted from CLUSTAN I which will be incorporated in CLUSTAN II are:

1. The provision for 'missing' data.
2. An alphabetic sample label feature.
3. Simpler definition of new similarity coefficients (by a user function).
4. A facility for defining new cluster diag-

Table 10. - Index to subroutines and programs of CLUSTAN I.

Program	Card No.	Called by
FILE	10	-
FILEIN	1140	FILE
LOAD	3140	FILE
TRANUM	4560	FILE
EIGEN	5530	FILE
READ	see trial data	FILE
CORREL	7010	-
COREAD	8430	CORREL
ANALYS	10710	CORREL
COEF	12210	CORREL
INSERT	14040	CORREL
MODE	14710	-
INIT	16460	MODE
MINIMD	17230	MODE
RECLAS	17690	MODE
INTRO	18100	MODE
REVISE	19210	MODE
OUTPUT	20030	MODE
RESULT	22030	-
IOINP	22500	RESULT
IONUM	24370	RESULT
IOBIN	26830	RESULT
IOCOEF	27720	RESULT
IOCLUS	28580	RESULT
HIERAR	30930	-
START	31830	HIERAR
TRANS	33270	HIERAR
TFUN	35460	HIERAR
CLASS	36080	HIERAR
ALLK	37230	-
DISKIO	580	FILE, CORREL, MODE, RESULT, HIERAR,ALLK.
GET	15570	MODE, ALLK
ORDER	23100	MODE, RESULT, ALLK
IOFILE	23390	MODE, RESULT, HIERAR
DISKIO	38700	(SPS VERSION)
ORDER	37910	(QUICKERSORT VERSION)
DENDRO	36830	HIERAR

nostics (by a user function).

5. Generalization of the '10-word-per-record' disk-file format to make efficient use of disk storage on all machines, and a general review of the disk-data file layout.

6. Two input schedules: 'SIMPLE' and 'OPTIONS'. SIMPLE input will permit a casual user to enter every program with the minimum parameter specifications. The system's special features will be available under the 'OPTIONS' input alternative.

CLUSTAN II is scheduled to be defined for the IBM 360 series during 1970, and completed by 1971. The data file will be disk oriented and all programs will be written in FORTRAN.

Trial Data Schedule

Following the program listing, test results are given for three operating sequences which together form a comprehensive checking schedule. The small artificial data set used is in mixed-mode form, consisting of 9 cases with 9 binary and 6 numeric variables. Note that sequences 2 and 3 do not require the reloading of the data matrix by FILE because a valid date file can be assumed present after sequence 1.

Sequence 1 - Job order: FILE CORREL MODE RESULT

Checks:
(a) filing a mixed-mode data set, computation of product-moment correlations, standard scores and a principal components analysis (FILE).
(b) computation of a similarity matrix using coefficient 13, and assembly of 5 k-linkage lists - a total numeric mask is used (CORREL).
(c) clustering by mode analysis (MODE)
(d) printout of all data-file values, and classification diagnostics from the above MODE grouping (RESULT).

Notes:
(i) The input data listing obtained from FILE is produced by a PRINT statement in subroutine READ. This helps data errors to be detected.
(ii) Each program after FILE prints an identification heading which details the state of the data file. Because MODE does not alter the data file, this heading for RESULT indicates that the similarity matrix is yet filed.

Sequence 2 - Job order: CORREL MODE HIERAR RESULT

Checks:
(a) the computation of distance coefficients from the first 4 factor scores (derived from the principal components analysis), and assembly of 5 k-linkage lists (CORREL).
(b) clustering by mode analysis (MODE)
(c) clustering by Lance-Williams flexible method - last 3 groupings requested (HIERAR).
(d) printout of cluster diagnostics for one of the groupings punched by HIERAR (RESULT).

Notes:
(i) From these data, MODE produces no groupings suggesting that no 'natural' classification exists.
(ii) The output from HIERAR first details the fusion sequence, then prints the base numbering of the dendrogram. The branches of this diagram have been completed roughly by hand (stem unions have not been drawn according to the coefficient scale).
(iii) Because HIERAR destroys the coefficients matrix, the subsequent data-file summary given by RESULT states that no coefficients are stored. This does not, however, inhibit the calculation of cluster diagnostics from the remainder of the data file.
(iv) The printout of raw data-file values and the 'binary variable frequences' of the cluster diagnostics has been inhibited by the nonselection of the options of the RESULT parameter card.

Sequence 3 - Job order: CORREL RESULT

Checks:
(a) the computation of numeric coefficient 3 from a partial mask of the numeric data file (CORREL).
(b) the inhibit-selection features of the results output (RESULT).

Notes:
(i) Numeric variables 4 and 6 are masked, together with the entire binary data file, from the computation of product-moment correlation coefficients by CORREL. The value of MSKB is 9, and MSKN = 2. Codes 4 and 6 appear on the partial mask card which is input to CORREL.
(ii) The data file identification by RESULT prints a list of the numeric variables which were not masked. Only the output options for the similarity matrix and k-linkage lists are selected for RESULT to check that all other PRINT options have been inhibited.

COMPARISON OF CLUSTERING METHODS

Data were collected from an areal geology map, given in the Hollidaysburg - Huntingdon (Pennsylvania) Folio (Butts, 1945), which shows the distribution at land surface of 48 igneous, sedimentary and metamorphic rock units in a region of about 430 square miles. Figure 1 indicate the location of major units, and provides a rough representation of the region's topography. The map was divided gridwise into 176 4 - cm.square units, and the extent of each formation was estimated visually, from the mapped colored sections, as a score out of 10 for each unit (these data are listed immediately after the program). A key to the formation codes is given in Table 11, together with the overall percentage extent of these units as estimated.

Table 11. - Key to formation codes used in Table 12, and measured from areal geology map (Butts, 1945).

Rock Code	Symbol	Formation	Total Percentage Cover
1	QAL	ALLUVIUM	5.9
2	CA	ALLEGHENY FORMATION	0.1
3	CPV	PUTTSVILLE FORMATION	0.8
4	CMC	MAUCH CHUNK FORMATION - TOP LAYER	5.5
5	CTC	MAUCH CHUNK FORMATION - BOTTOM LAYER	0.1
6	CB	POCONO FORMATION - TOP LAYER	6.7
7	CPO	POCONO FORMATION - BOTTOM LAYER	0.8
8	DHA	HAMPSHIRE FORMATION	4.1
9	DCC	CHEMUNG FORMATION - CONGLOMERATE LENTILS	0.1
10	DSX	CHEMUNG FORMATION - SAXTON CONGLOMERATE MEMBER	0.3
11	DA	CHEMUNG FORMATION - ALLEGRIPPIS SANDSTONE MEMBER	0.1
12	DP	CHEMUNG FORMATION - PINEY RIDGE SANDSTONE MEMBER	0.1
13	DCH	CHEMUNG FORMATION	9.4
14	DB	BRALLIER SHALE	6.6
15	DHR	HARRELL SHALE - UPPER LAYER	0.4
16	DH	HAMILTON FORMATION	3.7
17	DM	MARCELLUS SHALE	1.1
18	DO	ONONDAGA FORMATION	0.5
19	DR	RIDGELEY SANDSTONE	1.4
20	DS	SHRIVER LIMESTONE	1.2
21	DHB	HELDERBERG LIMESTONE	0.9
22	STW	TONOLOWAY LIMESTONE	3.7
23	SWC	WILLS CREEK SHALE	3.1
24	SB	BLOOMSBURG REDBEDS	1.1
25	SMK	MCKENZIE FORMATION	1.8
26	SK	CLINTON FORMATION - LAYER NEAR TOP	0.2
27	SC	CLINTON FORMATION -	5.7
28	SCS	CLINTON FORMATION - BOTTOM LAYER	0.1
29	ST	TUSCARORA QUARTZITE	3.4
30	OJ	JUNIATA FORMATION	2.8
31	OO	OSWEGO SANDSTONE	2.3
32	ORV	REEDSVILLE SHALE	3.5
33	OT	TRENTON LIMESTONE	1.1
34	OR	RODMAN LIMESTONE	0.2
35	OL	LOWVILLE LIMESTONE	0.7
36	OC	CARLIM LIMESTONE	0.6
37	OB	BELLEFONTE DOLOMITE	6.1
38	OA	AXEMANN LIMESTONE	0.4
39	ON	NITTANY DOLOMITE	4.7
40	OLA	LARKE DOLOMITE	1.0
41	OM	MINES DOLOMITE	1.0
42	EO	GATESBURG FORMATION - MIDDLE LAYER	0.1
43	EG	GATESBURG FORMATION	3.9
44	ES	GATESBURG FORMATION - BOTTOM LAYER	0.6
45	EW	WARRIOR LIMESTONE	0.6
46	EPH	PLEASANT HILL LIMESTONE	0.1
47	DBK	HARRELL SHALE - LOWER PART	0.2
48	EWB	WAYNESBORO FORMATION	0.1

The object of the exercise was to classify the units into groups using each of the 8 clustering methods, reconstruct a mapping of the groups to obtain a simplified representation of the areal geology, and provide a visual means of comparing the methods. Standardization of the variable values (rock unit percentages) seemed unwise in this instance, because this would tend to promote the small percentages of the thin formations to relatively high values. Consequently, the distance components of such variables would be out of proportion to their importance. It was decided, therefore, to use a similarity matrix of Euclidean distances (code 1) computed from the raw data values, and the units were classified using KL = 3 for mode analysis, and BETA = -.25 for the flexible option of HIERAR.

The results of the methods nearest neighbor, median, group average and centroid exhibited the

Table 12. - Rock distributions for each cluster derived by mode analysis (Mode 1-12), farthest neighbor (FN 1-10), Ward's error sum method (W 1-10) and Lance-Williams flexible (LW 1-10) for areal geology map data.

CLUSTER METHOD	CLUSTER CODE	CLUSTER SIZE	MAIN CONSTITUENT ROCKS (NUMBERS ARE ROCK CODES - SEE TABLE 11 - WITH PERCENTAGE COVER IN BRACKETS. ROCKS WITH LESS THAN 4% COVER ARE OMITTED)	PERCENTAGE OTHER ROCKS
MODE	1	15	4(60.2) 6(20.7) 1(5.1) 8(4.2)	9.8
MODE	2	9	6(86.4) 4(4.6) 8(4.6)	4.4
MODE	3	18	13(72.7) 8(9.2) 14(8.7)	9.4
MODE	4	14	39(36.7) 37(27.4) 43(5.1) 40(4.9) 32(4.5)	21.4
MODE	5	8	8(56.4) 6(12.0) 13(11.8) 7(10.0) 1(7.8)	2.0
MODE	6	14	14(50.3) 16(14.4) 1(12.1)	23.2
MODE	7	35	32(15.0) 30(12.8) 29(12.5) 31(10.9) 27(9.9) (37)8.1 33(4.0)	26.8
MODE	8	23	22(18.4) 1(15.9) 16(13.1) 23(10.2) 19(8. 7) (20)6.0 17(5.9) 21(4.3)	17.5
MODE	9	13	43(43.8) 39(18.2) 41(8.4) 44(7.5) 40(7.0) 45(5.4)	9.7
MODE	10	15	27(37.1) 25(12.3) 23(11.3) 29(8.7) 22(7.7) 24(5.5)	17.4
MODE	11	6	37(61.3) 43(5.0) 36(4.5) 1(4.3) 45(4.2)	20.7
MODE	12	6	14(37.2) 13(32.5) 16(13.0) 1(6.7)	10.6
FN	1	15	SAME AS MODE 1	9.8
FN	2	9	SAME AS MODE 2	4.4
FN	3	17	13(74.6) 14(9.2) 8(6.2) 1(4.1)	5.9
FN	4	23	37(39.6) 39(13.2) 32(10.6) 33(5.0)	31.6
FN	5	9	8(56.8) 13(14.9) 6(10.7) 7(9.0) 1(6.9)	1.7
FN	6	17	14(50.1) 16(13.5) 13(12.6) 1(10.7)	13.1
FN	7	77	33(9.4) 22(8.0) 43(7.9) 29(6.8) 23(6.7) 1(6.6) 30(5.8) 16(5.3)	43.5
FN	8	1	3(77.0) 4(23.0)	0.0
FN	9	4	39(63.8) 37(17.0) 38(5.3)	13.9
FN	10	4	27(65.0) 29(11.8) 1(6.0) 23(4.0) 25(4.0)	9.2
W	1	14	4(62.2) 6(18.9) 3(9.6) 1.(4.4)	4.9
W	2	8	6(90.6) 4(5.1)	4.0
W	3	16	13(76.1) 14(8.7) 8(6.6) 1(4.2)	4.4
W	4	23	SAME AS FN 4	31.6
W	5	12	8(49.3) 6(16.2) 13(12.2) 7(9.4) 1(6.5) 4(4.8)	1.6
W	6	18	14(48.3) 13(14.7) 16(13.4) 1(10.2)	13.4
W	7	40	27(22.1) 29(13.6) 30(10.7) 32(8.5) 31(8.2) 23(6.8) 25(6.2) 22(4.4)	19.5
W	8	27	22(16.7) 1(15.1) 16(13.2) 23(9.7) 19(8.1) 14(5.6) 20(5.5)	26.1
W	9	14	43(42.5) 39(17.9) 41(8.4) 40(7.2) 44(6.9) 45(5.0)	12.1
W	10	4	SAME AS FN 9	13.9
LW	1	13	4(65.2) 6(20.4) 1(4.7) 3(4.4)	8.0
LW	2	8	SAME AS W 2	4.0
LW	3	17	SAME AS FN 3	5.9
LW	4	27	37(36.2) 39(20.7) 32(9.3) 33(4.4)	29.4
LW	5	12	SAME AS W 5	1.6
LW	6	17	SAME AS FN 6	13.1
LW	7	41	27(22.3) 29(13.3) 30(10.4) 32(8.3) 31(8.0) 23(6.8) 25(6.2) 22(4.4)	20.3
LW	8	26	22(17.0) 1(14.2) 16(13.7) 23(9.8) 19(8.4) 14(5.8) 17(5.8) 20(5.6)	19.7
LW	9	14	SAME AS W 9	12.1
LW	10	1	SAME AS FN 8	0.0

'chaining effect', that is, the fusion hierarchy tended to clump individual units successively into one universal group (for a discussion of chaining, see Williams, Lambert and Lance, 1966; or Wishart, 1968b). For this reason, these methods are omitted in the analysis of performance. From the remaining analyses, the 10 cluster level of fusion was chosen for the comparison of the hierarchical procedures using farthest neighbor, Ward's error sum and the Lance-Williams flexible method. Mode analysis produced 16 different groupings, of which the 5th contained the maximum of 12 clusters, and is considered here because this widest separation of the measurement units may be treated as the most general classification possible. The dominant characteristics of each cluster, together with their sizes, are set out in Table 12, and in Figures 2 - 5 the cluster distributions are mapped onto the original measurement grid using distinctive shading to demark each region. It is apparent immediately, from a comparison of Figures 4 and 5 with reference to Table 12, that the groupings of Ward's method and the Lance-Williams flexible method are practically identical. In fact, these two methods differ in their allocation of only 7 of the 176 measurement units. The distinction, by Ward's method, between cluster 10 and cluster 4 does not seem particularly useful despite being consistent with the results of farthest neighbor. For this reason, the flexible method seems marginally preferable. Farthest neighbor tended to oversimplify the geology of cluster 7, and this criticism is substantiated by the relatively low percentage extent attributed to the dominant units in this cluster, as shown in Table 12. By contrast, mode analysis produced rather too complex a general structure - we can expect clustered units to be reasonably contiguous by virtue of the nature of stratification and sedimentation, but although Figure 2 repeats the basic patterns common to all the maps, the number of regional discontinuities is markedly higher (this might, however, be due partly to the extra two clusters present in the MODE grouping).

For the purpose of finding a classification tool which will produce simplified patterns of areal geology from sampled data, the methods discussed here can be provisionally rated in the following preferential order:

1. Lance-Williams flexible method
2. Ward's method
3. Mode analysis
4. Farthest neighbor

The conclusion that can be drawn from this experiment could probably have been stated at the outset, namely, that in isolating regions of uniform geology the classification method should search for groups of units which possess overall uniformity or lack of variation. Excluding mode analysis, the other three methods are of the 'minimum-variance' type and have been found to succeed in similar applications, noticeably ecology. The relative success of mode analysis, which is normally used to detect the presence (or absence) of 'natural' classes and can be considered out of context here, can be attributed to the contrasting topographic features of the area which are associated with distinctive geologic formations.

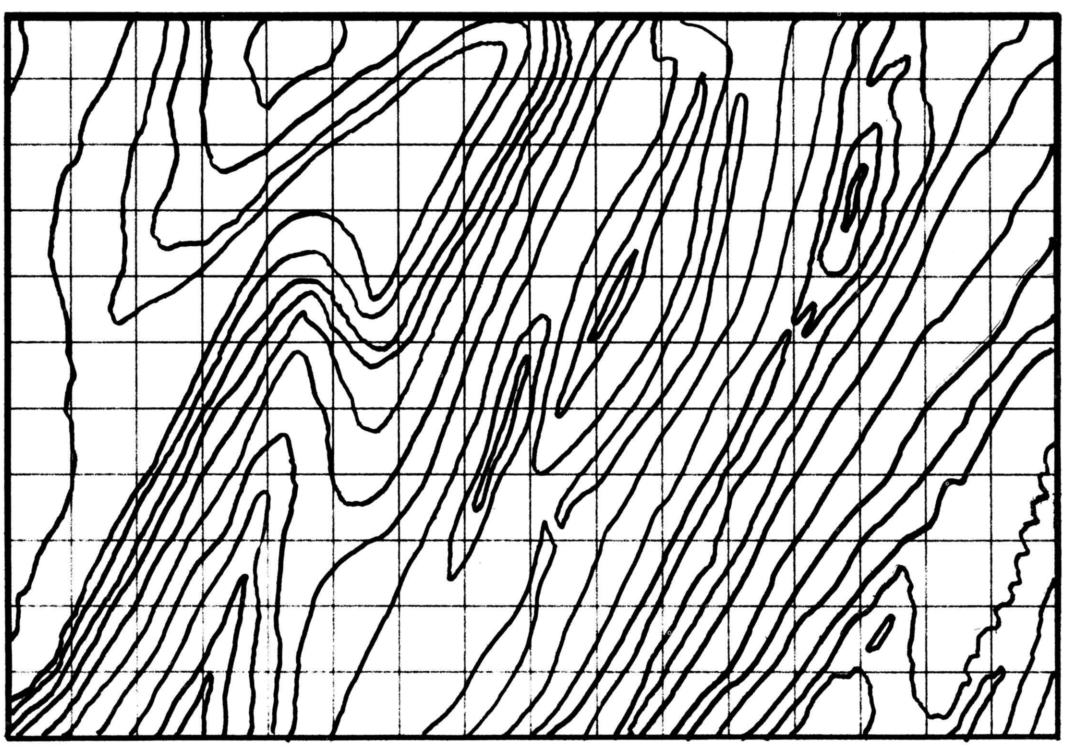


Figure 1. - Line drawing showing separation of strata on areal geology map (Butts, 1945)

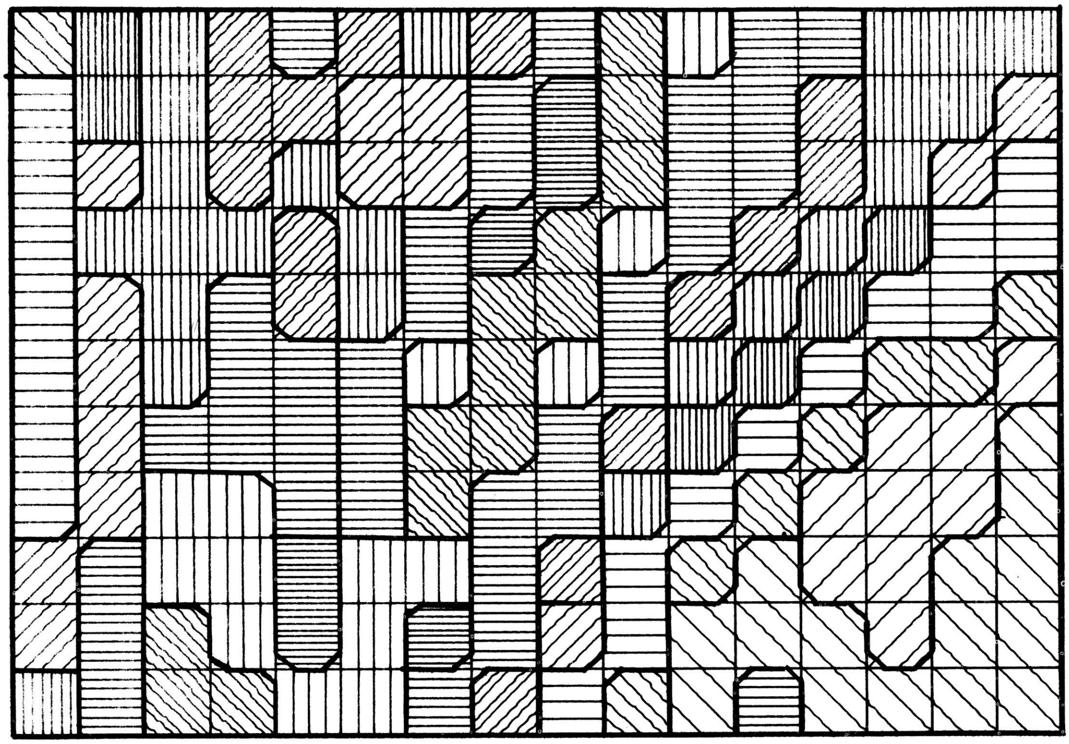
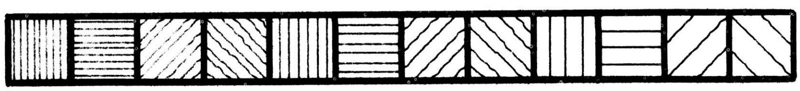


Figure 2. - Classification into 12 clusters by mode analysis.

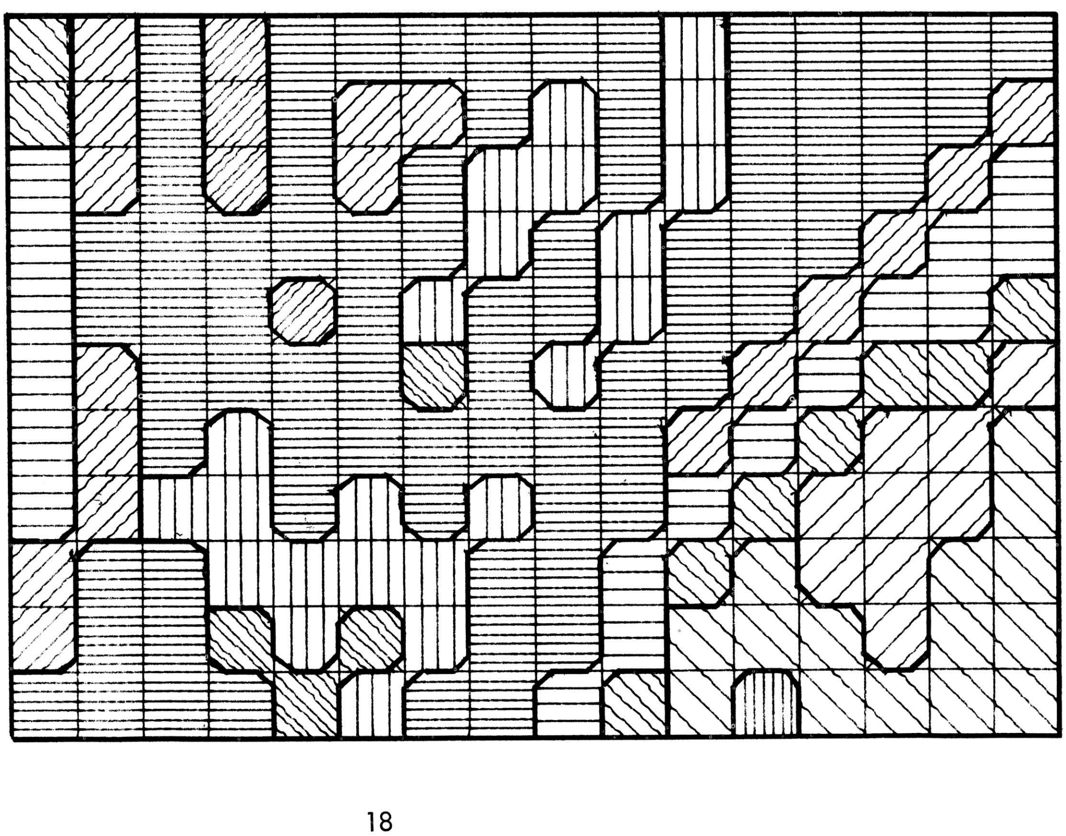
Type of shading



Cluster Number: 12 11 10 9 8 7 6 5 4 3 2 1

Key to shaded regions of Figure 2 - 5.

Figure 3. - Classification into 10 clusters by farthest neighbor.



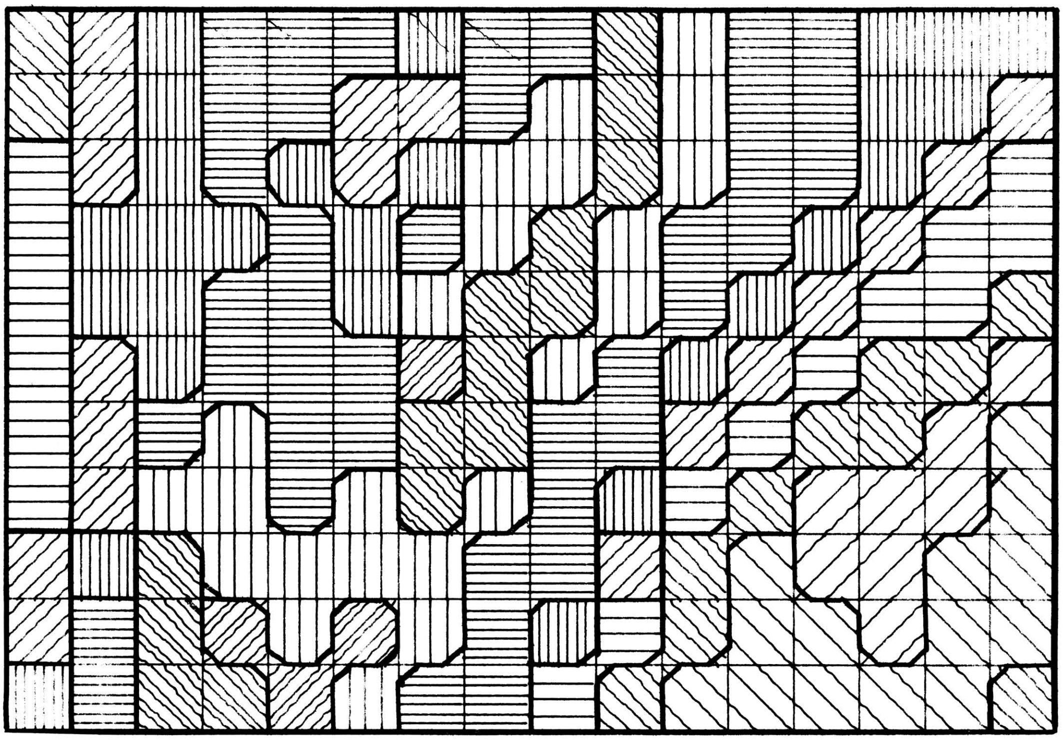


Figure 4. - Classification into 10 clusters by Ward's error sum method.

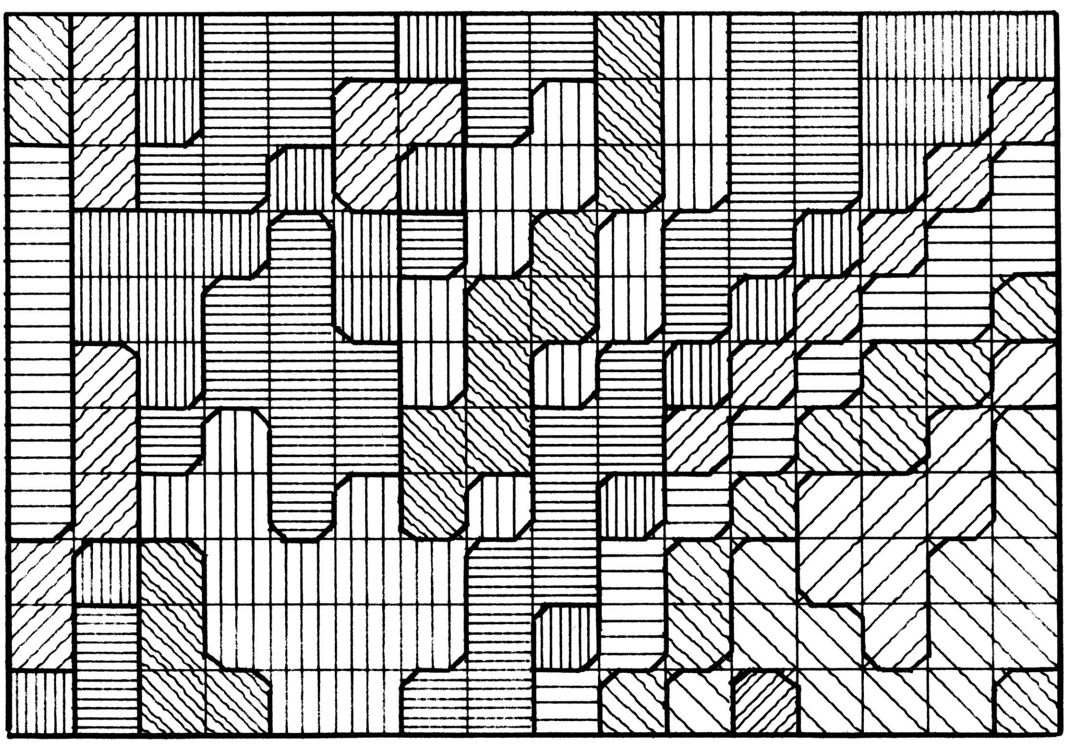


Figure 5. - Classification into 10 clusters by Lance-Williams' flexible method.

REFERENCES

Butts, C., 1945, Description of the Hollidaysburg and Huntingdon Quadrangles (Pa.): U.S. Geological Survey, Geol. Atlas of U.S., Hollidaysburg - Huntingdon Folio No. 227 (Pennsylvania), 20 p.

Crawford, R.M.M., and Wishart, D., 1967, A rapid multivariate method for the detection and classification of groups of ecologically related species: Jour. Ecology, v. 55, no. 2, p. 505.

Crawford, R.M.M., and Wishart, D., 1968, A rapid classification and ordination method and its application to vegetation mapping: Jour. Ecology, v. 56, no. 2, p. 385.

Forgey, E.W., 1964, Evaluation of several methods for detecting sample mixtures from different N-dimensional populations: Amer. Psychol. Assoc., Los Angeles, Calif.

Forgey, E.W., 1965, Cluster analysis of multivariate data - efficiency versus interpretability of classifications: AAAS - Biometric Soc. (WNAR), Riverside, Calif.

Jardine, N., and Sibson, R., 1968, The construction of hierarchic and non-hierarchic classifications: Computer Jour., v. 11, no. 2, p. 177-184.

Lance, G.N., and Williams, W.T., 1965, Computer programs for monothetic classification (association analysis): Computer Jour., v. 8, no. 3, p. 246-249.

Lance, G.N., and Williams, W.T., 1966, Computer programs for hierarchical polythetic classification (similarity analyses): Computer Jour., v. 9, no. 1, p. 60-64.

Lance, G.N., and Williams, W.T., 1967, A general theory of classificatory sorting strategies 1. Hierarchical systems: Computer Jour., v. 9, no. 4, p. 373-380.

Ward, J.H., 1963, Hierarchical grouping to optimize an objective function, Jour. Amer. Statist. Assoc., v. 58, p. 236-244.

Williams, W.T., and Lambert, J.M., and Lance, G.N., 1966, Multivariate methods in plant ecology: Jour. Ecology, v. 54, no. 2, p. 427.

Wishart, D., 1968a, A FORTRAN II programme for numerical classification (system operating manual - CLUSTAN I): Univ. of St. Andrews, Computing Laboratory, St. Andrews,(Scotland),72 p.

Wishart, D., 1968b, Mode analysis - a generalization of nearest neighbor which reduces chaining effect: Univ. of St. Andrews Colloquium in Numerical Taxonomy, Computing Laboratory, St. Andrews (Scotland), p. 233-254.

Wishart, D., 1968c, A numerical classification method for deriving natural classes: Nature, v. 221, no. 5175, p. 97-98.

Wishart, D., 1969, An algorithm for hierarchical classifications: Biometrics, v. 22, no. 1, p. 165-170.

CLUSTAN IA

The extensions of the CLUSTAN I package referred to in the text of this paper are incorporated in a second release called CLUSTAN IA which will be available shortly. CLUSTAN IA also includes the methods of iterative relocation, monothetic division and centroid-forming hierarchic fusion in conjunction with one of 13 standard measures of intercluster similarity, and Calinski's minimum spanning tree method of optimizing the error sum of squares. A USER function is supplied with which all CLUSTAN methods can be programmed for use with a new similarity coefficient.

The CLUSTAN IA programs are written in FORTRAN IV for the IBM System 360/ Model 44 to be consistent with the future policy of Computer Contributions (see Editor's Remarks), and it is therefore recommended that the FORTRAN IV version of CLUSTAN I should be requested.

PROGRAM, GEOLOGIC DATA, AND TRIAL DATA SCHEDULE LISTING, AND TRIAL DATA RESULTS.

```
C   ROUTINE - FILE
C   AUTHOR - DAVID WISHART, MATHEMATICAL INSTITUTE, UNIVERSITY OF ST.
C   ANDREWS, FIFE, GREAT BRITAIN.  (PROGRAMME WRITTEN - JUNE 1968)
      DEFINE DISK (10,4300)
C   MAIN PROGRAMME TO FILE DATA DECK AND COMPUTE BASIC STATISTICS
C    FUNCTIONS  (A) NUMERIC DATA
C                   MEANS, STANDARD DEVIATIONS, VARIANCES, PRODUCT-MOMENT
C                   CORRELATIONS AND STANDARD SCORES
C                   PRINCIPAL COMPONENTS (FACTOR) ANALYSIS - EIGENVALUES,
C                   EIGENVECTORS, PERCENTAGE AND CUMULATIVE VARIANCE, FACTOR
C                   SCORES
C    FUNCTIONS  (B) BINARY DATA
C                   VARIABLE FREQUENCIES, PERCENTAGE OCCURRENCES
C   CALLS SUBROUTINES - TRANUM,DISKIO,FILEIN,LOAD,EIGEN AND (USER
C   SUBROUTINE - RECOMPILED FOR EACH DATA DECK) READ
C   NOTE - OUTPUT OF STATISTICS IS OBTAINED FROM ROUTINE (RESULT), (FILE)
C   MERELY LOADS THESE VALUES TO DISK FILE
C   DIMENSIONS AND COMMON AREAS FOR ALL FILE ROUTINES ******************
C   END OF COMMON RESERVED FOR DISK I/O SUBROUTINE (DISKIO) PARAMETERS
      DIMENSION TEXT(20)
      COMMON N,MB,MN,NPCF,NPC,ISTAND,IMASK,IDATA,ICOEF,ITYPE,KMAX
      COMMON LNDATA,LBDATA,LMEANS,LVARS,LCORS,LEIGS,LEIGVS,LSCORS,LENGS,
     1LFREQS,LNMASK,LBMASK,LMAT,LKLIST,LNEXT,TEXT
C   LIMIT ON MN = 200 ( NUMBER OF NUMERIC VARIABLES )
C   LIMIT ON MB = 401 (NUMBER OF BINARY VARIABLES)
C   LIMIT ON N = 250 ( NUMBER OF POINTS/CASES )
C   THE FOLLOWING DIMENSION, COMMON AND EQUIVALENCE STATEMENTS PERMIT
C   REAL AND INTEGER ARRAYS TO BE EQUIVALENCED (MOSTLY ILLEGAL IN 1620
C   FORTRAN II). THE REQUIRED LENGTHS OF THE INTEGER ARRAYS CONCERNED ARE
C   LIST(401), IFREQ(401), LENG(250). THE EQUIVALENCE WORKS ON THE
C   PRINCIPLE THAT 5 STORAGE LOCATIONS ARE REQUIRED FOR EACH INTEGER, AND
C   10 LOCATIONS FOR EACH REAL WORD - THUS A REAL WORD CAN SHARE SPACE
C   WITH TWO INTEGERS.
      DIMENSION COR(630),XM(200),VAR(200),X(200),Y(200),EIG(35)
     1,EIGV(1225),LIST(1),IFREQ(1),LENG(1)
      COMMON COR,XM,VAR,X,Y,EIG,EIGV,LIST
      EQUIVALENCE (LIST(402),IFREQ),(LIST(803),LENG)
C   HENCE LENG, LIST AND IFREQ ALL WRITE OVER THE ARRAY EIGV
C   SINGLE VARIABLE COMMON
      COMMON ID,IPRC,JEIGV,ICOR,JCOR
C   DIMENSIONS AND COMMON AREAS FOR ALL FILE ROUTINES ******************
C   INITIALISE DISK PARAMETERS, READ AND ANALYSE INPUT PARAMETER CARD
      CALL FILEIN
C   READ AND LOAD RAW DATA, NUMERIC MEANS, VARIANCES AND CORRELATIONS -
C   BINARY FREQUENCIES AND SAMPLE LIST LENGTHS
      CALL LOAD
C   TRANSFORM NUMERIC DATA TO STANDARD SCORES, COMPONENT SCORES, AND
C   COMPUTE AND STORE EIGENVALUES AND EIGENVECTORS
      CALL TRANUM
C   LOAD FILE PARAMETERS TO DISK
      CALL DISKIO(2,LDUMY,IDUMY,IBDUMY,XDUMY,LNDUMY)
C   PRINT ENDING MESSAGE
      PRINT 5
    5 FORMAT (14H FILE COMPLETE/9H JOB ENDS/1H1)
      CALL EXIT
C   DUMMY CALL STATEMENTS TO ENABLE CORE LOADING OF (READ) AND (TRANUM)
C   WHEN (FILEIN),(LOAD) AND (EIGEN)) ARE IN LOCAL (IBM 1620 MONITOR II)
   10 CALL READ
      CALL EIGEN
      P=ABSF(P)
```

```
      END

      SUBROUTINE DISKIO (ISEL,LSEC,IBIN,LB,XAR,LN)
C     FUNCTION - TO PERFORM ALL DISK INPUT/OUTPUT OPERATIONS FOR ROUTINES
C     (FILE),(CORREL),(MODE) AND (RESULT)
C     DIMENSIONS AND COMMON FOR DISK INPUT/OUTPUT (DISKIO) ****************
C     END OF COMMON RESERVED FOR DISK I/O SUBROUTINE (DISKIO) PARAMETERS
      DIMENSION TEXT(20)
      COMMON N,MB,MN,NPCF,NPC,ISTAND,IMASK,IDATA,ICOEF,ITYPE,KMAX
      COMMON LNDATA,LBDATA,LMEANS,LVARS,LCORS,LEIGS,LEIGVS,LSCORS,LENGS,
     1LFREQS,LNMASK,LBMASK,LMAT,LKLIST,LNEXT,TEXT
C     DIMENSIONS AND COMMON FOR DISK INPUT/OUTPUT (DISKIO) ****************
C     DUMMY DIMENSION STATEMENT
      DIMENSION IBIN(1),XAR(1)
C     ISEL = INPUT/OUTPUT OPERATION SELECTOR
C     LSEC = DISK SECTOR ADDRESS FOR THAT OPERATION
C     BRANCH ON OPERATION SELECTOR
      GO TO (5,10,15,20,25,30),ISEL
C     ISEL=1, READ DISK FILE PARAMETERS
    5 FETCH (1) N,MB,MN,NPCF,NPC,ISTAND,IMASK,IDATA,ICOEF,ITYPE,KMAX
      FETCH (3) LNDATA,LBDATA,LMEANS,LVARS,LCORS,LEIGS,LEIGVS,LSCORS,
     1LENGS,LFREQS,LNMASK,LBMASK,LMAT,LKLIST,LNEXT
      FETCH (6) (TEXT(I),I=1,20)
      RETURN
C     ISEL=2, WRITE DISK FILE PARAMETERS
   10 RECORD (1) N,MB,MN,NPCF,NPC,ISTAND,IMASK,IDATA,ICOEF,ITYPE,KMAX
      RECORD (3) LNDATA,LBDATA,LMEANS,LVARS,LCORS,LEIGS,LEIGVS,LSCORS,
     1LENGS,LFREQS,LNMASK,LBMASK,LMAT,LKLIST,LNEXT
      RETURN
C
C     TRANSLATION NOTES
C     THE NEXT FOUR FETCH AND RECORD STATEMENTS OPERATE AS FOLLOWS -
C     THE ARRAY IS SPLIT INTO 10 WORD BLOCKS, AND EACH BLOCK IS WRITTEN TO
C     ONE DISK RECORD
C     THE RECORD NUMBER (LSEC) IS AUTOMATICALLY INCREMENTED BY THE NUMBER
C     OF BLOCKS OCCUPIED (INCLUDING THE LAST RECORD IF PARTIALLY OCCUPIED)
C     EXAMPLE
C     FETCH (LSEC) (IBIN(I),I=1,LB) WITH LSEC=41 AND LB=71 READS 71
C     ELEMENTS OF IBIN, AT 10 PER RECORD, STARTING AT RECORD 41 AND
C     RETURNS THE VALUE LSEC=49 (SINCE THE ARRAY BLOCK OCCUPIES 8 RECORDS)
C     THUS MAINLINE STATEMENTS SUCH AS (LBDATA = N*(MN+9)/10+8) OPERATE ON
C     THE PRINCIPLE THAT (MN+9)/10 IS THE NUMBER OF RECORDS REQUIRED TO
C     STORE ONE NUMERIC DATA SAMPLE, AND N*(MN+9)/10 IS THEREFORE THE
C     TOTAL DISK STORAGE COMMITMENT FOR THE ENTIRE RAW NUMERIC DATA FILE
C
C     ISEL=3, WRITE A BINARY SAMPLE LIST ARRAY (IBIN(I),I=1,LB)
   15 RECORD (LSEC) (IBIN(I),I=1,LB)
      RETURN
C     ISEL=4, READ A BINARY SAMPLE LIST ARRAY (IBIN(I),I=1,LB)
   20 FETCH (LSEC) (IBIN(I),I=1,LB)
      RETURN
C     ISEL=5, WRITE A NUMERIC ARRAY (XAR(I),I=1,LN)
   25 RECORD (LSEC) (XAR(I),I=1,LN)
      RETURN
C     ISEL=6, READ A NUMERIC ARRAY (XAR(I),I=1,LN)
   30 FETCH (LSEC) (XAR(I),I=1,LN)
      RETURN
      END
```

```
      SUBROUTINE FILEIN
C   CALLED BY (FILE)
C   FUNCTION - TO INITIALISE DISK STORAGE PARAMETERS, READ AND ANALYSE
C   TEXTUAL HEADING AND CARD JOB PARAMETERS AND SET UP CONTROL
C   INTEGERS FOR (FILE)
C   DIMENSIONS AND COMMON AREAS FOR ALL FILE ROUTINES ******************
C   END OF COMMON RESERVED FOR DISK I/O SUBROUTINE (DISKIO) PARAMETERS
      DIMENSION TEXT(20)
      COMMON N,MB,MN,NPCF,NPC,ISTAND,IMASK,IDATA,ICOEF,ITYPE,KMAX
      COMMON LNDATA,LBDATA,LMEANS,LVARS,LCORS,LEIGS,LEIGVS,LSCORS,LENGS,
     1LFREQS,LNMASK,LBMASK,LMAT,LKLIST,LNEXT,TEXT
C   LIMIT ON MN = 200 ( NUMBER OF NUMERIC VARIABLES )
C   LIMIT ON MB = 401 (NUMBER OF BINARY VARIABLES)
C   LIMIT ON N = 250 ( NUMBER OF POINTS/CASES )
C   THE FOLLOWING DIMENSION, COMMON AND EQUIVALENCE STATEMENTS PERMIT
C   REAL AND INTEGER ARRAYS TO BE EQUIVALENCED (MOSTLY ILLEGAL IN 1620
C   FORTRAN II). THE REQUIRED LENGTHS OF THE INTEGER ARRAYS CONCERNED ARE
C   LIST(401), IFREQ(401), LENG(250). THE EQUIVALENCE WORKS ON THE
C   PRINCIPLE THAT 5 STORAGE LOCATIONS ARE REQUIRED FOR EACH INTEGER, AND
C   10 LOCATIONS FOR EACH REAL WORD - THUS A REAL WORD CAN SHARE SPACE
C   WITH TWO INTEGERS.
      DIMENSION COR(630),XM(200),VAR(200),X(200),Y(200),EIG(35)
     1,EIGV(1225),LIST(1),IFREQ(1),LENG(1)
      COMMON COR,XM,VAR,X,Y,EIG,EIGV,LIST
      EQUIVALENCE (LIST(402),IFREQ),(LIST(803),LENG)
C   HENCE LENG, LIST AND IFREQ ALL WRITE OVER THE ARRAY EIGV
C   SINGLE VARIABLE COMMON
      COMMON ID,IPRC,JEIGV,ICOR,JCOR
C   DIMENSIONS AND COMMON AREAS FOR ALL FILE ROUTINES ******************
C   READ HEADER JOB SPECIFICATION
      READ 5,(TEXT(I),I=1,20)
    5 FORMAT (20A4)
C   RECORD HEADER JOB SPECIFICATION
      IREC=6
      CALL DISKIO(5,IREC,IBDUMY,LBDUMY,TEXT,20)
C   READ CARD PARAMETERS
      READ 10,N,MN,MB,ISTAND,JCOR,NPCF,JEIGV
   10 FORMAT (3I4,A1,A1,I2,A1)
C   PRINT HEADING
      PRINT 15,(TEXT(I),I=1,20)
   15 FORMAT (1X,20A4//)
      PRINT 20,N,MB,MN
   20 FORMAT (19H NUMBER OF CASES = ,I4/30H NUMBER OF BINARY VARIABLES =
     1 ,I4,/31H NUMBER OF NUMERIC VARIABLES = ,I4//17H FILE DATA MATRIX/
     1/)
C   INITIALISE DISK FILE PARAMETERS TO NULL
      NPC=0
      IMASK=0
      IDATA=0
      ICOEF=0
      ITYPE=0
      KMAX=0
      LNDATA=0
      LBDATA=0
      LMEANS=0
      LVARS=0
      LCORS=0
      LEIGS=0
      LEIGVS=0
```

```
      LSCORS=0
      LENGS=0
      LFREQS=0
      LNMASK=0
      LBMASK=0
       LMAT=0
      LKLIST=0
      LNEXT=0
C   CHECK VALUES OF N,MN,MB
      IF (N-250)45,45,25
C   ERROR - N EXCEEDS 250
   25 PRINT 30
   30 FORMAT (44H ERROR - NUMBER OF CASES MUST NOT EXCEED 250)
C   ENTER ABORT ROUTINE
   35 PRINT 40
      CALL DISKIO(2,LDUMY,IDUMY,IBDUMY,XDUMY,LNDUMY)
   40 FORMAT (17H PHASE TERMINATED/9H JOB ENDS/1H1)
      CALL EXIT
   45 IF (MN-200)60,60,50
C   NUMBER OF NUMERIC VARIABLES EXCEEDS 200
   50 PRINT 55
   55 FORMAT (48H ERROR - NUMBER OF NUMERIC VARIABLES EXCEEDS 200)
      GO TO 35
   60 IF (MB-401)75,75,65
C   NUMBER OF BINARY VARIABLES EXCEEDS 401
   65 PRINT 70
   70 FORMAT (47H ERROR - NUMBER OF BINARY VARIABLES EXCEEDS 401)
      GO TO 35
C   INPUT DIMENSIONS ARE OK
C   IS PRINCIPAL COMPONENTS SOLUTION REQUIRED
   75 IF (NPCF)80,80,90
C   NO PRINCIPAL COMPONENTS SOLUTION SELECTED
C   TEST IF CORRELATION MATRIX REQUIRED
   80 IF (JCOR)125,125,85
C   CORRELATIONS REQUIRED
C   SET PRINCIPAL COMPONENTS INDICATOR OFF
   85 IPRC=1
C   GO TO CHECK DIMENSION MN - GREATER THAN ZERO, LESS THAN 35
      GO TO 94
C   PRINCIPAL COMPONENTS SOLUTION SELECTED - DOES NPCF EXCEED MN
   90 IF (NPCF-MN)93,93,91
C   NPCF EXCEEDS MN - PRINT ERROR
   91 PRINT 92
   92 FORMAT (77H ERROR - NUMBER OF COMPONENT SCORES CANNOT EXCEED NUMBE
     1R OF NUMERIC VARIABLES)
      NPCF=MN
C   SET COMPONENTS INDICATOR ON
   93 IPRC=2
C   ARE NUMERIC VARIABLES SUPPLIED
   94 IF (MN)95,95,105
C   NO NUMERIC VARIABLES SUPPLIED
   95 PRINT 100
  100 FORMAT (95H ERROR - PRINCIPAL COMPONENTS SOLUTION OR CORRELATIONS
     1ARE AVAILABLE FOR NUMERIC VARIABLES ONLY)
      GO TO 125
C   NUMERIC VARIABLES SUPPLIED - TEST IF MN EXCEEDS 35
  105 IF (MN-35)120,120,110
C   MORE THAN 35 VARIABLES
  110 PRINT 115
  115 FORMAT (84H ERROR - LIMIT ON NUMBER OF VARIABLES FOR PRINCIPAL COM
```

```
     1PONENTS OR CORRELATIONS IS 35)
      GO TO 125
C  VARIABLE TESTS OK - SET CORRELATION INDICATOR
  120 ICOR=2
C  GO TO DATA-MODE ANALYSIS
      GO TO 130
C  NO PRINCIPAL COMPONENTS SOLUTION OR CORRELATIONS SELECTED - OR ERROR
C  PHASE
C  SET INDICATORS TO NULL COMPONENT SCORES AND CORRELATIONS
  125 ICOR=1
      IPRC=1
      NPCF=0
C  ENTER DATA-MODE ANALYSIS. SET INDICATOR TO ZERO
  130 ID=0
C  ARE NUMERIC VARIABLES SUPPLIED
      IF (MN)140,140,135
C  NUMERIC VARIABLES SUPPLIED
  135 ID=1
C  ARE BINARY VARIABLES SUPPLIED
  140 IF (MB)150,150,145
C  BINARY VARIABLES SUPPLIED
  145 ID=ID+2
C  CHECK FOR ANY VARIABLES
  150 IF (ID)155,155,165
C  NO VARIABLES SUPPLIED - PRINT ERROR
  155 PRINT 160
  160 FORMAT (41H ERROR - MUST DECLARE NUMBER OF VARIABLES)
      GO TO 35
C  BRANCH ON MODE INDICATOR
  165 GO TO (170,180,190),ID
C  NUMERIC DATA ONLY - SET DISK FILE INDICATORS
  170 LNDATA=8
      LNEXT=N*((MN+9)/10)+8
      PRINT 175
  175 FORMAT (31H READ AND FILE RAW NUMERIC DATA)
      GO TO 200
C  BINARY DATA ONLY - SET FILE INDICATORS
  180 LBDATA=8
      LNEXT=8
      PRINT 185
  185 FORMAT (26H READ AND FILE BINARY DATA)
      GO TO 200
C  MIXED-MODE DATA - SET FILE INDICATORS
  190 LNDATA=8
      LBDATA=N*((MN+9)/10)+8
      LNEXT=LBDATA
      PRINT 195
  195 FORMAT (30H READ AND FILE MIXED-MODE DATA)
C  BRANCH ON MODE INDICATOR
  200 GO TO (205,270,205),ID
C  NUMERIC DATA - BRANCH ON STANDARDISATION INDICATOR
  205 IF (ISTAND)220,220,210
C  STANDARDISATION SELECTED
  210 PRINT 215
  215 FORMAT (40H REPLACE NUMERIC DATA BY STANDARD SCORES)
C  SET INDICATOR ON
      ISTAND=2
      GO TO 230
  220 PRINT 225
  225 FORMAT (29H STANDARD SCORES NOT REQUIRED)
```

```
C  SET INDICATOR OFF
      ISTAND=1
C  BRANCH ON PRINCIPAL COMPONENTS INDICATOR
  230 GO TO (245,235),IPRC
C  PRINCIPAL COMPONENTS SELECTED
  235 PRINT 240
  240 FORMAT (56H PRINCIPAL COMPONENTS SOLUTION SELECTED FOR NUMERIC DAT
     1A)
C  ARE CORRELATIONS TO BE FILED
  245 IF (JCOR)290,290,260
C  FILE CORRELATIONS MESSAGE
  260 PRINT 265
  265 FORMAT (25H CORRELATIONS TO BE FILED)
C  NOW EXIT
      GO TO 290
C  BINARY DATA ONLY - IS STANDARDISATION SELECTED
  270 IF (ISTAND)285,285,275
C  STANDARDISATION SELECTED
  275 PRINT 280
  280 FORMAT (56H STANDARD SCORES ARE ONLY OBTAINED FOR NUMERIC VARIABLE
     1S)
C  SET INDICATOR OFF
  285 ISTAND=1
C  EXIT
  290 RETURN
      END


      SUBROUTINE LOAD
C  CALLED BY (FILE)
C  FUNCTION - TO LOAD NUMERIC, BINARY OR MIXED-MODE DATA TO DISK FILE
C  FOR MAIN PROGRAMME (FILE)
C  USING NUMERIC VARIABLES -
C       MEANS AND VARIANCES ARE COMPUTED AND FILED
C       PRINCIPAL COMPONENTS AND CORRELATIONS ARE COMPUTED AND FILED
C       WHEN SELECTED
C  RAW DATA ARE REPLACED BY STANDARD SCORES IF SELECTED
C  USING BINARY DAYA -
C       BINARY VARIABLE FREQUENCIES AND SAMPLE LIST LENGTHS ARE COMPUTED
C       AND FILED
C  DIMENSIONS AND COMMON AREAS FOR ALL FILE ROUTINES ******************
C  END OF COMMON RESERVED FOR DISK I/O SUBROUTINE (DISKIO) PARAMETERS
      DIMENSION TEXT(20)
      COMMON N,MB,MN,NPCF,NPC,ISTAND,IMASK,IDATA,ICOEF,ITYPE,KMAX
      COMMON LNDATA,LBDATA,LMEANS,LVARS,LCORS,LEIGS,LEIGVS,LSCORS,LENGS,
     1LFREQS,LNMASK,LBMASK,LMAT,LKLIST,LNEXT,TEXT
C  LIMIT ON MN = 200 ( NUMBER OF NUMERIC VARIABLES )
C  LIMIT ON MB = 401 (NUMBER OF BINARY VARIABLES)
C  LIMIT ON N = 250 ( NUMBER OF POINTS/CASES )
C  THE FOLLOWING DIMENSION, COMMON AND EQUIVALENCE STATEMENTS PERMIT
C  REAL AND INTEGER ARRAYS TO BE EQUIVALENCED (MOSTLY ILLEGAL IN 1620
C  FORTRAN II). THE REQUIRED LENGTHS OF THE INTEGER ARRAYS CONCERNED ARE
C  LIST(401), IFREQ(401), LENG(250). THE EQUIVALENCE WORKS ON THE
C  PRINCIPLE THAT 5 STORAGE LOCATIONS ARE REQUIRED FOR EACH INTEGER, AND
C  10 LOCATIONS FOR EACH REAL WORD - THUS A REAL WORD CAN SHARE SPACE
C  WITH TWO INTEGERS.
      DIMENSION COR(630),XM(200),VAR(200),X(200),Y(200),EIG(35)
     1,EIGV(1225),LIST(1),IFREQ(1),LENG(1)
      COMMON COR,XM,VAR,X,Y,EIG,EIGV,LIST
      EQUIVALENCE (LIST(402),IFREQ),(LIST(803),LENG)
```

```
C   HENCE LENG, LIST AND IFREQ ALL WRITE OVER THE ARRAY EIGV
C   SINGLE VARIABLE COMMON
      COMMON ID,IPRC,JEIGV,ICOR,JCOR
C   DIMENSIONS AND COMMON AREAS FOR ALL FILE ROUTINES ******************
C   BRANCH ON NUMERIC DATA-MODE INDICATOR
      GO TO (5,25,5),ID
C   NUMERIC DATA SUPPLIED - INITIALISE MEANS,VARIANCES AND CROSS-PRODUCTS
    5 DO 10 I=1,MN
      XM(I)=0.
   10 VAR(I)=0.
C   BRANCH ON CORRELATION INDICATOR
      GO TO (25,15),ICOR
C   INITIALISE CROSS-PRODUCTS - THE TRIANGULAR CORRELATION MATRIX IS
C   STORED IN THE LINEAR ARRAY COR OF LENGTH MCOR
   15 MCOR=(MN*MN+MN)/2
      DO 20 I=1,MCOR
   20 COR(I)=0.
C   BRANCH ON BINARY DATA-MODE INDICATOR
   25 GO TO (40,30,30),ID
C   INITIALISE FREQUENCY COUNTS
   30 DO 35 I=1,MB
   35 IFREQ(I)=0
C   ENTER DATA READING CYCLE
C   SET NUMERIC MODE DISK SECTOR IDISK
   40 IDISK=LNDATA
      DO 75 I=1,N
C   READ A CASE CARD
      IZ=I
      CALL READ (IZ,X,MN,LIST,LB)
C   BRANCH ON BINARY DATA-MODE INDICATOR
      GO TO (55,45,45),ID
C   BINARY DATA SUPPLIED - ADD TO FREQUENCY COUNTS
   45 DO 50 J=1,LB
      L=LIST(J)
   50 IFREQ(L)=IFREQ(L)+1
C   NOTE LENGTH OF ITH. BINARY SAMPLE LIST
      LENG(I)=LB
C   FILE BINARY SAMPLE LIST AT SECTOR LNEXT
      CALL DISKIO(3,LNEXT,LIST,LB,XDUMY,LNDUMY)
C   BRANCH ON NUMERIC DATA-MODE INDICATOR
      GO TO (75,75,55),ID
C   NUMERIC DATA SUPPLIED - CONSIDER MEANS, VARIANCES AND CORRELATIONS
C   SET LOCATION POINTER FOR LINEAR CORRELATION MATRIX
   55 IZ=0
      DO 70 J=1,MN
      P=X(J)
      XM(J)=XM(J)+P
      VAR(J)=VAR(J)+P*P
C   BRANCH ON CORRELATION INDICATOR
      GO TO (70,60),ICOR
C   CORRELATIONS SELECTED - ADD TO CROSS-PRODUCTS
   60 DO 65 L=1,J
      IZ=IZ+1
   65 COR(IZ)=COR(IZ)+P*X(L)
   70 CONTINUE
C   FILE NUMERIC LIST AT SECTOR IDISK
      CALL DISKIO(5,IDISK,IDUMY,IBDUMY,X,MN)
C   END OF I SAMPLE LOOP
   75 CONTINUE
      PRINT 76
```

```
   76 FORMAT (15H RAW DATA FILED)
C  SET FLOATING N
      FN=N
C  BRANCH ON BINARY DATA-MODE INDICATOR
      GO TO (95,80,80),ID
C  BINARY DATA - FILE LENGTHS AND FREQUENCIES
C  NOTE DISK SECTOR ADDRESS
   80 LENGS=LNEXT
C  FILE LENGTHS
      CALL DISKIO(3,LNEXT,LENG,N,XDUMY,LNDUMY)
      PRINT 85
   85 FORMAT (33H BINARY SAMPLE LIST LENGTHS FILED)
C  FILE FREQUENCIES
      LFREQS=LNEXT
      CALL DISKIO(3,LNEXT,IFREQ,MB,XDUMY,LNDUMY)
      PRINT 90
   90 FORMAT (34H BINARY VARIABLE FREQUENCIES FILED)
C  BRANCH ON NUMERIC DATA MODE INDICATOR
      GO TO (130,130,95),ID
C  NUMERIC DATA - COMPUTE AND FILE MEANS XM, AND VARIANCES VAR
   95 DO 100 J=1,MN
      P=XM(J)/FN
      VAR(J)=VAR(J)/FN-P*P
  100 XM(J)=P
C  FILE MEANS
      LMEANS=LNEXT
      CALL DISKIO(5,LNEXT,IDUMY,LBDUMY,XM,MN)
      PRINT 105
  105 FORMAT (20H NUMERIC MEANS FILED)
C  FILE VARIANCES
      LVARS=LNEXT
      CALL DISKIO(5,LNEXT,IDUMY,LBDUMY,VAR,MN)
      PRINT 110
  110 FORMAT (24H NUMERIC VARIANCES FILED)
C  BRANCH ON CORRELATION INDICATOR
      GO TO (130,115),ICOR
C  COMPUTE CORRELATIONS
  115 IZ=0
      DO 120 J=1,MN
      P=XM(J)
      Q=SQRT(VAR(J))
      DO 120 L=1,J
      IZ=IZ+1
  120 COR(IZ)=(COR(IZ)/FN-XM(L)*P)/(Q*SQRT(VAR(L)))
C  ARE CORRELATIONS TO BE FILED
      IF (JCOR)130,130,125
C  FILE CORRELATIONS
  125 LCORS=LNEXT
      CALL DISKIO(5,LNEXT,IDUMY,LBDUMY,COR,MCOR)
      PRINT 126
  126 FORMAT (27H NUMERIC CORRELATIONS FILED)
C  EXIT FROM LOAD
  130 RETURN
      END


      SUBROUTINE TRANUM
C  CALLED BY (FILE)
C  FUNTION - TO DERIVE STANDARD SCORES, PRINCIPAL COMPONENT EIGENVALUES
C  AND EIGENVECTORS FOR NUMERIC DATA FOR ROUTINE (FILE)
```

```
C   DIMENSIONS AND COMMON AREAS FOR ALL FILE ROUTINES ******************
C   END OF COMMON RESERVED FOR DISK I/O SUBROUTINE (DISKIO) PARAMETERS
      DIMENSION TEXT(20)
      COMMON N,MB,MN,NPCF,NPC,ISTAND,IMASK,IDATA,ICOEF,ITYPE,KMAX
      COMMON LNDATA,LBDATA,LMEANS,LVARS,LCORS,LEIGS,LEIGVS,LSCORS,LENGS,
     1LFREQS,LNMASK,LBMASK,LMAT,LKLIST,LNEXT,TEXT
C   LIMIT ON MN = 200 ( NUMBER OF NUMERIC VARIABLES )
C   LIMIT ON MB = 401 (NUMBER OF BINARY VARIABLES)
C   LIMIT ON N = 250 ( NUMBER OF POINTS/CASES )
C   THE FOLLOWING DIMENSION, COMMON AND EQUIVALENCE STATEMENTS PERMIT
C   REAL AND INTEGER ARRAYS TO BE EQUIVALENCED (MOSTLY ILLEGAL IN 1620
C   FORTRAN II). THE REQUIRED LENGTHS OF THE INTEGER ARRAYS CONCERNED ARE
C   LIST(401), IFREQ(401), LENG(250). THE EQUIVALENCE WORKS ON THE
C   PRINCIPLE THAT 5 STORAGE LOCATIONS ARE REQUIRED FOR EACH INTEGER, AND
C   10 LOCATIONS FOR EACH REAL WORD - THUS A REAL WORD CAN SHARE SPACE
C   WITH TWO INTEGERS.
      DIMENSION COR(630),XM(200),VAR(200),X(200),Y(200),EIG(35)
     1,EIGV(1225),LIST(1),IFREQ(1),LENG(1)
      COMMON COR,XM,VAR,X,Y,EIG,EIGV,LIST
      EQUIVALENCE (LIST(402),IFREQ),(LIST(803),LENG)
C   HENCE LENG, LIST AND IFREQ ALL WRITE OVER THE ARRAY EIGV
C   SINGLE VARIABLE COMMON
      COMMON ID,IPRC,JEIGV,ICOR,JCOR
C   DIMENSIONS AND COMMON AREAS FOR ALL FILE ROUTINES ******************
C   BRANCH ON PRINCIPAL COMPONENTS INDICATOR
      GO TO (5,10),IPRC
C   PRINCIPAL COMPONENTS SOLUTION NOT REQUIRED - ARE STANDARD SCORES
C   REQUIRED
C   BRANCH ON STANDARDISATION INDICATOR (EXIT AT 125 IF NOT SELECTED)
    5 GO TO (125,40),ISTAND
C   PRINCIPAL COMPONENTS REQUIRED - COMPUTE EIGENVALUES AND EIGENVECTORS
   10 CALL EIGEN(COR,EIGV,MN,0)
C   PLACE EIGENVALUES IN EIG
      DO 15 I=1,MN
      J=I+(I*I-I)/2
   15 EIG(I)=COR(J)
C   PRINT EIGENVALUES MESSAGE
      PRINT 20
   20 FORMAT (17H FILE EIGENVALUES)
C   FILE EIGENVALUES
      LEIGS=LNEXT
      CALL DISKIO(5,LNEXT,IDUMY,LBDUMY,EIG,MN)
C   ARE EIGENVECTORS TO BE FILED
      IF (JEIGV)40,40,30
C   FILE EIGENVECTORS
   30 LEIGVS=LNEXT
      IZ=MN*MN
      CALL DISKIO(5,LNEXT,IDUMY,LBDUMY,EIGV,IZ)
      PRINT 35
   35 FORMAT (23H ALL EIGENVECTORS FILED)
C   COMPUTE RECIPROCAL STANDARD DEVIATIONS
   40 DO 45 J=1,MN
   45 VAR(J)=1./SQRT(VAR(J))
C   SCAN FOR NUMERIC DATA - SET DISK INDICATORS
      IDISK=LNDATA
      JDISK=LNDATA
C   ARE COMPONENT SCORES TO BE FILED - BRANCH ON INDICATOR
      GO TO (55,50),IPRC
C   COMPONENT SCORES TO BE FILED - SET DISK INDICATOR
   50 LSCORS=LNEXT
```

```
   55 DO 95 I=1,N
C  GET THE ITH. SAMPLE
      CALL DISKIO(6,IDISK,IDUMY,LBDUMY,X,MN)
C  STANDARDISE VALUES
      DO 60 J=1,MN
   60 X(J)=(X(J)-XM(J))*VAR(J)
C  ARE STANDARD SCORES TO BE FILED
      GO TO (70,65),ISTAND
C  FILE STANDARD SCORES
   65 CALL DISKIO(5,JDISK,IDUMY,LBDUMY,X,MN)
C  BRANCH ON PRINCIPAL COMPONENTS INDICATOR
   70 GO TO (95,75),IPRC
C  COMPUTE PRINCIPAL COMPONENT SCORES
   75 IZ=0
      DO 90 J=1,NPCF
      P=0.
      DO 80 K=1,MN
      IZ=IZ+1
   80 P=P+EIGV(IZ)*X(K)
   90 Y(J)=P
C  FILE COMPONENT SCORES
      CALL DISKIO(5,LNEXT,IDUMY,JDUMY,Y,NPCF)
C  END OF SAMPLE I LOOP
   95 CONTINUE
C  BRANCH ON STANDARDISATION INDICATOR
      GO TO (110,100),ISTAND
C  STANDARD SCORES FILED MESSAGE
  100 PRINT 105
  105 FORMAT (22H STANDARD SCORES FILED)
C  BRANCH ON PRINCIPAL COMPONENTS INDICATOR
  110 GO TO (125,115),IPRC
C  COMPONENT SCORES FILED MESSAGE
  115 PRINT 120,NPCF
  120 FORMAT (I4,23H COMPONENT SCORES FILED)
C  EXIT
  125 RETURN
      END


C
C  THIS IS A MODIFIED VERSION OF THE EIGENVECTOR/VALUE SUBROUTINE
C  (EIGEN) PUBLISHED IN SYSTEM/360 SCIENTIFIC SUBROUTINE PACKAGE
C  (360A-CM-03X) - REF H20-0205-0
C  **********************************************************************
C  SUBROUTINE EIGEN
C  PURPOSE
C     COMPUTE EIGENVALUES AND EIGENVECTORS OF A REAL SYMMETRIC MATRIX
C  USAGE
C     CALL EIGEN(A,R,N,MV)
C  DESCRIPTION OF PARAMETERS
C     A - ORIGINAL MATRIX (SYMMETRIC), DESTROYED IN COMPUTATION.
C         RESULTANT EIGENVALUES ARE DEVELOPED IN DIAGONAL OF MATRIX A
C         IN DESCENDING ORDER.
C     R - RESULTANT MATRIX OF EIGENVECTORS (STORED COLUMNWISE,IN SAME
C         SEQUENCE AS EIGENVALUES)
C     N - ORDER OF MATRICES A AND R
C     MV- INPUT CODE
C             0  COMPUTE EIGENVALUES AND EIGENVECTORS
C             1  COMPUTE EIGENVALUES ONLY (R NEED NOT BE DIMENSIONED
C                BUT MUST STILL APPEAR IN CALLING SEQUENCE)
```

```
C   REMARKS
C      ORIGINAL MATRIX A MUST BE REAL SYMMETRIC (STORAGE MODE=1)
C      MATRIX A CANNOT BE IN THE SAME LOCATION AS MATRIX R
C   SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C      NONE
C   METHOD
C      DIAGONALIZATION METHOD ORIGINATED BY JACOBI AND ADAPTED BY VON
C      NEUMANN FOR LARGE COMPUTERS AS FOUND IN  - MATHEMATICAL METHODS FOR
C      DIGITAL COMPUTERS - EDITED BY A. RALSTON AND H. S. WILF, JOHN WILEY
C      AND SONS, NEW YORK, 1962, CHAPTER 7
C   *******************************************************************
      SUBROUTINE EIGEN(A,R,N,MV)
      DIMENSION A(1),R(1)
C   *******************************************************************
C   GENERATE IDENTITY MATRIX
      FN=N
      IF (MV-1)10,25,10
   10 IQ=-N
      DO 20 J=1,N
      IQ=IQ+N
      DO 20 I=1,N
      IJ=IQ+I
      R(IJ)=0.0
      IF(I-J)20,15,20
 15   R(IJ)=1.0
 20   CONTINUE
C   COMPUTE INITIAL AND FINAL NORMS (ANORM AND ANORMX)
 25   ANORM=0.0
      DO 35 I=1,N
      DO 35 J=I,N
      IF(I-J)30,35,30
 30   IA=I+(J*J-J)/2
      ANORM=ANORM+A(IA)*A(IA)
 35   CONTINUE
      IF (ANORM)165,165,40
 40   ANORM=1.414*SQRT(ANORM)
      ANRMX=ANORM*1.0E-6/FN
C   INITIALIZE INDICATORS AND COMPUTE THRESHOLD, THR
      IND=0
      THR=ANORM
   45 THR=THR/FN
 50   L=1
 55   M=L+1
C   COMPUTE SIN AND COS
 60   MQ=(M*M-M)/2
      LQ=(L*L-L)/2
      LM=L+MQ
   62 IF (ABSF(A(LM))-THR)130,65,65
 65   IND=1
      LL=L+LQ
      MM=M+MQ
      X=0.5*(A(LL)-A(MM))
   68 Y=-A(LM)/SQRT(A(LM)*A(LM)+X*X)
      IF(X)70,75,75
 70   Y=-Y
 75   SINX=Y/SQRT(2.0*(1.0+(SQRT(1.0-Y*Y))))
      SINX2=SINX*SINX
 78   COSX=SQRT(1.0-SINX2)
      COSX2=COSX*COSX
      SINCS=SINX*COSX
```

```
C   ROTATE L AND M COLUMNS
        ILQ=N*(L-1)
        IMQ=N*(M-1)
        DO 125 I=1,N
        IQ=(I*I-I)/2
        IF (I-L)80,115,80
 80     IF(I-M)85,115,90
 85     IM=I+MQ
        GO TO 95
 90     IM=M+IQ
 95     IF(I-L)100,105,105
 100    IL=I+LQ
        GO TO 110
 105    IL=L+IQ
 110    X=A(IL)*COSX-A(IM)*SINX
        A(IM)=A(IL)*SINX+A(IM)*COSX
        A(IL)=X
 115    IF(MV-1)120,125,120
 120    ILR=ILQ+I
        IMR=IMQ+I
        X=R(ILR)*COSX-R(IMR)*SINX
        R(IMR)=R(ILR)*SINX+R(IMR)*COSX
        R(ILR)=X
 125    CONTINUE
        X=2.0*A(LM)*SINCS
        Y=A(LL)*COSX2+A(MM)*SINX2-X
        X=A(LL)*SINX2+A(MM)*COSX2+X
        A(LM)=(A(LL)-A(MM))*SINCS+A(LM)*(COSX2-SINX2)
        A(LL)=Y
        A(MM)=X
C   TESTS FOR COMPLETION
C   TEST FOR M = LAST COLUMN
 130    IF(M-N)135,140,135
 135    M=M+1
        GO TO 60
C   TEST FOR L = SECOND FROM LAST COLUMN
 140    IF(L-(N-1))145,150,145
 145    L=L+1
        GO TO 55
 150    IF(IND-1)160,155,160
 155    IND=0
        GO TO 50
C   COMPARE THRESHOLD WITH FINAL NORM
  160   IF (THR-ANRMX)165,165,45
C   SORT EIGENVALUES AND EIGENVECTORS
 165    IQ=-N
        DO 185 I=1,N
        IQ=IQ+N
        LL=I+(I*I-I)/2
        JQ=N*(I-2)
        DO 185 J=I,N
        JQ=JQ+N
        MM=J+(J*J-J)/2
        IF(A(LL)-A(MM))170,185,185
 170    X=A(LL)
        A(LL)=A(MM)
        A(MM)=X
        IF (MV-1)175,185,175
 175    DO 180 K=1,N
        ILR=IQ+K
        IMR=JQ+K
```

```
      X=R(ILR)
      R(ILR)=R(IMR)
  180 R(IMR)=X
  185 CONTINUE
      RETURN
      END

C  ROUTINE CORREL
C  AUTHOR - DAVID WISHART, MATHEMATICAL INSTITUTE, UNIVERSITY OF ST.
C  ANDREWS, FIFE, GREAT BRITAIN.  (PROGRAMME WRITTEN - JUNE 1968)
C  CALLS SUBROUTINES INSERT, DISKIO, COREAD, ANALYS AND COEF
      DEFINE DISK (10,4300)
C  DIMENSIONS AND COMMON AREA FOR ALL CORREL ROUTINES *****************
C  END OF COMMON RESERVED FOR DISK I/O SUBROUTINE (DISKIO) PARAMETERS
      DIMENSION TEXT(20)
      COMMON N,MB,MN,NPCF,NPC,ISTAND,IMASK,IDATA,ICOEF,ITYPE,KMAX
      COMMON LNDATA,LBDATA,LMEANS,LVARS,LCORS,LEIGS,LEIGVS,LSCORS,LENGS,
     1LFREQS,LNMASK,LBMASK,LMAT,LKLIST,LNEXT,TEXT
C  LIMIT ON MN = 200 ( NUMBER OF NUMERIC VARIABLES )
C  LIMIT ON MB = 401 (NUMBER OF BINARY VARIABLES)
C  LIMIT ON N = 250 ( NUMBER OF POINTS/CASES )
C  LIMIT ON KMAX = 1400/N
C  THIS DIMENSION STATEMENT COUPLED WITH THE USE OF COMMON ENABLES AN
C  EFFECTIVE EQUIVALENCE (XMASK,MASKB),(X,IX),(Y,LIST) TO BE USED
C  WITHOUT EQUAL FANDK.  THUS THE EFFECTIVE (WRITE OVER) DIMENSION OF
C  THE ARRAYS MASKB,IX, AND LIST FOR AN FANDK OF 0805 IS 401
      DIMENSION DAR(1400),KLINK(1400),XMASK(200),MASKB(1),X(200),IX(1),
     1Y(200),LIST(1),C(250),LENG(250)
C  ARRAY COMMON
      COMMON DAR,KLINK,XMASK,MASKB,X,IX,Y,LIST,C,LENG
C  SINGLE VARIABLE COMMON
      COMMON M,LX,PX,PY,LY,IDISK,PM
C  DIMENSIONS AND COMMON AREA FOR ALL CORREL ROUTINES *****************
C  FUNCTION - TO COMPUT THE COEFFICIENT MATRIX AND K-LINKAGE LISTS DAR
C  AND KLINK FOR INPUT TO MODE-ANALYSIS CLASSIFICATION PROGRAMME (MODE)
C  READ DISK FILE PARAMETERS
      CALL DISKIO(1,LDUMY,IDUMY,LBDUMY,XDUMY,LNDUMY)
C  READ CARD INPUT, ANALYSE FOR ERRORS AND ASSEMBLE MASK
      CALL COREAD
      CALL ANALYS
C  INITIALISE THE K-LINKAGE LISTS TO EXTREME VALUES
C  CONSIDER THE TWO-DIMENSIONAL ARRAYS DAR AND KLINK AS LINEAR ARRAYS
C  OF LENGTH N*KMAX
      IZ=N*KMAX
      MNF=MN
      IF (NPC)3,3,2
    2 MN=NPCF
    3 DO 15 I=1,IZ
C  BRANCH ON COEFFICIENT TYPE INDICATOR
      GO TO (5,10),ITYPE
C  SIMILARITY COEFFICIENT
    5 DAR(I)=-10.E+90
      GO TO 15
C  DISSIMILARITY COEFFICIENT
   10 DAR(I)=10.E+90
   15 KLINK(I)=0
C  SET FLOATING VERSION OF M, NUMBER OF UNMASKED VARIABLES
      PM=M
C  SET STARTING SECTOR ADDRESS OF SAMPLE 2, ID
C  BRANCH ON DATA TYPE INDICATOR
      GO TO (16,17),IDATA
```

```
C   IDATA=1, BINARY DATA
    16 ID=IDISK+(LENG(1)+9)/10
       GO TO 18
C   IDATA =2, NUMERIC DATA
    17 ID=IDISK+(MN+9)/10
C   SCAN DATA FILE FOR (1ST. SAMPLE - I)
    18 DO 65 I=2,N
C   BRANCH ON DATA TYPE INDICATOR
       GO TO (20,35),IDATA
C   IDATA=1, BINARY DATA
    20 LX=LENG(I)
C   GET THE ITH. SAMPLE LIST (LIST(L),L=1,LX)
       CALL DISKIO(4,ID,LIST,LX,XDUMY,LNDUMY)
C   ASSEMBLE BINARY IX(L) CONTAINING 1 IF ATTRIBUTE L UNMASKED AND
C   POSSESSED BY SAMPLE I, 0 OTHERWISE. NX = NUMBER POSSESSED
       DO 25 L=1,MB
    25 IX(L)=0
       NX=0
       DO 30 L=1,LX
       J=LIST(L)
       IX(J)=MASKB(J)
    30 NX=NX+MASKB(J)
C   COMPUTE PX, THE NUMBER OF UNMASKED ATTRIBUTES POSSESSED BY SAMPLE I
       PX=NX
C   GO TO SCAN DATA FILE FOR (2ND. SAMPLE - J)
       GO TO 40
C   IDATA =2, NUMERIC DATA
C   GET THE ITH. SAMPLE NUMERIC VARIABLE VALUES (X(L),L=1,MN)
    35 CALL DISKIO(6,ID,IDUMY,LBDUMY,X,MN)
C   NOW SCAN DATA FILE FOR (2ND. SAMPLE - J)
    40 JX=I-1
C   SET STARTING SECTOR ADDRESS OF SAMPLE 1
       JD=IDISK
       DO 60 J=1,JX
C   BRANCH ON DATA TYPE INDICATOR
       GO TO (45,50),IDATA
C   IDATA =1, BINARY DATA
    45 LY=LENG(J)
C   GET THE JTH. SAMPLE LIST (LIST(L),L=1,LY)
       CALL DISKIO(4,JD,LIST,LY,XDUMY,LNDUMY)
C   NOW GO TO COEFFICIENT CALCULATION
       GO TO 55
C   IDATA =2, NUMERIC DATA
C   GET THE JTH. SAMPLE NUMERIC VARIABLE VALUES (Y(L),L=1,MN)
    50 CALL DISKIO(6,JD,IDUMY,LBDUMY,Y,MN)
C   ENTER COEFFICIENT CALCULATOR
    55 CALL COEF(P)
C   ASSEMBLE COEFFICIENT MATRIX ROW
       C(J)=P
C   INSERT C(J) INTO THE APPROPRIATE K-LISTS FOR SAMPLES (I,J)
       CALL INSERT (I,J,P)
       CALL INSERT (J,I,P)
C   END OF (2ND. SAMPLE - J) SCANNING LOOP
    60 CONTINUE
C   NOW FILE THE (1ST. SAMPLE) COEFFICIENT MATRIX ROW
       CALL DISKIO(5,LNEXT,IDUMY,LBDUMY,C,JX)
C   TRACE FEATURE - REMOVE THIS SECTION IF NOT REQUIRED ****************
       IF (SENSE SWITCH 4)600,610
   600 JX=N-I
       TYPE 605,JX
```

```
  605 FORMAT (I3,13H CYCLES TO GO)
  610 CONTINUE
C  END OF TRACE  - REMOVE THIS SECTION IF NOT REQUIRED ****************
C  END OF (1ST. SAMPLE - I) SCANNING LOOP
   65 CONTINUE
C  FILE THE K-LINKAGE LISTS DAR AND KLINK
C  THIS IS ACCOMPLISHED BY FILING EACH DAR AND KLINK LIST FOR EACH VALUE
C  OF K SEPARATELY
C  NOTE STARTING POSITION OF LISTS
      LKLIST=LNEXT
      DO 70 K=1,KMAX
C  COMPUTE THE STARTING POSITION OF THE KTH. LIST IN THE LINEAR ARRAYS
C  DAR AND KLINK
      IZ=(K-1)*N+1
C  NOW FILE DAR
      CALL DISKIO(5,LNEXT,IDUMY,LBDUMY,DAR(IZ),N)
C  NOW FILE KLINK
      CALL DISKIO(3,LNEXT,KLINK(IZ),N,XDUMY,LNDUMY)
C  END OF LOOP
   70 CONTINUE
C  NOW WRITE DISK FILE PARAMETERS
      MN=MNF
      CALL DISKIO(2,LDUMY,IDUMY,LBDUMY,XDUMY,LNDUMY)
C  PRINT ENDING MESSAGE
      PRINT 75
   75 FORMAT (54H COEFFICIENTS AND K-LINKAGE LISTS CALCULATED AND FILED/
     119H JOB ENDS/1H1)
      CALL EXIT
C  DUMMY CALL STATEMENT TO ENABLE LOCALISATION OF COREAD, ANALYS AND
C  COEF (IBM 1620 MONITOR II SYSTEM)
   80 P=SQRT(P)
      END


      SUBROUTINE COREAD
C  CALLED BY (CORREL)
C  FUNCTION - TO CHECK FILE PARAMETERS AND CARD INPUT FOR
C  COEFFICIENT MATRIX AND K-LIST CALCULATION (CORREL)
C  DIMENSIONS AND COMMON AREA FOR ALL CORREL ROUTINES *****************
C  END OF COMMON RESERVED FOR DISK I/O SUBROUTINE (DISKIO) PARAMETERS
      DIMENSION TEXT(20)
      COMMON N,MB,MN,NPCF,NPC,ISTAND,IMASK,IDATA,ICOEF,ITYPE,KMAX
      COMMON LNDATA,LBDATA,LMEANS,LVARS,LCORS,LEIGS,LEIGVS,LSCORS,LENGS,
     1LFREQS,LNMASK,LBMASK,LMAT,LKLIST,LNEXT,TEXT
C  LIMIT ON MN = 200 ( NUMBER OF NUMERIC VARIABLES )
C  LIMIT ON MB = 401 (NUMBER OF BINARY VARIABLES)
C  LIMIT ON N = 250 ( NUMBER OF POINTS/CASES )
C  LIMIT ON KMAX = 1400/N
C  THIS DIMENSION STATEMENT COUPLED WITH THE USE OF COMMON ENABLES AN
C  EFFECTIVE EQUIVALENCE (XMASK,MASKB),(X,IX),(Y,LIST) TO BE USED
C  WITHOUT EQUAL FANDK.  THUS THE EFFECTIVE (WRITE OVER) DIMENSION OF
C  THE ARRAYS MASKB,IX, AND LIST FOR AN FANDK OF 0805 IS 401
      DIMENSION DAR(1400),KLINK(1400),XMASK(200),MASKB(1),X(200),IX(1),
     1Y(200),LIST(1),C(250),LENG(250)
C  ARRAY COMMON
      COMMON DAR,KLINK,XMASK,MASKB,X,IX,Y,LIST,C,LENG
C  SINGLE VARIABLE COMMON
      COMMON M,LX,PX,PY,LY,IDISK,PM
C  DIMENSIONS AND COMMON AREA FOR ALL CORREL ROUTINES *****************
C  SET COEFFICIENT PARAMETERS TO NULL
      NPC=0
```

```
      IMASK=0
      IDATA=0
      ITYPE=0
C     PRINT DATA FILE IDENTIFICATION
      PRINT 5,(TEXT(I),I=1,20)
    5 FORMAT (1X,20A4//)
      PRINT 10,N,MB,MN
   10 FORMAT (19H NUMBER OF CASES = ,I4/30H NUMBER OF BINARY VARIABLES =
     1 ,I4/31H NUMBER OF NUMERIC VARIABLES = ,I4//54H CALCULATION OF COE
     1FFICIENT MATRIX AND K-LINKAGE LISTS//)
C     READ INPUT PARAMETERS
   12 FORMAT (3I2,2I3)
      READ 12,ICOEF,KMAX,NPC,MSKB,MSKN
C     TEST IF FILE DATA IS OK
      IF (MN*LNDATA+MB*LBDATA)11,11,14
   11 PRINT 13
   13 FORMAT (23H DATA FILE NOT COMPLETE)
C     ABORT
      GO TO 65
   15 FORMAT (20I4)
C     REDUCE LNEXT TO OBLITERATE EXISTING COEFFICIENT MATRIX FILE
   14 IF (LNMASK)17,17,16
   16 LNEXT=LNMASK
      GO TO 21
   17 IF (LBMASK)19,19,18
   18 LNEXT=LBMASK
      GO TO 21
   19 IF (LMAT)21,21,20
   20 LNEXT=LMAT
C     SET COEFFICIENT PARAMETERS TO NULL
   21 LMAT=0
      LNMASK=0
      LBMASK=0
      LKLIST=0
C     ARE PRINCIPAL COMPONENT SCORES SELECTED
      IF (NPC)25,25,240
C     PRINCIPAL COMPONENT SCORES NOT SELECTED
C     ENTER BINARY/NUMERIC MASK ANALYSER
   25 IMASK=1
      IF (MN)90,90,30
   30 IF (MB)165,165,35
C     THEREFORE MIXED MODE DATA
   35 IMASK=2
      IF (MSKB-MB)50,40,40
C     THEREFORE MASK ALL BINARY DATA
   40 LBMASK=-1
C     PRINT MASK
      PRINT 45
   45 FORMAT (21H MASK ALL BINARY DATA//)
      GO TO 165
C     ALL BINARY DATA NOT MASKED
   50 IF (MSKN-MN)55,75,75
C     MIXED-MODE DATA WITHOUT TOTAL MASK OF ONE MODE
   55 PRINT 60
   60 FORMAT (38H ERROR - MIXED-MODE DATA NOT PERMITTED)
C     GENERAL ABORT ROUTINE
   65 PRINT 70
   70 FORMAT (17H PHASE TERMINATED/9H JOB ENDS/1H1)
C     WRITE DISK FILE PARAMETERS
      CALL DISKIO(2,LSDUMY,IBDUMY,LBDUMY,XDUMY,LNDUMY)
```

```
      CALL EXIT
C  ALL NUMERIC DATA MASKED
   75 LNMASK=-1
C  PRINT MASK
      PRINT 85
   85 FORMAT (22H MASK ALL NUMERIC DATA//)
C  BINARY DATA TO BE USED
   90 PRINT 95
   95 FORMAT (16H USE BINARY DATA)
C  IS BINARY MASK APPLIED
      IF (MSKB)135,135,100
C  BINARY MASK APPLIED
  100 IF (MSKB-MB)115,105,105
C  ALL DATA MASKED
  105 PRINT 110
  110 FORMAT (29H ERROR - ALL VARIABLES MASKED)
      GO TO 65
C  READ BINARY MASK
  115 READ 15,(LIST(I),I=1,MSKB)
C  ASSEMBLE BINARY MASK ARRAY
      DO 120 I=1,MB
  120 MASKB(I)=1
      DO 125 I=1,MSKB
      J=LIST(I)
  125 MASKB(J)=0
C  FILE BINARY MASK
      LBMASK=LNEXT
      CALL DISKIO(3,LNEXT,MASKB,MB,XDUMY,LNDUMY)
C  PRINT MASK MESSAGE
      PRINT 130
  130 FORMAT (75H THE FOLLOWING BINARY VARIABLES ARE MASKED FROM THE COE
     1FFICIENT CALCULATION)
      PRINT 15,(LIST(I),I=1,MSKB)
C  SET MASKING INDICATOR
      IMASK=2
      GO TO 145
C  NO BINARY MASK APPLIED
  135 PRINT 140
  140 FORMAT (23H NO BINARY MASK APPLIED)
C  SET UP NULL BINARY MASK
      DO 142 I=1,MB
  142 MASKB(I)=1
C  SET DATA TYPE INDICATOR
  145 IDATA=1
C  SET DISK MARKER TO BINARY DATA
      IDISK=LBDATA
C  SET NUMBER OF VARIABLES M
      M=MB-MSKB
C  CHECK THAT BINARY SAMPLE LIST LENGTHS ARE FILED
      IF (LENGS)155,155,150
C  NOW READ BINARY SAMPLE LIST LENGTHS (LENG(I),I=1,N)
  150 IZ=LENGS
      CALL DISKIO(4,IZ,LENG,N,XDUMY,LNDUMY)
C  EXIT FROM MASK ANALYSER
      GO TO 305
C  LENG IS ABSENT FROM FILE
  155 PRINT 160
  160 FORMAT (45H ERROR - BINARY SAMPLE LIST LENGTHS NOT FILED)
      GO TO 65
C  NUMERIC DATA USED
```

```
  165 PRINT 170
  170 FORMAT (17H USE NUMERIC DATA)
C  TEST WHETHER NUMERIC MASK APPLIED
      IF (MSKN)230,230,175
C  TEST FOR TOTAL MASK
  175  IF (MSKN-MN)180,105,105
C  NUMERIC MASK APPLIED - READ MASK
  180 READ 15,(LIST(I),I=1,MSKN)
C  ASSEMBLE NUMERIC MASK
      DO 185 I=1,MN
  185 XMASK(I)=1.
      DO 190 I=1,MSKN
      J=LIST(I)
  190 XMASK(J)=0.
C  FILE NUMERIC MASK
      LNMASK=LNEXT
      CALL DISKIO(5,LNEXT,IDUMY,LBDUMY,XMASK,MN)
C  PRINT MASK MESSAGE
      PRINT 195
  195 FORMAT (76H THE FOLLOWING NUMERIC VARIABLES ARE MASKED FROM THE CO
     1EFFICIENT CALCULATION)
      PRINT 15,(LIST(I),I=1,MSKN)
C  SET MASKING INDICATOR
      IMASK=2
C  BRANCH ON STANDARDISATION INDICATOR
  200 GO TO (205,215),ISTAND
C  RAW DATA SELECTED
  205 PRINT 210
  210 FORMAT (18H RAW DATA SELECTED)
      GO TO 225
C  STANDARD SCORES SELECTED
  215 PRINT 220
  220 FORMAT (25H STANDARD SCORES SELECTED)
C  SET DISK MARKER TO NUMERIC DATA
  225 IDISK=LNDATA
C  SET NUMBER OF VARIABLES M
      M=MN-MSKN
C  SET DATA TYPE INDICATOR
      IDATA=2
C  EXIT FROM MASK ANALYSER
      GOTO 305
C  NO NUMERIC MASK APPLIED
  230 PRINT 235
  235 FORMAT (24H NO NUMERIC MASK APPLIED)
C  SET UP NULL NUMERIC MASK
      DO 237 I=1,MN
  237 XMASK(I)=1.
      GO TO 200
C  ENTER PRINCIPAL COMPONENTS ANALYSER
  240 IF (NPCF)245,245,255
C  NO COMPONENT SCORES FILED
  245 PRINT 250
  250 FORMAT (35H ERROR - COMPONENT SCORES NOT FILED)
      GO TO 65
C  ARE ENOUGH COMPONENT SCORES STORED
  255 IF (NPC-NPCF)275,270,260
C  NOT ENOUGH
  260 PRINT 265,NPCF
  265 FORMAT (13H ERROR - ONLY,I4,23H COMPONENT SCORES FILED)
      NPC=NPCF
```

```
C  NPC=NPCF, SET MASK INDICATOR OFF
  270 IMASK=1
      GO TO 285
C  MORE THAN NPC SCORES STORED - SET MASK FOR COMPONENT SCORES
  275 IZ=NPC+1
      DO 280 I=IZ,NPCF
  280 XMASK(I)=0.
C  SET MASK INDICATOR
      IMASK=2
  285 DO 290 I=1,NPC
  290 XMASK(I)=1.
C  SET NUMBER OF VARIABLES M, DATA TYPE INDICATOR = NUMERIC
      M=NPC
      IDATA=2
C  SET DISK MARKER TO COMPONENT SCORES
      IDISK=LSCORS
      PRINT 300,NPC
  300 FORMAT (13H SELECT FIRST,I4,17H COMPONENT SCORES)
C  EXIT FROM MASKER AND PRINCIPAL COMPONENTS ANALYSER
  305 RETURN
      END


      SUBROUTINE ANALYS
C  CALLED BY (CORREL)
C  FUNCTION - TO CHECK COEFFICIENT CODE, VALUE OF KMAX AND FILE
C  PARAMETERS N,MN,MB (FOLLOWS SUBROUTINE COREAD)
C  DIMENSIONS AND COMMON AREA FOR ALL CORREL ROUTINES *****************
C  END OF COMMON RESERVED FOR DISK I/O SUBROUTINE (DISKIO) PARAMETERS
      DIMENSION TEXT(20)
      COMMON N,MB,MN,NPCF,NPC,ISTAND,IMASK,IDATA,ICOEF,ITYPE,KMAX
      COMMON LNDATA,LBDATA,LMEANS,LVARS,LCORS,LEIGS,LEIGVS,LSCORS,LENGS,
     1LFREQS,LNMASK,LBMASK,LMAT,LKLIST,LNEXT,TEXT
C  LIMIT ON MN = 200 ( NUMBER OF NUMERIC VARIABLES )
C  LIMIT ON MB = 401 (NUMBER OF BINARY VARIABLES)
C  LIMIT ON N = 250 ( NUMBER OF POINTS/CASES )
C  LIMIT ON KMAX = 1400/N
C  THIS DIMENSION STATEMENT COUPLED WITH THE USE OF COMMON ENABLES AN
C  EFFECTIVE EQUIVALENCE (XMASK,MASKB),(X,IX),(Y,LIST) TO BE USED
C  WITHOUT EQUAL FANDK.  THUS THE EFFECTIVE (WRITE OVER) DIMENSION OF
C  THE ARRAYS MASKB,IX, AND LIST FOR AN FANDK OF 0805 IS 401
      DIMENSION DAR(1400),KLINK(1400),XMASK(200),MASKB(1),X(200),IX(1),
     1Y(200),LIST(1),C(250),LENG(250)
C  ARRAY COMMON
      COMMON DAR,KLINK,XMASK,MASKB,X,IX,Y,LIST,C,LENG
C  SINGLE VARIABLE COMMON
      COMMON M,LX,PX,PY,LY,IDISK,PM
C  DIMENSIONS AND COMMON AREA FOR ALL CORREL ROUTINES *****************
C  SET MAXIMUM COEFFICIENT CODE INDICATOR (REVISE WHEN ADDING TO (COEF)
C  OPTIONS)
      MAXCOF=19
C  TEST WHETHER DATA FILE IS STORED
      IF (IDISK)310,310,320
C  DATA FILE ABSENT
  310 PRINT 315
  315 FORMAT (31H ERROR - DATA FILE NOT COMPLETE)
C  GENERAL ABORT ROUTINE
   65 PRINT 70
   70 FORMAT (17H PHASE TERMINATED/9H JOB ENDS/1H1)
C  WRITE DISK FILE PARAMETERS
```

```
      CALL DISKIO(2,LSDUMY,IBDUMY,LBDUMY,XDUMY,LNDUMY)
      CALL EXIT
C  DATA FILE PRESENT
C  TEST IF COEFFICIENT SELECTED
  320 IF (ICOEF)325,325,340
C  COEFFICIENT NOT SELECTED - SUBSTITUTE EUCLIDEAN DISTANCE
C  BRANCH ON DATA TYPE INDICATOR
  325 GO TO (330,335),IDATA
C  BINARY DATA - SELECT (B+C)/M
  330 ICOEF=2
      GO TO 390
C  NUMERIC DATA - SELECT DSQ
  335 ICOEF=1
      GO TO 390
C  TEST IF VALID COEFFICIET CODE CHOSEN
  340 IF (ICOEF-MAXCOF)355,355,345
  345 PRINT 350,ICOEF
  350 FORMAT (27H ERROR - COEFFICIENT NUMBER,I4)
      GO TO 65
C  VALID COEFFICIENT CODE CHOSEN
C  BRANCH TO CHECK IF COEFFICIENT MATCHES DATA TYPE (REVISE WHEN ADDING
C  TO (COEF) OPTIONS)
  355 GO TO (360,375,360,375,375,375,375,375,375,375,375,375,375,375,375
     1,375,375,375,375),ICOEF
C  NUMERIC COEFFICIENT
  360 GO TO (365,390),IDATA
C  BINARY DATA - PRINT ERROR
  365 PRINT 370
  370 FORMAT (53H ERROR - NUMERIC COEFFICIENT SELECTED FOR BINARY DATA)
      GO TO 65
C  BINARY COEFFICIENT
  375 GO TO (390,380),IDATA
C  NUMERIC DATA - PRINT ERROR
  380 PRINT 385
  385 FORMAT (53H ERROR - BINARY COEFFICIENT SELECTED FOR NUMERIC DATA)
      GO TO 65
C  COEFFICIENT MATCHES DATA
  390 PRINT 395,ICOEF
  395 FORMAT (27H SELECT COEFFICIENT NUMBER ,I2)
C  SET COEFFICIENT TYPE INDICATOR
C  BRANCH ON COEFFICIENT CODE (REVISE WHEN ADDING TO (COEF) OPTIONS)
      GO TO (400,400,410,410,410,410,410,410,410,410,410,410,410,410,410
     1,410,410,410,410),ICOEF
C  DISSIMILARITY COEFFICIENT
  400 ITYPE=2
      PRINT 405
  405 FORMAT (34H COEFFICIENT IS DISSIMILARITY TYPE)
      GO TO 420
C  SIMILARITY COEFFICIENT
  410 ITYPE=1
      PRINT 415
  415 FORMAT (31H COEFFICIENT IS SIMILARITY TYPE)
C  IS KMAX NONZERO
  420 IF (KMAX)425,425,430
C  KMAX ZERO - SET TO 5
  425 KMAX=5
      GO TO 445
C  NONZERO KMAX - TEST FOR EXCESSIVE VALUE
  430 IF (KMAX*N-1400)445,445,435
C  EXCESSIVE VALUE SO MODIFY KMAX
```

```
  435 IZ=1400/N
      PRINT 440,N,IZ
  440 FORMAT (40H ERROR - MAXIMUM KMAX FOR DATA SET SIZE ,I4,4H IS ,I3)
      KMAX=IZ
C  IS KMAX LESS THAN N
  445 IF (KMAX-N)447,446,446
C  KMAX NOT LESS THAN N SO MODIFY
  446 KMAX=N-1
  447 PRINT 450,KMAX
  450 FORMAT  (6H FILE ,I3,16H K-LINKAGE LISTS)
C  NOTE START OF COEFFICIENT MATRIX FILE
      LMAT=LNEXT
C  CHECK VALUE OF N
      IF(N-250)465,465,455
C  N EXCEEDS 250
  455 PRINT 460,N
  460 FORMAT (25H ERROR - NUMBER OF CASES ,I4,12H EXCEEDS 250)
      GO TO 65
C  CHECK VALUE OF MB, MN
C  BRANCH ON DATA TYPE INDICATOR
  465 GO TO (470,485),IDATA
C  BINARY DATA - DOES MB EXCEED 401
  470 IF(MB-401)500,500,475
C  MB EXCEEDS 401
  475 PRINT 480,MB
  480 FORMAT (36H ERROR - NUMBER OF BINARY VARIABLES ,I4,12H EXCEEDS 401
     1)
      GO TO 65
C  NUMERIC DATA - DOES MN EXCEED 200
  485 IF(MN-200)500,500,490
  490 PRINT 495
  495 FORMAT (37H ERROR - NUMBER OF NUMERIC VARIABLES ,I4,12H EXCEEDS 20
     10)
      GO TO 65
C  COMPUTE END OF FILE RECORD LSEC
C  I= STORAGE REQUIRED FOR KMAX K-LISTS
  500 I=KMAX*2*((N+9)/10)
C  J= RECORD NUMBER FOR THE (N+1)TH. SAMPLE - IE THE START OF THE K-LIST
      J=10*(1+(N-1)/10)*((N-1)/10)/2+(N-1-((N-1)/10)*10)*((N-1)/10+1)+
     1LMAT
      LSEC=I+J
C  SET VALUE OF N2 IN DEFINE DISK STATEMENT (REVISE WHEN CHANGING FILE
C  CAPACITY - WORKING CYLINDERS LENGTH ON IBM 1620 MONITOR II SYSTEM)
      N2=4300
      IF (LSEC-N2)515,515,505
C  FILE OVERLAP ERROR DETECTED - PRINT AND ABORT
  505 PRINT 510,LSEC,N2
  510 FORMAT (28H FILE OVERLAP - LAST RECORD ,I5,16H EXCEEDS MAXIMUM,I6)
      GO TO 65
  515 PRINT 520,LSEC
  520 FORMAT (16H DISK FILE USES ,I5,8H RECORDS/)
      RETURN
      END


      SUBROUTINE COEF (COEFF)
C  CALLED BY (CORREL)
C  TO COMPUTE THE SELECTED SIMILARITY/DISSIMILARITY COEFFICIENT ICOEF
C  FOR (A) BINARY DATA IF IDATA=1
C  OR  (B) NUMERIC DATA IF IDATA=2
```

```
C   DIMENSIONS AND COMMON AREA FOR ALL CORREL ROUTINES *****************
C   END OF COMMON RESERVED FOR DISK I/O SUBROUTINE (DISKIO) PARAMETERS
      DIMENSION TEXT(20)
      COMMON N,MB,MN,NPCF,NPC,ISTAND,IMASK,IDATA,ICOEF,ITYPE,KMAX
      COMMON LNDATA,LBDATA,LMEANS,LVARS,LCORS,LEIGS,LEIGVS,LSCORS,LENGS,
     1LFREQS,LNMASK,LBMASK,LMAT,LKLIST,LNEXT,TEXT
C   LIMIT ON MN = 200 ( NUMBER OF NUMERIC VARIABLES )
C   LIMIT ON MB = 401 (NUMBER OF BINARY VARIABLES)
C   LIMIT ON N = 250 ( NUMBER OF POINTS/CASES )
C   LIMIT ON KMAX = 1400/N
C   THIS DIMENSION STATEMENT COUPLED WITH THE USE OF COMMON ENABLES AN
C   EFFECTIVE EQUIVALENCE (XMASK,MASKB),(X,IX),(Y,LIST) TO BE USED
C   WITHOUT EQUAL FANDK.  THUS THE EFFECTIVE (WRITE OVER) DIMENSION OF
C   THE ARRAYS MASKB,IX, AND LIST FOR AN FANDK OF 0805 IS 401
      DIMENSION DAR(1400),KLINK(1400),XMASK(200),MASKB(1),X(200),IX(1),
     1Y(200),LIST(1),C(250),LENG(250)
C   ARRAY COMMON
      COMMON DAR,KLINK,XMASK,MASKB,X,IX,Y,LIST,C,LENG
C   SINGLE VARIABLE COMMON
      COMMON M,LX,PX,PY,LY,IDISK,PM
C   DIMENSIONS AND COMMON AREA FOR ALL CORREL ROUTINES *****************
C   THE FOLLOWING VARIABLES USED BY THE SUBROUTINE ARE DECLARED IN COMMON
C   PM,M = NUMBER OF VARIABLES
C        (A) BINARY DATA
C        (IX(L),L=1,MB) HAS 1 IF ATTRIBUTE L IS UNMASKED AND POSSESSED BY
C                        SAMPLE I, 0 OTHERWISE
C        PX = NUMBER OF UNMASKED ATTRIBUTES POSSESSED BY SAMPLE I
C        (LIST(L),L=1,LY) LIST OF (MASKED AND UNMASKED) ATTRIBUTES
C                           POSSESSED BY SAMPLE J
C        (MASKB(L),L=1,MB) HAS 1 IF ATTRIBUTE L IS UNMASKED, 0 IF MASKED
C        (B)  NUMERIC DATA
C        (X(L),L=1,MN) NUMERIC DATA FOR SAMPLE I
C        (Y(L),L=1,MN) NUMERIC DATA FOR SAMPLE J
C        (XMASK(L),L=1,MN) HAS 1.0 IF NUMERIC VARIABLE L IS UNMASKED,0.0
C                            IF MASKED
C   THE SUBROUTINE ARGUMENT COEFF CONTAINS THE COMPUTED COEFFICIENT VALUE
C   BRANCH ON DATA TYPE INDICATOR
      GO TO (30,40),IDATA
C   BINARY DATA
C   COMPUTE VALUE OF ACELL (NUMBER OF BINARY VARIABLES IN COMMON FOR I,J)
C   PY, THE NUMBER OF UNMASKED ATTRIBUTES POSSESSED BY SAMPLE J
 30 IA=0
      NY=0
      DO 35 L=1,LY
      I=LIST(L)
      NY=NY+MASKB(I)
 35 IA=IA+IX(I)
C   SET ACELL AND PY
      ACELL=IA
      PY=NY
C   NOW BRANCH TO COMPUTE COEFFICIENT ICOEF
 40 GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23
     1,24,25),ICOEF
C   COEFFICIENT 1 - NUMERIC SQUARED EUCLIDEAN DISTANCE
  1 D=0.
      DO 41 L=1,MN
      P=X(L)-Y(L)
 41 D=D+P*P*XMASK(L)
      COEFF=D/PM
      RETURN
```

```
C   COEFFICIENT 2 - BINARY EUCLIDEAN DISTANCE (B+C)/M
  2 COEFF=(PX+PY-2.*ACELL)/PM
    RETURN
C   COEFFICIENT 3 - NUMERIC PRODUCT-MOMENT COERRELATION COEFFICIENT
  3 SUMX=0.
    SUMY=0.
    SSQX=0.
    SSQY=0.
    PRODXY=0.
    DO 43 L=1,MN
    IF (XMASK(L))43,43,42
 42 P=X(L)
    Q=Y(L)
    SUMX=SUMX+P
    SUMY=SUMY+Q
    SSQX=SSQX+P*P
    SSQY=SSQY+Q*Q
    PRODXY=PRODXY+P*Q
 43 CONTINUE
    XNUM=PM*PRODXY-SUMX*SUMY
    DEN=SQRT((PM*SSQX-SUMX*SUMX)*(PM*SSQY-SUMY*SUMY))
    GO TO 60
C   COEFFICIENT 4 - BINARY (A+D)/M
  4 COEFF=(PM+2.*ACELL-PX-PY)/PM
    RETURN
C   COEFFICIENT 5 - BINARY A/(A+B+C)
  5 COEFF=ACELL/(PX+PY-ACELL)
    RETURN
C   COEFFICIENT 6 - BINARY 2A/(2A+B+C)
  6 COEFF=2.*ACELL/(PX+PY)
    RETURN
C   COEFFICIENT 7 - BINARY 2(A+D)/(2(A+D)+(B+C))
  7 P=PM+2.*ACELL-PX-PY
    COEFF=2.*P/(P+PM)
    RETURN
C   COEFFICIENT 8 - BINARY A/(A+2(B+C))
  8 COEFF=ACELL/(2.*(PX+PY)-3.*ACELL)
    RETURN
C   COEFFICIENT 9 - BINARY (A+D)/(A+D+2(B+C))
  9 P=2.*ACELL-PX-PY
    COEFF=(PM+P)/(PM-P)
    RETURN
C   COEFFICIENT 10 - BINARY A/(B+C)
 10 XNUM=ACELL
    DEN=PX+PY-2.*ACELL
    GO TO 60
C   COEFFICIENT 11 - BINARY (A+D)/(B+C)
 11 P=PX+PY-2.*ACELL
    XNUM=PM-P
    DEN=P
    GO TO 60
C   COEFFICIENT 12 - BINARY ((A+D)-(B+C))/M
 12 COEFF=(PM+2.*(2.*ACELL-PX-PY))/PM
    RETURN
C   COEFFICIENT 13 - BINARY A/M
 13 COEFF=ACELL/PM
    RETURN
C   COEFFICIENT 14 - BINARY (A/(A+C)+A/(A+B))/2
 14 COEFF=ACELL*(PX+PY)/(2.*PX*PY)
    RETURN
```

```
C   COEFFICIENT 15 - BINARY (A/(A+C)+A/(A+B)+D/(B+D)+D/(C+D))/4
    15 P=(PM-PX)*(PM-PY)
       IF (P)65,65,44
    44 COEFF=(ACELL*(PX+PY)/(PX*PY)+(PM-PX-PY+ACELL)*(2.*PM-PX-PY)/P)/4.
       RETURN
C   COEFFICIENT 16 - BINARY A/SQRT((A+C)(A+B))
    16 COEFF=ACELL/SQRT(PX*PY)
       RETURN
C   COEFFICIENT 17 - BINARY AD/SQRT((A+B)(A+C)(B+D)(C+D))
    17 P=(PM-PX)*(PM-PY)
       IF (P)65,65,45
    45 COEFF=ACELL*(PM-PX-PY+ACELL)/SQRT(PX*PY*P)
       RETURN
C   COEFFICIENT 18 - BINARY (AD-BC)/SQRT((A+B)(A+C)(B+D)(C+D))
    18 P=(PM-PX)*(PM-PY)
       IF (P)65,65,46
    46 COEFF=(ACELL*PM-PX*PY)/SQRT(PX*PY*P)
       RETURN
C   COEFFICIENT 19 - BINARY (AD-BC)/(AD+BC)
    19 QR=(ACELL*PM)-(PX*PY)
       Q=(ACELL*(PM+2.*(ACELL-PX-PY)))+(PX*PY)
       IF (Q)47,48,47
    47 COEFF=QR/Q
       RETURN
    48 IF (QR)75,70,70
C   COEFFICIENT 20 - SPACE AVAILABLE
    20 CONTINUE
       RETURN
C   COEFFICIENT 21 - SPACE AVAILABLE
    21 CONTINUE
       RETURN
C   COEFFICIENT 22 - SPACE AVAILABLE
    22 CONTINUE
       RETURN
C   COEFFICIENT 23 - SPACE AVAILABLE
    23 CONTINUE
       RETURN
C   COEFFICIENT 24 - SPACE AVAILABLE
    24 CONTINUE
       RETURN
C   COEFFICIENT 25 - SPACE AVAILABLE
    25 CONTINUE
       RETURN
C   THIS SECTION IS UTILISED WHEN THE COEFFICIENT CAN BE INDETERMINATE
C   BRANCH IF ZERO DENOMINATOR
    60 IF (DEN)80,65,80
C   ZERO DENOMINATOR - BRANCH ON COEFFICIENT TYPE INDICATOR
    65 GO TO (70,75),ITYPE
C   SIMILARITY COEFFICIENT
    70 COEFF=999.999
       RETURN
C   DISSIMILARITY COEFFICIENT
    75 COEFF=-999.999
       RETURN
C   NON-ZERO DENOMINATOR
    80 COEFF=XNUM/DEN
       RETURN
       END
```

```
      SUBROUTINE INSERT (IP,JP,CO)
C     CALLED BY (CORREL)
C     FUNCTION - TO COMPARE THE COEFFICIENT CO FOR SAMPLE JP TAKEN WITH
C     SAMPLE IP AND INSERT CO IN THE APPROPRIATE K-LIST DAR(IP,K) FOR IP
C     IF APPROPRIATE.  THE LINKAGE ARRAY KLINK(IP,K) IS ALSO UPDATED.
C     DIMENSIONS AND COMMON AREA FOR ALL CORREL ROUTINES ****************
C     END OF COMMON RESERVED FOR DISK I/O SUBROUTINE (DISKIO) PARAMETERS
      DIMENSION TEXT(20)
      COMMON N,MB,MN,NPCF,NPC,ISTAND,IMASK,IDATA,ICOEF,ITYPE,KMAX
      COMMON LNDATA,LBDATA,LMEANS,LVARS,LCORS,LEIGS,LEIGVS,LSCORS,LENGS,
     1LFREQS,LNMASK,LBMASK,LMAT,LKLIST,LNEXT,TEXT
C     LIMIT ON MN = 200 ( NUMBER OF NUMERIC VARIABLES )
C     LIMIT ON MB = 401 (NUMBER OF BINARY VARIABLES)
C     LIMIT ON N = 250 ( NUMBER OF POINTS/CASES )
C     LIMIT ON KMAX = 1400/N
C     THIS DIMENSION STATEMENT COUPLED WITH THE USE OF COMMON ENABLES AN
C     EFFECTIVE EQUIVALENCE (XMASK,MASKB),(X,IX),(Y,LIST) TO BE USED
C     WITHOUT EQUAL FANDK.  THUS THE EFFECTIVE (WRITE OVER) DIMENSION OF
C     THE ARRAYS MASKB,IX, AND LIST FOR AN FANDK OF 0805 IS 401
      DIMENSION DAR(1400),KLINK(1400),XMASK(200),MASKB(1),X(200),IX(1),
     1Y(200),LIST(1),C(250),LENG(250)
C     ARRAY COMMON
      COMMON DAR,KLINK,XMASK,MASKB,X,IX,Y,LIST,C,LENG
C     SINGLE VARIABLE COMMON
      COMMON M,LX,PX,PY,LY,IDISK,PM
C     DIMENSIONS AND COMMON AREA FOR ALL CORREL ROUTINES ****************
C     ARITHMETIC STATEMENT FUNCTION INDEX(J) IS THE POSITION OF DAR(IP,J)
C     IN THE LINEAR ARRAY DAR(1400)
      INDEX(J)=(J-1)*N+IP
      J=INDEX(KMAX)
C     BRANCH ON COEFFICIENT TYPE INDICATOR
      GO TO (5,10),ITYPE
C     SIMILARITY COEFFICIENT
    5 IF (CO-DAR(J))50,50,15
C     DISSIMILARITY COEFFICIENT
   10 IF (CO-DAR(J))15,50,50
C     THEREFORE CO MUST BE INSERTED IN THE IPTH. ROW OF DAR
C     FIND THE POSITION OF CO IN THE IPTH. ROW OF DAR
   15 DO 30 L=1,KMAX
      J=INDEX(L)
C     BRANCH ON COEFFICIENT TYPE INDICATOR
      GO TO (20,25),ITYPE
C     SIMILARITY COEFFICIENT
   20 IF (CO-DAR(J))30,30,35
C     DISSIMILARITY COEFFICIENT
   25 IF (CO-DAR(J))35,30,30
   30 CONTINUE
C     INSERT CO INTO DAR(IP,L) AND INSERT JP INTO KLINK(IP,L)
   35 Q=DAR(J)
      DAR(J)=CO
      JQ=KLINK(J)
      KLINK(J)=JP
C     IS L LESS THAN KMAX
      IF (L-KMAX)40,50,50
C     L LESS THAN KMAX, SO SHIFT REMAINING HIGH ORDER COEFFICIENTS IN IPTH.
C     ROW OF DAR TO RIGHT, UPDATING KLINK AT THE SAME TIME.
   40 LZ=L+1
      DO 45 I=LZ,KMAX
      J=INDEX(I)
      P=Q
```

```
      LP=JQ
      Q=DAR(J)
      JQ=KLINK(J)
      DAR(J)=P
   45 KLINK(J)=LP
   50 RETURN
      END


C  ROUTINE - MODE
C  AUTHOR - DAVID WISHART, MATHEMATICAL INSTITUTE, UNIVERSITY OF ST.
C  ANDREWS, FIFE, GREAT BRITAIN.  (PROGRAMME WRITTEN - JUNE 1968)
      DEFINE DISK (10,4300)
C  MAIN PROGRAMME (MODE) - PERFORMS THE EXTERNAL CONTROL OF A MODE
C  ANALYSIS HIERARCHICAL FUSION PROCESS
C  THE DECISION TO FUSE EXISTING CLUSTERS OR INTRODUCE A NEW DENSE POINT
C  IS MADE IN STATEMENTS 15 AND 20.
C  CALLS SUBROUTINES INIT,REVISE,OUTPUT,IOFILE,ORDER,GET,MINIMD,INTRO,
C  RECLAS,DISKIO
C  DIMENSIONS AND COMMON AREA FOR ALL (MODE) ROUTINES *****************
C  END OF COMMON RESERVED FOR DISK I/O SUBROUTINE (DISKIO) PARAMETERS
      DIMENSION TEXT(20)
      COMMON N,MB,MN,NPCF,NPC,ISTAND,IMASK,IDATA,ICOEF,ITYPE,KMAX
      COMMON LNDATA,LBDATA,LMEANS,LVARS,LCORS,LEIGS,LEIGVS,LSCORS,LENGS,
     1LFREQS,LNMASK,LBMASK,LMAT,LKLIST,LNEXT,TEXT
C  LIMIT ON MN = 200 ( NUMBER OF NUMERIC VARIABLES )
C  LIMIT ON MB = 401 (NUMBER OF BINARY VARIABLES)
C  LIMIT ON N = 250 ( NUMBER OF POINTS/CASES )
C  LIMIT ON KMAX = 1400/N
C  COMMUNICATION AREA SPECIFICALLY RESERVED FOR (MODE) ROUTINES
C  ARRAY COMMON
      DIMENSION K(250),KP(251),KD(250),KCOP(250),KDCOP(250),IUN(10),
     1PD(251),D(250),C(250),DCOP(250)
      COMMON K,KP,KD,KCOP,KDCOP,IUN,PD,D,C,DCOP
C  SINGLE VARIABLE COMMON
      COMMON IP,IFUSE,LIM,LINK,MINC,KL,KMIN,ISTAGE,DMIN,PMIN,PERC
     1,NUMOUT,COEF,ION,MINFUS
C  DIMENSIONS AND COMMON AREA FOR ALL (MODE) ROUTINES *****************
C  INITIALISE ARRAYS D,K,PD,KP, FIND MATRIX FILE AND READ ANALYSIS
C  PARAMETERS KL,PERC,MINC,ION AND MINFUS
      CALL INIT
C  SET NUMBER OF CLASSIFICATIONS COUNTER TO ZERO
      NUMOUT=0
      IP=1
C  FIND NON-DENSE POINT KMIN WHICH IS NEXT TO BECOME DENSE FOR
C  COEFFICIENT VALUE PMIN
    5 PMIN=PD(IP)
      KMIN=KP(IP)
C  SET FUSION INDICATOR TO ZERO
      IFUSE=0
C  COPY CURRENT VALUES OF CLASSIFICATION ARRAYS K,KD,D
      DO 10 I=1,N
      DCOP(I)=D(I)
      KCOP(I)=K(I)
   10 KDCOP(I)=KD(I)
C  SEARCH FOR LEAST ELEMENT IN D FOR DENSE POINTS - IE LEAST INTER-
C  CLUSTER NUCLEUS LINK
      CALL MINIMD
C  AT EXIT D(LIM) IS LEAST VALUE DMIN IN D FOR DENSE POINTS OR 10.E+90
C  (DISSIMILARITY, -10.E+90 (SIMILARITY) IF NO SUCH LINK EXISTS.
```

```
C   BRANCH ON COEFFICIENT TYPE INDICATOR TO MAKE (INTRODUCE) OR (DIRECT
C   FUSION) DECISION
       GO TO (15,20),ITYPE
C   SIMILARITY COEFFICIENT
    15 IF (DMIN-PMIN)25,30,30
C   DISSIMILARITY COEFFICIENT
    20 IF (DMIN-PMIN)30,30,25
C   NOW INTRODUCE NEW DENE POINT KMIN.
C   SET STAGE INDICATOR
    25 ISTAGE=1
       CALL INTRO
C   NOTE LIMITING COEFFICIENT VALUE PMIN FOR (OUTPUT) PARAMETER
       RAD=PMIN
C   INCREASE NEXT-DENSE COUNTER
       IP=IP+1
C   BRANCH IF DENSE POINT KMIN CAUSES CLUSTER FUSION
       IF (IFUSE-1)40,40,35
C   NOW FUSE TWO EXISTING CLUSTERS - SET STAGE INDICATOR
    30 ISTAGE=2
C   NOTE LIMITING COEFFICIENT VALUE DMIN FOR (OUTPUT) PARAMETER
       RAD=DMIN
       LINK=KD(LIM)
C   RECLASSIFY CLUSTERS CONTAINING POINTS LIM,LINK
       CALL RECLAS (LIM,LINK)
C   SET FUSION INDICATOR
       IFUSE=2
C   NOW REVISE KD,D VALUES AFTER CLUSTER FUSION
    35 CALL REVISE
C   NOW ENTER RESULTS ROUTINE
    40 CALL OUTPUT(RAD)
C   NOTE LAST COEFFICIENT VALUE COEF
       COEF=RAD
C   NOW RETURN FOR NEXT CYCLE
       GO TO 5
       END


       SUBROUTINE GET (ICON,IZ)
C   CALLED BY (MODE)
C   DIMENSIONS AND COMMON AREA FOR ALL (MODE) ROUTINES *****************
C   END OF COMMON RESERVED FOR DISK I/O SUBROUTINE (DISKIO) PARAMETERS
       DIMENSION TEXT(20)
       COMMON N,MB,MN,NPCF,NPC,ISTAND,IMASK,IDATA,ICOEF,ITYPE,KMAX
       COMMON LNDATA,LBDATA,LMEANS,LVARS,LCORS,LEIGS,LEIGVS,LSCORS,LENGS,
     1LFREQS,LNMASK,LBMASK,LMAT,LKLIST,LNEXT,TEXT
C   LIMIT ON MN = 200 ( NUMBER OF NUMERIC VARIABLES )
C   LIMIT ON MB = 401 (NUMBER OF BINARY VARIABLES)
C   LIMIT ON N = 250 ( NUMBER OF POINTS/CASES )
C   LIMIT ON KMAX = 1400/N
C   COMMUNICATION AREA SPECIFICALLY RESERVED FOR (MODE) ROUTINES
C   ARRAY COMMON
       DIMENSION K(250),KP(251),KD(250),KCOP(250),KDCOP(250),IUN(10),
     1PD(251),D(250),C(250),DCOP(250)
       COMMON K,KP,KD,KCOP,KDCOP,IUN,PD,D,C,DCOP
C   SINGLE VARIABLE COMMON
       COMMON IP,IFUSE,LIM,LINK,MINC,KL,KMIN,ISTAGE,DMIN,PMIN,PERC
     1,NUMOUT,COEF,ION,MINFUS
C   DIMENSIONS AND COMMON AREA FOR ALL (MODE) ROUTINES *****************
C   SUBROUTINE TO READ ELEMENTS FROM THE COEFFICIENT MATRIX FILE.
C   WRITTEN IN PACKAGE FORM FOR SIMPLE MODIFICATION TO NON-DISK SYSTEMS.
```

```
      IFN(I)=10*(1+(I-2)/10)*((I-2)/10)/2+(I-2-((I-2)/10)*10)*((I-2)/10+
     11)+LMAT
C  ARITHMETIC STATEMENT FUNCTION WHICH COMPUTES THE DISK RECORD NUMBER
C  CONTAINING THE FIRST SEGMENT OF THE (I)TH. ROW IN A TRIANGULAR MATRIX
C  GIVEN THAT LMAT IS THE DISK RECORD NUMBER OF THE FIRST MATRIX ROW
C  (THE SINGLE COEFFICIENT C(2,1)) AND EACH ROW IS STORED SEQUENTIALLY
C  WITH 10 COEFFICIENTS PER RECORD)
C  BRANCH ON CONTROL INTEGER ICON -
C      ICON = 1 GET THE COMPLETE ROW OF COEFFICIENTS (C(IZ,I),I=1,N) FOR
C               THE (IZ)TH. INDIVIDUAL
C      ICON = 2 GET THE PARTIAL ROW OF COEFFICIENTS (C(IZ,I),I=1,IZ-1)
      GO TO (5,20),ICON
C  GET THE COMPLETE ROW OF COEFFICIENTS (C(IZ,I),I=1,N)
C  INITIALLY SELECT THE ELEMENTS (C(IZ,I),I=IZ+1,N), ONE FROM EACH OF
C  THE ROWS C(I,-) AND THEN PROCEED TO ICON=2 TO COMPLETE THE INPUT BY
C  ADDING THE PARTIAL LIST.
C  FIRST CHECK THAT IZ IS LESS THAN N
    5 IF (IZ-N)10,25,25
C  IZ LESS THAN N
   10 JD=IZ+1
C  THEREFORE READ THE COEFFICIENTS (C(I,IZ),I=JD,N), ONE AT A TIME
C  IF THE ROWS ARE OBTAINED IN REVERSE ORDER, THE ARRAY C CAN BE USED
C  AS A BUFFER AREA SO THAT THE RELEVANT COEFFICIENT IN EACH ROW C(I,IZ)
C  WITH I DECREASING IS RETAINED BY AN OVERLAP AT EACH STAGE.
      LROW=(IZ-1)/10
C  COMPUTE THE LENGTH OF THE SEGMENT, L
      L=IZ-10*LROW
      DO 15 I=JD,N
      MA=N-I+JD
C  THUS WE REQUIRE C(MA,IZ) FROM THE (MA)TH. ROW
C  NOW FIND THE DISK RECORD NUMBER JREC FOR THE LAST TEN-COEFFICIENT
C  SEGMENT OF THIS ROW.
      JREC=IFN(MA)+LROW
C  COMPUTE THE INDIVIDUAL NUMBER IA CORRESPONDING TO THE START OF THIS
C  LAST SEGMENT
      IA=MA-L+1
C  NOW FETCH THE SEGMENT, ALLOWING THE LOWER PART TO WRITE INTO THE C
C  BUFFER AREA AND LEAVING C(MA,IZ) AS AN OVERLAP.
      CALL DISKIO(6,JREC,IBDUMY,LBDUMY,C(IA),L)
   15 CONTINUE
C  NOW CONTINUE TO GET THE PARTIAL LIST.
C  TEST WHETHER IZ EXCEEDS 1
   20 IF (IZ-1)30,30,25
C  FIND THE DISK RECORD OF THE (IZ)TH. ROW
   25 JREC=IFN(IZ)
C  GET THE MATRIX ROW.
      MA=IZ-1
      CALL DISKIO(6,JREC,IBDUMY,LBDUMY,C,MA)
   30 CONTINUE
C  TRACE FEATURE - REMOVE THIS SECTION IF NOT DESIRED *****************
      IF (SENSE SWITCH 2)500,535
  500 GO TO (505,515),ICON
  505 C(IZ)=999.99
      PRINT 510,IZ,(C(I),I=1,N)
  510 FORMAT (55H *****TRACE COMPLETE COEFFICIENT MATRIX ROW FOR SAMPLE
     1,I4/18(15F8.3/))
      GO TO 525
  515 PRINT 520,IZ,(C(I),I=1,MA)
  520 FORMAT (54H *****TRACE PARTIAL COEFFICIENT MATRIX ROW FOR SAMPLE ,
     1I4/18(15F8.3/))
```

```
  525 PRINT 530
  530 FORMAT (31H *****END OF COEFFICIENTS TRACE)
  535 CONTINUE
C     TRACE FEATURE - REMOVE THIS SECTION IF NOT DESIRED *****************
      RETURN
      END


      SUBROUTINE INIT
C     CALLED BY (MODE)
C     DIMENSIONS AND COMMON AREA FOR ALL (MODE) ROUTINES *****************
C     END OF COMMON RESERVED FOR DISK I/O SUBROUTINE (DISKIO) PARAMETERS
      DIMENSION TEXT(20)
      COMMON N,MB,MN,NPCF,NPC,ISTAND,IMASK,IDATA,ICOEF,ITYPE,KMAX
      COMMON LNDATA,LBDATA,LMEANS,LVARS,LCORS,LEIGS,LEIGVS,LSCORS,LENGS,
     1LFREQS,LNMASK,LBMASK,LMAT,LKLIST,LNEXT,TEXT
C     LIMIT ON MN = 200 ( NUMBER OF NUMERIC VARIABLES )
C     LIMIT ON MB = 401 (NUMBER OF BINARY VARIABLES)
C     LIMIT ON N = 250 ( NUMBER OF POINTS/CASES )
C     LIMIT ON KMAX = 1400/N
C     COMMUNICATION AREA SPECIFICALLY RESERVED FOR (MODE) ROUTINES
C     ARRAY COMMON
      DIMENSION K(250),KP(251),KD(250),KCOP(250),KDCOP(250),IUN(10),
     1PD(251),D(250),C(250),DCOP(250)
      COMMON K,KP,KD,KCOP,KDCOP,IUN,PD,D,C,DCOP
C     SINGLE VARIABLE COMMON
      COMMON IP,IFUSE,LIM,LINK,MINC,KL,KMIN,ISTAGE,DMIN,PMIN,PERC
     1,NUMOUT,COEF,ION,MINFUS
C     DIMENSIONS AND COMMON AREA FOR ALL (MODE) ROUTINES *****************
C     SUBROUTINE TO INITIALISE THE MODE-ANALYSIS ROUTINE (MODE)
C     READ THE DISK FILE PARAMETERS
      CALL DISKIO(1,LSDUMY,IBDUMY,LBDUMY,XDUMY,LNDUMY)
C     IDENTIFY THE FILE
      CALL IOFILE
C     IS THE FILE COMPLETE
      IF (KMAX*LMAT*LKLIST)5,5,15
C     FILE INCOMPLETE - ABORT
    5 PRINT 10
   10 FORMAT (24H ERROR - FILE INCOMPLETE/9H JOB ENDS/1H1)
      CALL EXIT
C     READ CARD PARAMETERS
   15 READ 20,KL,MINC,MINFUS,PERC,ION
   20 FORMAT (3I2,F5.0,A1)
C     TEST KL AGAINST LIMITS (0,KMAX)
      IF (KL)25,25,30
C     KL ZERO - SO SET TO ARBITRARY 3
   25 KL=3
C     DOES KL EXCEED KMAX
   30 IF (KL-KMAX)40,40,35
   35 KL=KMAX
C     TEST VALUE OF PERC AGAINST LIMITS (0,1)
   40 IF (PERC)50,50,45
   45 IF (PERC-1.)55,55,50
   50 PERC=.8
C     TEST VALUE OF MINC
   55 IF (MINC)60,60,65
   60 MINC=1
C     PRINT INPUT PARAMETERS
   65 PRINT 70,KL,MINC,PERC,MINFUS
   70 FORMAT (14H1MODE ANALYSIS//16H DENSITY LEVEL =,I3/37H MINIMUM NUMB
```

```
     1ER OF CLUSTERS CONTROL =,I3/27H MINIMUM ENCLOSURE RATIO = ,F5.2/45
     2H FUSION OUTPUT OF CLUSTERS SIZE - MUST EXCEED,I3,13H DENSE POINTS
     3//)
C  READ THE KLTH. K-LINKAGE LIST FROM FILE
      IR=LKLIST+(KL-1)*2*((N+9)/10)
      CALL DISKIO(6,IR,IDUMY,LBDUMY,PD,N)
C  BRANCH ON COEFFICIENT TYPE INDICATOR
      GO TO (75,80),ITYPE
C  SIMILARITY COEFFICIENT
   75 Z=-10.E+90
      GO TO 85
C  DISSIMILARITY COEFFICIENT
   80 Z= 10.E+90
C  SET D, K, KP
   85 DO 90 I=1,N
      D(I)=Z
      KD(I)=0
      K(I)=0
   90 KP(I)=I
C  SET (N+1)TH. VALUES OF PD,KP FOR TOTAL ENCLOSURE OVERLAP
      PD(N+1)=Z
      KP(N+1)=0
C  ORDER PD MAINTAINING KP AS A POINTER
      CALL ORDER(ITYPE,PD,KP,N)
      RETURN
      END


      SUBROUTINE MINIMD
C  CALLED BY (MODE)
C  TO FIND THE LEAST INTER-CLUSTER NUCLEUS LINK DMIN, AND POSITION LIM
C  DIMENSIONS AND COMMON AREA FOR ALL (MODE) ROUTINES *****************
C  END OF COMMON RESERVED FOR DISK I/O SUBROUTINE (DISKIO) PARAMETERS
      DIMENSION TEXT(20)
      COMMON N,MB,MN,NPCF,NPC,ISTAND,IMASK,IDATA,ICOEF,ITYPE,KMAX
      COMMON LNDATA,LBDATA,LMEANS,LVARS,LCORS,LEIGS,LEIGVS,LSCORS,LENGS,
     1LFREQS,LNMASK,LBMASK,LMAT,LKLIST,LNEXT,TEXT
C  LIMIT ON MN = 200 ( NUMBER OF NUMERIC VARIABLES )
C  LIMIT ON MB = 401 (NUMBER OF BINARY VARIABLES)
C  LIMIT ON N = 250 ( NUMBER OF POINTS/CASES )
C  LIMIT ON KMAX = 1400/N
C  COMMUNICATION AREA SPECIFICALLY RESERVED FOR (MODE) ROUTINES
C  ARRAY COMMON
      DIMENSION K(250),KP(251),KD(250),KCOP(250),KDCOP(250),IUN(10),
     1PD(251),D(250),C(250),DCOP(250)
      COMMON K,KP,KD,KCOP,KDCOP,IUN,PD,D,C,DCOP
C  SINGLE VARIABLE COMMON
      COMMON IP,IFUSE,LIM,LINK,MINC,KL,KMIN,ISTAGE,DMIN,PMIN,PERC
     1,NUMOUT,COEF,ION,MINFUS
C  DIMENSIONS AND COMMON AREA FOR ALL (MODE) ROUTINES *****************
C  CORRESPONDING TO D(LIM)
C  SET INITIAL LIMIT VALUE AT EXTREME
      LIM=0
C  BRANCH ON COEFFICIENT TYPE INDICATOR
      GO TO (5,10),ITYPE
C  SIMILARITY COEFFICIENT
    5 DMIN=-10.E+90
      GO TO 15
C  DISSIMILARITY COEFFICIENT
   10 DMIN=10.E+90
   15 DO 40 I=1,N
```

```
      IF (K(I))40,40,20
C  THEREFOR DENSE POINT
C  BRANCH ON COEFFICIENT TYPE INDICATOR
   20 GO TO (25,30),ITYPE
C  SIMILARITY COEFFICIENT
   25 IF (D(I)-DMIN)40,40,35
C  DISSIMILARITY COEFFICIENT
   30 IF (D(I)-DMIN)35,40,40
   35 DMIN=D(I)
      LIM=I
   40 CONTINUE
      RETURN
      END


      SUBROUTINE RECLAS(KA,KB)
C  CALLED BY (MODE)
C  DIMENSIONS AND COMMON AREA FOR ALL (MODE) ROUTINES *****************
C  END OF COMMON RESERVED FOR DISK I/O SUBROUTINE (DISKIO) PARAMETERS
      DIMENSION TEXT(20)
      COMMON N,MB,MN,NPCF,NPC,ISTAND,IMASK,IDATA,ICOEF,ITYPE,KMAX
      COMMON LNDATA,LBDATA,LMEANS,LVARS,LCORS,LEIGS,LEIGVS,LSCORS,LENGS,
     1LFREQS,LNMASK,LBMASK,LMAT,LKLIST,LNEXT,TEXT
C  LIMIT ON MN = 200 ( NUMBER OF NUMERIC VARIABLES )
C  LIMIT ON MB = 401 (NUMBER OF BINARY VARIABLES)
C  LIMIT ON N = 250 ( NUMBER OF POINTS/CASES )
C  LIMIT ON KMAX = 1400/N
C  COMMUNICATION AREA SPECIFICALLY RESERVED FOR (MODE) ROUTINES
C  ARRAY COMMON
      DIMENSION K(250),KP(251),KD(250),KCOP(250),KDCOP(250),IUN(10),
     1PD(251),D(250),C(250),DCOP(250)
      COMMON K,KP,KD,KCOP,KDCOP,IUN,PD,D,C,DCOP
C  SINGLE VARIABLE COMMON
      COMMON IP,IFUSE,LIM,LINK,MINC,KL,KMIN,ISTAGE,DMIN,PMIN,PERC
     1,NUMOUT,COEF,ION,MINFUS
C  DIMENSIONS AND COMMON AREA FOR ALL (MODE) ROUTINES *****************
C  SUBROUTINE TO RECLASSIFY THE CLUSTERS CONTAINING DENSE POINTS KA, KB
C  PROVIDED THAT -
C  A)  KA, KB ARE BOTH DENSE, IE K(KA) AND K(KB) ARE NONZERO
C  B)  KA,KB BELONG TO DIFFERENT CLUSTERS, IE K(KA) AND K(KB) DIFFER
C  THE RESULTANT CLUSTER CODE IS MIN ( K(KA),K(KB) )
      KX=K(KA)
      KY=K(KB)
C  SET KX TO LEAST CLUSTER CODE
      IF (KX-KY)10,30,5
    5 KX=K(KB)
      KY=K(KA)
C  NOW TEST THAT KX,KY ARE NONZERO
   10 IF (KX)30,30,15
C  RECLASSIFY CLUSTER KY INTO KX
   15 DO 25 I=1,N
      IF (K(I)-KY)25,20,25
   20 K(I)=KX
   25 CONTINUE
   30 RETURN
      END


      SUBROUTINE INTRO
C  CALLED BY (MODE)
```

```
C   DIMENSIONS AND COMMON AREA FOR ALL (MODE) ROUTINES *****************
C   END OF COMMON RESERVED FOR DISK I/O SUBROUTINE (DISKIO) PARAMETERS
        DIMENSION TEXT(20)
        COMMON N,MB,MN,NPCF,NPC,ISTAND,IMASK,IDATA,ICOEF,ITYPE,KMAX
        COMMON LNDATA,LBDATA,LMEANS,LVARS,LCORS,LEIGS,LEIGVS,LSCORS,LENGS,
      1LFREQS,LNMASK,LBMASK,LMAT,LKLIST,LNEXT,TEXT
C   LIMIT ON MN = 200 ( NUMBER OF NUMERIC VARIABLES )
C   LIMIT ON MB = 401 (NUMBER OF BINARY VARIABLES)
C   LIMIT ON N = 250 ( NUMBER OF POINTS/CASES )
C   LIMIT ON KMAX = 1400/N
C   COMMUNICATION AREA SPECIFICALLY RESERVED FOR (MODE) ROUTINES
C   ARRAY COMMON
        DIMENSION K(250),KP(251),KD(250),KCOP(250),KDCOP(250),IUN(10),
      1PD(251),D(250),C(250),DCOP(250)
        COMMON K,KP,KD,KCOP,KDCOP,IUN,PD,D,C,DCOP
C   SINGLE VARIABLE COMMON
        COMMON IP,IFUSE,LIM,LINK,MINC,KL,KMIN,ISTAGE,DMIN,PMIN,PERC
      1,NUMOUT,COEF,ION,MINFUS
C   DIMENSIONS AND COMMON AREA FOR ALL (MODE) ROUTINES *****************
C   FUNCTION - TO INTRODUCE A NEW DENSE POINT KMIN WITH (K)TH. LEAST
C   COEFFICIENT PMIN
C   INITIALISE OWN CLUSTER AT START
        K(KMIN)=KMIN
C   GET COEFFICIENTS (C(J),J=1,N) FOR POINT KMIN - SECTION (A) ON FLOW
C   CHART
        CALL GET(1,KMIN)
C   SCAN ALL COEFFICIENTS FOR COMPLETION OF SECTIONS (B) AND (E) ON FLOW
C   CHART
        DO 55 J=1,N
        IF (J-KMIN)5,55,5
      5 IF (K(J)-K(KMIN))10,55,10
C   THEREFORE DIFFERENT CLUSTERS OR J NON-DENSE
     10 IF (K(J))35,35,15
C   THEREFORE J DENSE - NOW EVALUATE SECTION (B)
C   BRANCH ON COEFFICIENT TYPE INDICATOR
     15 GO TO (20,25),ITYPE
C   SIMILARITY COEFFICIENT
     20 IF (C(J)-PMIN)55,55,30
C   DISSIMILARITY COEFFICIENT
     25 IF (C(J)-PMIN)30,55,55
C   NOW FUSE CLUSTERS K(J),K(KMIN)
C   INCREASE FUSION COUNTER
     30 IFUSE=IFUSE+1
C   NOTE FUSION FOR LATER OUTPUT
        IUN(IFUSE)=K(J)
C   RECLASSIFY CLUSTERS K(J),K(KMIN)
        CALL RECLAS(KMIN,J)
        GO TO 55
C   NON-DENSE POINT - CONTINUE WITH SECTION (E) OF FLOW CHART
C   TEST FOR NEW LIMITING COEFFICIENT VALUE
C   BRANCH ON COEFFICIENT TYPE INDICATOR
     35 GO TO (40,45),ITYPE
C   SIMILARITY COEFFICIENT
     40 IF (D(J)-C(J))50,55,55
C   DISSIMILARITY COEFFICIENT
     45 IF (D(J)-C(J))55,55,50
C   NEW LIMITING COEFFICIENT VALUE FOUND
C   RESET D AND KD VALUES
     50 D(J)=C(J)
        KD(J)=KMIN
```

```
C   END OF LOOP
    55 CONTINUE
C   SET UPPER LIMITS ON SEARCH FOR LIMITING D(KMIN)
C   BRANCH ON COEFFICIENT TYPE INDICATOR
       GO TO (60,65),ITYPE
C   SIMILARITY COEFFICIENT
    60 CMIN=-10.E+90
       GO TO 70
C   DISSIMILARITY COEFFICIENT
    65 CMIN=10.E+90
    70 IMIN=0
C   SCAN FOR SECTIONS (C) AND (D) OF FLOW CHART
       DO 130 J=1,N
       IF (J-KMIN)75,130,75
    75 IF (K(J))130,130,80
C   THEREFORE J DENSE POINT
    80 IF (K(J)-K(KMIN))85,130,85
C   THEREFORE DIFFERENT CLUSTER FROM K(KMIN)
    85 IF (J-KMIN)90,130,110
C   SECTION (C) OF FLOW CHART
C   TEST FOR NEW LIMITING COEFFICIENT FOR KMIN
C   BRANCH ON COEFFICIENT TYPE INDICATOR
    90 GO TO (95,100),ITYPE
C   SIMILARITY COEFFICIENT
    95 IF (C(J)-CMIN)130,130,105
C   DISSIMILARITY COEFFICIENT
   100 IF (C(J)-CMIN)105,130,130
C   NEW LIMITING VALUE CMIN
   105 CMIN=C(J)
       IMIN=J
       GO TO 130
C   NOW EVALUATE SECTION (D) OF FLOW CHART
C   TEST FOR NEW LIMITING COEFFICIENT VALUE AGAINST D(J)
C   BRANCH ON COEFFICIENT TYPE INDICATOR
   110 GO TO (115,120),ITYPE
C   SIMILARITY COEFFICIENT
   115 IF (D(J)-C(J))125,130,130
C   DISSIMILARITY COEFFICIENT
   120 IF (D(J)-C(J))130,130,125
C   NEW LIMITING VALUE FOUND
   125 D(J)=C(J)
       KD(J)=KMIN
C   END OF LOOP
   130 CONTINUE
C   UPDATE D(KMIN)
       D(KMIN)=CMIN
       KD(KMIN)=IMIN
       RETURN
       END


       SUBROUTINE REVISE
C   CALLED BY (MODE)
C   DIMENSIONS AND COMMON AREA FOR ALL (MODE) ROUTINES *****************
C   END OF COMMON RESERVED FOR DISK I/O SUBROUTINE (DISKIO) PARAMETERS
       DIMENSION TEXT(20)
       COMMON N,MB,MN,NPCF,NPC,ISTAND,IMASK,IDATA,ICOEF,ITYPE,KMAX
       COMMON LNDATA,LBDATA,LMEANS,LVARS,LCORS,LEIGS,LEIGVS,LSCORS,LENGS,
      1LFREQS,LNMASK,LBMASK,LMAT,LKLIST,LNEXT,TEXT
C   LIMIT ON MN = 200 ( NUMBER OF NUMERIC VARIABLES )
```

```
C  LIMIT ON MB = 401 (NUMBER OF BINARY VARIABLES)
C  LIMIT ON N = 250 ( NUMBER OF POINTS/CASES )
C  LIMIT ON KMAX = 1400/N
C  COMMUNICATION AREA SPECIFICALLY RESERVED FOR (MODE) ROUTINES
C  ARRAY COMMON
      DIMENSION K(250),KP(251),KD(250),KCOP(250),KDCOP(250),IUN(10),
     1PD(251),D(250),C(250),DCOP(250)
      COMMON K,KP,KD,KCOP,KDCOP,IUN,PD,D,C,DCOP
C  SINGLE VARIABLE COMMON
      COMMON IP,IFUSE,LIM,LINK,MINC,KL,KMIN,ISTAGE,DMIN,PMIN,PERC
     1,NUMOUT,COEF,ION,MINFUS
C  DIMENSIONS AND COMMON AREA FOR ALL (MODE) ROUTINES *****************
C  IN THE ARRAY D, D(I) FOR A DENSE POINT I CONTAINS THE LEAST LINK
C  FOR I WITH A DENSE POINT (J,J=1,I-1) BELONGING TO A CLUSTER
C  OTHER THAN THE CLUSTER CONTAINING I. IF, AFTER ANY SERIES OF FUSIONS,
C  THIS LINK NOW REFERS TO A WITHIN-CLUSTER COEFFICIENT THEN D(I)
C  REQUIRES REVISION.
C  THE FUNCTION OF(REVISE)IS TO UPDATE D AND THE CORRESPONDING POINTER
C  ARRAY KD WHERE SUCH WITHIN-CLUSTER D VALUES REQUIRE REVISION.
      DO 80 I=2,N
      IF (K(I))80,80,5
C  THEREFORE POINT I IS DENSE
    5 L=KD(I)
      IF (L)80,80,10
   10 IF (K(I)-K(L))80,15,80
C  LEAST LINK FOR POINT I IS NOW WITHIN-CLUSTER
C  SET EXTREME LIMIT CMIN FOR LEAST LINK SEARCH
C  BRANCH ON COEFFICIENT TYPE INDICATOR
   15 GO TO (20,25),ITYPE
C  SIMILARITY COEFFICIENT
   20 CMIN=-10.E+90
      GO TO 30
C  DISSIMILARITY COEFFICIENT
   25 CMIN=10.E+90
   30 IMIN=0
      ICL=K(I)
C  ICL IS CODE OF CLUSTER CONTAINING POINT I
C  NOW SEARCH TO SEE IF ANY LINKS TO OTHER CLUSTERS EXIST FOR POINT I
C  WITH POINTS (J,J=1,I-1)
      IX=I-1
      DO 38 J=1,IX
      IF (K(J))38,38,35
C  THEREFORE POINT J IS DENSE
   35 IF (K(J)-ICL)40,38,40
   38 CONTINUE
      GO TO 75
C  THEREFORE POINTS I,J BELONG TO DIFFERENT CLUSTERS - GET COEFFICIENTS
C  (C(L),L=1,I-1) FOR POINT I
   40 CALL GET (2,I)
C  NOW SEARCH COMPLETE LIST OF COEFFICIENTS FOR LIMITING VALUE
      DO 70 L=J,IX
      IF (K(L))70,70,45
C  THEREFORE POINT L IS DENSE
   45 IF (K(L)-ICL)50,70,50
C  POINTS I,L BELONG TO DIFFERENT CLUSTERS
C  NOW TEST COEFFICIENT FOR NEW LIMITING VALUE
C  BRANCH ON COEFFICIENT TYPE INDICATOR
   50 GO TO (55,60),ITYPE
C  SIMILARITY COEFFICIENT
   55 IF (C(L)-CMIN)70,70,65
```

```
C   DISSIMILARITY COEFFICIENT
    60 IF (C(L)-CMIN)65,70,70
C   FOUND NEW LIMITING VALUE - RESET CMIN,IMIN
    65 CMIN=C(L)
       IMIN=L
C   END OF LOOP
    70 CONTINUE
C   SET LIMITING VALUES OF D,KD
    75 D(I)=CMIN
       KD(I)=IMIN
    80 CONTINUE
       RETURN
       END


       SUBROUTINE OUTPUT(R)
C   CALLED BY (MODE)
C   FUNCTION - TO OUTPUT
C   (A) THE CLASSIFICATION ARRAY COPIED PRIOR TO THE FUSION OF CLUSTERS
C   (B) INFORMATION CONCERNING SUBSEQUENT FUSIONS AND THE INTRODUCTION OF
C   DENSE POINTS
C   THE FOLLOWING CONTROLS ARE IMPOSED ON THE OUTPUT OF CLASSIFICATIONS
C    (1)  THE SUBSEQUENT FUSION MUST INVOLVE AT LEAST TWO CLUSTERS WHICH
C         CONTAIN MORE THAN MINFUS DENSE POINTS.
C    (2)  NUCLEI CLASSIFICATIONS ARE ONLY GIVEN WHEN ION IS NONZERO
C   DIMENSIONS AND COMMON AREA FOR ALL (MODE) ROUTINES *****************
C   END OF COMMON RESERVED FOR DISK I/O SUBROUTINE (DISKIO) PARAMETERS
       DIMENSION TEXT(20)
       COMMON N,MB,MN,NPCF,NPC,ISTAND,IMASK,IDATA,ICOEF,ITYPE,KMAX
       COMMON LNDATA,LBDATA,LMEANS,LVARS,LCORS,LEIGS,LEIGVS,LSCORS,LENGS,
      1LFREQS,LNMASK,LBMASK,LMAT,LKLIST,LNEXT,TEXT
C   LIMIT ON MN = 200 ( NUMBER OF NUMERIC VARIABLES )
C   LIMIT ON MB = 401 (NUMBER OF BINARY VARIABLES)
C   LIMIT ON N = 250 ( NUMBER OF POINTS/CASES )
C   LIMIT ON KMAX = 1400/N
C   COMMUNICATION AREA SPECIFICALLY RESERVED FOR (MODE) ROUTINES
C   ARRAY COMMON
       DIMENSION K(250),KP(251),KD(250),KCOP(250),KDCOP(250),IUN(10),
      1PD(251),D(250),C(250),DCOP(250)
       COMMON K,KP,KD,KCOP,KDCOP,IUN,PD,D,C,DCOP
C   SINGLE VARIABLE COMMON
       COMMON IP,IFUSE,LIM,LINK,MINC,KL,KMIN,ISTAGE,DMIN,PMIN,PERC
      1,NUMOUT,COEF,ION,MINFUS
C   DIMENSIONS AND COMMON AREA FOR ALL (MODE) ROUTINES *****************
C   DIMENSIONS FOR (OUTPUT) OWN ARRAYS ***********************************
       DIMENSION KK(250),KNUC(250)
C   DIMENSIONS FOR (OUTPUT) OWN ARRAYS ***********************************
C   TRACE EACH CYCLE RESULTS WITH SENSE SWITCH 1 ON ********************
       IF (SENSE SWITCH 1)100,5
   100 PRINT 105
   105 FORMAT (51H TRACE RESULTS AT END OF FUSION/INTRODUCE OPERATION)
       PRINT 110
   110 FORMAT (8H ARRAY K)
       PRINT 115,(K(I),I=1,N)
   115 FORMAT (30I4)
       PRINT 120
   120 FORMAT (8H ARRAY D)
       PRINT 125,(D(I),I=1,N)
   125 FORMAT (1X,10E10.3)
       PRINT 130
```

```
  130 FORMAT (9H ARRAY KD)
      PRINT 115,(KD(I),I=1,N)
      PRINT 135,DMIN,LIM,PMIN,KMIN
  135 FORMAT (8H DMIN = ,E10.3,7H LIM = ,I4,8H PMIN = ,E10.3,8H KMIN = ,
     1I4)
      PRINT 140
  140 FORMAT (13H END OF TRACE)
C  END OF TRACE - REMOVE THIS PROGRAMME SECTION IF NOT REQIRED *********
C  TEST WHETHER FUSION OCCURS (IFUSE GREATER THAN 1)
    5 IF (IFUSE-1)205,205,10
C  TEST IF AT LEAST TWO CLUSTERS COMPRISING MORE THAN MINFUS DENSE
C  POINTS ARE INVOLVED IN THE FUSION
   10 NFUS=0
      GO TO (18,17),ISTAGE
C  SET IUN FOR DIRECT FUSION CASE
   17 IUN(1)=KCOP(LINK)
      IUN(2)=KCOP(LIM)
C  COUNT NUMBER OF DENSE POINTS IN EACH FUSED CLUSTER
   18 DO 14 I=1,IFUSE
      L=IUN(I)
      KDEN=0
      DO 12 J=1,N
      IF (KCOP(J)-L)12,11,12
   11 KDEN=KDEN+1
   12 CONTINUE
      IF (KDEN-MINFUS)14,14,13
   13 NFUS=NFUS+1
   14 CONTINUE
C  TEST IF AT LEAST TWO CLUSTERS ARE INVOLVED
      IF (NFUS-1)205,205,15
C  OUTPUT CLASSIFICATIONS PRIOR TO FUSION
   15 NUMOUT=NUMOUT+1
      NDEN=0
      IENC=0
      NCLUS=0
      DO 16 I=1,N
   16 KK(I)=0
C  ASSEMBLE CLASSIFICATION ARRAYS KCOP,KNUC AND COMPUTE ENCLOSURE RATIO
C  ENCLOS, NUMBER OF CLUSTERS NCLUS AND NUMBER OF DENSE POINTS NDEN
      DO 60 I=1,N
      L=KCOP(I)
      LL=KDCOP(I)
      KNUC(I)=0
      IF (L)35,35,20
   20 NDEN=NDEN+1
      IENC=IENC+1
      IF (KK(L))30,25,30
   25 KK(L)=1
      NCLUS=NCLUS+1
   30 KNUC(I)=L
      GO TO 60
C  BRANCH ON COEFFICIENT TYPE INDICATOR
   35 GO TO (40,45),ITYPE
C  SIMILARITY COEFFICIENT
   40 IF (DCOP(I)-COEF)55,50,50
C  DISSIMILARITY COEFFICIENT
   45 IF (DCOP(I)-COEF)50,50,55
   50 IENC=IENC+1
      KNUC(I)=KCOP(LL)
   55 KCOP(I)=KCOP(LL)
```

```
   60 CONTINUE
C  COMPUTE ENCLOSURE RATIO ENCLOS
      ENCLOS=IENC
      PP=N
      ENCLOS=ENCLOS/PP+.005
      COEF=COEF+.0005
      PRINT 61
   61 FORMAT (/32H CLASSIFICATIONS PRIOR TO FUSION)
C  BRANCH ON NUCLEI  CLASSIFICATION OUTPUT INDICATOR
      IF (ION)90,90,65
C  PRINT AND PUNCH OUT NUCLEI  CLASSIFICATIONS KNUC
   65 PRINT 70,NUMOUT,KL,NDEN,NCLUS,ENCLOS,COEF
      PUNCH 70,NUMOUT,KL,NDEN,NCLUS,ENCLOS,COEF
   70 FORMAT   (2X,18H MODE NUCLEI GROUP,I3,4H KL=,I2,I4,6H DENSE,I3,19H
     1 CLUSTERS ENC RATIO,F5.2,5H COEF,F9.3)
   95 FORMAT (1X,20H MODE COMPLETE GROUP,I3,4H KL=,I2,I4,6H DENSE,I3,19H
     1 CLUSTERS ENC RATIO,F5.2,5H COEF,F9.3)
  200 FORMAT    (20H MODE COMPLETE GROUP,I3,4H KL=,I2,I4,6H DENSE,I3,19H
     1 CLUSTERS ENC RATIO,F5.2,5H COEF,F9.3)
      PRINT 80,(KNUC(I),I=1,N)
      PUNCH 85,(KNUC(I),I=1,N)
   80 FORMAT (20I5)
   85 FORMAT (26I3)
C  PRINT AND PUNCH OUT COMPLETE CLASSIFICATIONS KCOP
   90 PRINT  95,NUMOUT,KL,NDEN,NCLUS,ENCLOS,COEF
      PUNCH 200,NUMOUT,KL,NDEN,NCLUS,ENCLOS,COEF
      PRINT 80,(KCOP(I),I=1,N)
      PUNCH 85,(KCOP(I),I=1,N)
  205 S=R+.0005
C  BRANCH ON CYCLE STAGE INDICATOR
      GO TO (220,210),ISTAGE
C  FUSION OF TWO CLUSTERS AT NEAREST NEIGHBOUR (DIRECT FUSION) CASE
  210 I=K(LINK)
      PRINT 215,KCOP(LINK),KCOP(LIM),S,I
  215 FORMAT (18H NOW FUSE CLUSTERS,2I5,15H AT COEFFICIENT,F9.3,22H. NEW
     1 CLUSTER CODE IS ,I3)
      GO TO 255
C  INTRODUCTION OF A NEW DENSE POINT - DOES IT CAUSE FUSION
  220 IF (IFUSE-1)225,235,245
C  NO FUSION - CREATES NEW CLUSTER NUCLEUS
  225 PRINT 230,KMIN,S,KMIN
  230 FORMAT (27H NOW INTRODUCE DENSE POINT ,I3,15H AT COEFFICIENT,F9.3,
     130H AND FORM NEW CLUSTER NUCLEUS ,I3)
      GO TO 255
C  JOINS ONE EXISTING CLUSTER
  235 I=K(KMIN)
      PRINT 240,KMIN,S,IUN(1),I
  240 FORMAT (27H NOW INTRODUCE DENSE POINT ,I3,15H AT COEFFICIENT,F9.3,
     115H JOINS CLUSTER ,I3,25H AND NEW CLUSTER CODE IS ,I3)
      GO TO 255
C  CAUSES FUSION OF MORE THAN ONE CLUSTER
  245 I=K(KMIN)
      PRINT 250,KMIN,S,I,(IUN(J),J=1,IFUSE)
  250 FORMAT (27H NOW INTRODUCE DENSE POINT ,I3,15H AT COEFFICIENT,F9.3,
     115H FORMS CLUSTER ,I3,25H FROM FUSION OF CLUSTERS ,5I4)
C  TEST FOR END CONDITIONS
C  NUMBER OF CLUSTERS LESS THAN OR EQUAL TO MINC AND ENCLOSURE RATIO
C  GREATER THAN PERC
  255 DO 260 I=1,N
  260 KCOP(I)=0
```

```
      IC=0
      NCLUS=0
      DO 295 I=1,N
      IF (K(I))275,275,265
  265 IC=IC+1
      L=K(I)
      IF (KCOP(L))270,270,295
  270 KCOP(L)=1
      NCLUS=NCLUS+1
      GO TO 295
C  BRANCH ON COEFFICIENT TYPE INDICATOR
  275 GO TO (280,285),ITYPE
C  SIMILARITY COEFFICIENT
  280 IF (D(I)-R)295,290,290
C  DISSIMILARITY COEFFICIENT
  285 IF (D(I)-R)290,290,295
C  POINT ENCLOSED BY SPHERE
  290 IC=IC+1
  295 CONTINUE
      ENCLOS=IC
      P=N
      ENCLOS=ENCLOS/P+.005
C  DOES ENCLOSURE RATIO EXCEED PERC
      IF (ENCLOS-PERC)315,300,300
C  ENCLOSURE RATIO EXCEEDS PERC - DOES NUMBER OF CLUSTERS EXCEED MINC
  300 IF (NCLUS-MINC)305,305,315
C  NUMBER OF CLUSTERS EQUAL TO OR LESS THAN MINC
C  END CONDITIONS SATISFIED - PRINT MESSAGE AND TERMINATE
  305 PRINT 310,NCLUS,ENCLOS
  310 FORMAT (22H NUMBER OF CLUSTERS = ,I3/19H ENCLOSURE RATIO = ,F5.2/2
     15H END CONDITIONS SATISFIED/9H JOB ENDS/1H1)
      CALL EXIT
C  END CONDITIONS NOT SATISFIED - RETURN FOR NEXT CYCLE
  315 RETURN
      END


C  ROUTINE (RESULT)
C  AUTHOR - DAVID WISHART, MATHEMATICAL INSTITUTE, UNIVERSITY OF ST.
C  ANDREWS, FIFE, GREAT BRITAIN.  (PROGRAMME WRITTEN - JUNE 1968)
C  MAIN PROGRAMME RESULT - TO PRINT DATA FILE VALUES AND ANALYSE CLUSTER
C  DIAGNOSTICS FROM CARD INPUT CLASSIFICATION ARRAY ICLA (MODE OUTPUT)
C  CALLS SUBROUTINES IOFILE,ORDER,IOCOEF,DISKIO,IOINP,IONUM,IOBIN,IOCLUS
C  DIMENSIONS AND COMMON AREA FOR ALL (RESULT) ROUTINES ****************
C  END OF COMMON RESERVED FOR DISK I/O SUBROUTINE (DISKIO) PARAMETERS
      DIMENSION TEXT(20)
      COMMON N,MB,MN,NPCF,NPC,ISTAND,IMASK,IDATA,ICOEF,ITYPE,KMAX
      COMMON LNDATA,LBDATA,LMEANS,LVARS,LCORS,LEIGS,LEIGVS,LSCORS,LENGS,
     1LFREQS,LNMASK,LBMASK,LMAT,LKLIST,LNEXT,TEXT
C  LIMIT ON MN = 200 ( NUMBER OF NUMERIC VARIABLES )
C  LIMIT ON MB = 401 (NUMBER OF BINARY VARIABLES)
C  LIMIT ON N = 250 ( NUMBER OF POINTS/CASES )
C  LIMIT ON KMAX = 1400/N
C  COMMUNICATION AREA SPECIFICALLY RESERVED FOR RESULTS ROUTINES
C  ARRAY COMMON
      DIMENSION ICLA(250),XM(200),XSD(200),IFREQ(401),LENG(250)
      COMMON ICLA,XM,XSD,IFREQ,LENG
C  SINGLE VARIABLE COMMON - OUTPUT CONTROL INTEGERS
      COMMON INRAW,INSTAN,INMEAN,INCOR,INEIG,INEIGV,INSCOR,
     1IBRAW,IBFREQ,IBPERC,IMCOEF,IMKLIS,
```

```
      2ICARAY,ICLIST,ICMEAN,ICFREQ,ICPERC,ICRAT
C   DIMENSIONS AND COMMON AREA FOR ALL (RESULT) ROUTINES ****************
      DEFINE DISK (10,4300)
C   INITIALISE AND READ OUTPUT CONTROL PARAMETERS
      CALL IOINP(1)
C   ANALYSE NUMERIC FILE DATA
      CALL IONUM
C   ANALYSE BINARY FILE DATA
      CALL IOBIN
C   ANALYSE COEFFICIENTS MATRIX DATA
      CALL IOCOEF
C   READ CLASSIFICATION ARRAY
    5 CALL IOINP(2)
C   ENTER CLUSTER DIAGNOSIS
      CALL IOCLUS
C   RETURN TO READ A NEW CLASSIFICATION ARRAY
      GO TO 5
C   DUMMY ENTRIES TO LOAD SUBROUTINES ENABLING LOCALISTATION OF
C   IOINP,IONUM,IOBIN, AND IOCLUS (IBM 1620 MONITOR II SYSTEM)
   10 CALL DISKIO
      CALL IOFILE
      CALL ORDER
      Z=SQRT(Z)
      END


      SUBROUTINE IOINP(INP)
C   CALLED BY (RESULT)
C   FUNCTION - INP=1, CONTROL ALL PARAMETRIC READING FOR RESULT ROUTINE
C              INP=2, TO CONTROL READING OF CLASSIFICATION ARRAYS ICLA.
C   DIMENSIONS AND COMMON AREA FOR ALL (RESULT) ROUTINES ****************
C   END OF COMMON RESERVED FOR DISK I/O SUBROUTINE (DISKIO) PARAMETERS
      DIMENSION TEXT(20)
      COMMON N,MB,MN,NPCF,NPC,ISTAND,IMASK,IDATA,ICOEF,ITYPE,KMAX
      COMMON LNDATA,LBDATA,LMEANS,LVARS,LCORS,LEIGS,LEIGVS,LSCORS,LENGS,
     1LFREQS,LNMASK,LBMASK,LMAT,LKLIST,LNEXT,TEXT
C   LIMIT ON MN = 200 ( NUMBER OF NUMERIC VARIABLES )
C   LIMIT ON MB = 401 (NUMBER OF BINARY VARIABLES)
C   LIMIT ON N = 250 ( NUMBER OF POINTS/CASES )
C   LIMIT ON KMAX = 1400/N
C   COMMUNICATION AREA SPECIFICALLY RESERVED FOR RESULTS ROUTINES
C   ARRAY COMMON
      DIMENSION ICLA(250),XM(200),XSD(200),IFREQ(401),LENG(250)
      COMMON ICLA,XM,XSD,IFREQ,LENG
C   SINGLE VARIABLE COMMON - OUTPUT CONTROL INTEGERS
      COMMON INRAW,INSTAN,INMEAN,INCOR,INEIG,INEIGV,INSCOR,
     1IBRAW,IBFREQ,IBPERC,IMCOEF,IMKLIS,
     2ICARAY,ICLIST,ICMEAN,ICFREQ,ICPERC,ICRAT
C   DIMENSIONS AND COMMON AREA FOR ALL (RESULT) ROUTINES ****************
C   SPECIFIC ARRAYS FOR THIS SUBROUTINE **********************************
      DIMENSION CTEXT(20)
C   SPECIFIC ARRAYS FOR THIS SUBROUTINE **********************************
C   BRANCH ON INPUT INDICATOR
      GO TO (5,15),INP
C   READ DISK FILE PARAMETERS
    5 CALL DISKIO(1,LSDUMY,IBDUMY,LBDUMY,XDUMY,LNDUMY)
C   PRINT FILE IDENTIFICATION
      CALL IOFILE
C   READ OUTPUT CONTROL INTEGERS
      READ 10,INRAW,INSTAN,INMEAN,INCOR,INEIG,INEIGV,INSCOR,IBRAW,IBFREQ
```

```
     1,IBPERC,IMCOEF,IMKLIS,ICARAY,ICLIST,ICMEAN,ICFREQ,ICPERC,ICRAT
   10 FORMAT (6A1,I2,2X,3A1,7X,A1,I2,7X,6A1)
      RETURN
C  INPUT OF CLASSIFICATION ARRAY ON CARDS
C  FIRST CARD IS IDENTIFICATION HEADING. SUBSEQUENT CARDS CONTAIN THE
C  ARRAY ICLA WHERE ICLA(I) IS THE CLUSTER CODE CONTAINING THE ITH.
C  SAMPLE, OR ZERO IF THAT SAMPLE IS UNCLASSIFIED
C  TEST WHETHER CLASSIFICATION DIAGNOSIS IS SELECTED
   15 IF (ICARAY+ICLIST+ICMEAN+ICFREQ+ICPERC+ICRAT)20,20,30
   20 PRINT 25
   25 FORMAT (28H NO CLASSIFICATION DIAGNOSIS/9H JOB ENDS/1H1)
      CALL EXIT
C  READ CLASSIFICATION ARRAY IDENTIFICATION
   30 READ 35,(CTEXT(I),I=1,20)
   35 FORMAT (20A4)
C  READ CLASSIFICATION ARRAY ICLA
      READ 40,(ICLA(I),I=1,N)
   40 FORMAT (26I3)
C  PRINT DATA FILE IDENTIFICATION
      PRINT 45,(TEXT(I),I=1,20)
   45 FORMAT (1H1,20A4)
C  PRINT CLASSIFICATION IDENTIFICATION
      PRINT 50,(CTEXT(I),I=1,20)
   50 FORMAT (1X,20A4//)
      RETURN
      END


      SUBROUTINE ORDER (IC,X,KA,N)
C  CALLED BY (RESULT), (MODE) AND (ALLK)
C  SEE ALSO VERSION ORDER(2) - CARDS 37910 TO 38690 - FOR FASTER SORTING
      DIMENSION X(1),KA(1)
C  FUNCTION - TO ORDER THE ARRAY X, HAVING N ELEMENTS, MAINTAINING THE
C  POINTER ARRAY KA SO THAT
C            FOR IC=1   X(1) IS THE GREATEST ELEMENT
C            FOR IC=2   X(1) IS THE LEAST ELEMENT
      IF (N-1)30,30,1
    1 N1=N-1
      DO 25 I=1,N1
      XLIM=X(I)
      KALIM=KA(I)
      JX=I+1
      DO 20 J=JX,N
      GO TO (5,10),IC
    5 IF (X(J)-XLIM)20,20,15
   10 IF (X(J)-XLIM)15,20,20
   15 R=X(J)
      KR=KA(J)
      X(J)=XLIM
      KA(J)=KALIM
      XLIM=R
      KALIM=KR
   20 CONTINUE
      X(I)=XLIM
      KA(I)=KALIM
   25 CONTINUE
   30 RETURN
      END
```

```
      SUBROUTINE IOFILE
C  CALLED BY (RESULT), (MODE) AND (HIERAR)
C  FUNCTION - TO PRINT THE DATA FILE IDENTIFICATION INFORMATION
C  END OF COMMON RESERVED FOR DISK I/O SUBROUTINE (DISKIO) PARAMETERS
      DIMENSION TEXT(20)
      COMMON N,MB,MN,NPCF,NPC,ISTAND,IMASK,IDATA,ICOEF,ITYPE,KMAX
      COMMON LNDATA,LBDATA,LMEANS,LVARS,LCORS,LEIGS,LEIGVS,LSCORS,LENGS,
     1LFREQS,LNMASK,LBMASK,LMAT,LKLIST,LNEXT,TEXT
C  LIMIT ON MN = 200 ( NUMBER OF NUMERIC VARIABLES )
C  LIMIT ON MB = 401 (NUMBER OF BINARY VARIABLES)
C  LIMIT ON N = 250 ( NUMBER OF POINTS/CASES )
C  LIMIT ON KMAX = 1400/N
C  SPECIFIC ARRAYS FOR THIS SUBROUTINE **********************************
      DIMENSION XMASK(200),MASKB(1),LIST(401)
      COMMON XMASK,MASKB,LIST
C  SPECIFIC ARRAYS FOR THIS SUBROUTINE **********************************
C  PRINT TEXTUAL IDENTIFICATION
      PRINT 5,(TEXT(I),I=1,20)
    5 FORMAT (1X,20A4//)
      PRINT 10,N,MB,MN
   10 FORMAT (19H NUMBER OF CASES = ,I4/30H NUMBER OF BINARY VARIABLES =
     1 ,I4/31H NUMBER OF NUMERIC VARIABLES = ,I4//)
C  ARE NUMERIC VARIABLES FILED
      IF (MN)40,40,15
C  BRANCH ON STANDARDISATION INDICATOR
   15 GO TO (20,30),ISTAND
   20 PRINT 25
   25 FORMAT (23H RAW NUMERIC DATA FILED)
      GO TO 40
   30 PRINT 35
   35 FORMAT (22H STANDARD SCORES FILED)
C  TEST IF BINARY DATA FILED
   40 IF (MB)55,55,45
   45 PRINT 50
   50 FORMAT (18H BINARY DATA FILED)
C  TEST IF COMPONENT SCORES FILED
   55 IF (NPCF)80,80,60
   60 PRINT 65,NPCF
   65 FORMAT (I4,23H COMPONENT SCORES FILED)
C  TEST IF SCORES USED IN COMPUTING COEFFICIENTS
      IF (NPC)80,80,70
   70 PRINT 75,ICOEF,NPC
   75 FORMAT (13H COEFFICIENT ,I2,22H COMPUTED USING FIRST ,I3,17H COMPO
     1NENT SCORES)
      GO TO 195
C  TEST IF COEFFICIENTS MATRIX FILED
   80 IF (LMAT)85,85,95
   85 PRINT 90
   90 FORMAT (30H COEFFICIENTS MATRIX NOT FILED)
      GO TO 195
C  BRANCH ON DATA INDICATOR
   95 GO TO (125,100),IDATA
  100 PRINT 105
  105 FORMAT (31H NUMERIC COEFFICIENT CALCULATED)
C  TEST IF MASK USED
      IF (LNMASK)165,165,110
C  FETCH NUMERIC MASK
  110 IR=LNMASK
      CALL DISKIO(6,IR,IBDUMY,LBDUMY,XMASK,MN)
C  ASSEMBLE LIST OF UNMASKED NUMERIC VARIABLES
```

```
      J=0
      DO 120 I=1,MN
      IF (XMASK(I))120,120,115
  115 J=J+1
      LIST(J)=I
  120 CONTINUE
      GO TO 150
  125 PRINT 130
  130 FORMAT (30H BINARY COEFFICIENT CALCULATED)
C  TEST IF MASK USED
      IF (LBMASK)165,165,135
C  FETCH BINARY MASK
  135 IR=LBMASK
      CALL DISKIO(4,IR,MASKB,MB,XDUMY,LNDUMY)
C  ASSEMBLE LIST OF UNMASKED BINARY VARIABLES
      J=0
      DO 145 I=1,MB
      IF (MASKB(I))145,145,140
  140 J=J+1
      LIST(J)=I
  145 CONTINUE
C  PRINT UNMASKED VARIABLES FOR CHECKING PURPOSES
  150 PRINT 155
  155 FORMAT (58H THE FOLLOWING VARIABLES WERE USED TO COMPUTE COEFFICIE
     1NTS)
      PRINT 160,(LIST(I),I=1,J)
  160 FORMAT (30I4)
      GO TO 175
  165 PRINT 170
  170 FORMAT (13H NO MASK USED)
  175 PRINT 180,ICOEF
  180 FORMAT (20H COEFFICIENT NUMBER ,I2,11H CALCULATED)
C  ARE K-LINKAGE LISTS STORED
      IF (KMAX)195,195,185
  185 PRINT 190,KMAX
  190 FORMAT (I4,22H K-LINKAGE LISTS FILED)
  195 RETURN
      END


      SUBROUTINE IONUM
C  CALLED BY (RESULT)
C  FUNCTION - TO PRINT THE FOLLOWING DATA FILE VALUES FOR NUMERIC
C  VARIABLES
C         (A)   RAW NUMERIC DATA
C         (B)   STANDARD SCORES
C         (C)   MEANS AND STANDARD DEVIATIONS
C         (D)   CORRELATIONS
C         (E)   EIGENVALUES, PERCENTAGE VARIANCE AND CUMULATIVE VARIANCE
C         (F)   EIGENVECTORS
C         (G)   FACTOR SCORES
C  MEANS AND VARIANCES ARE READ INTO CORE IF CLUSTER DIAGNOSIS OF
C  NUMERIC VARIABLES IS REQUESTED
C  DIMENSIONS AND COMMON AREA FOR ALL (RESULT) ROUTINES ***************
C  END OF COMMON RESERVED FOR DISK I/O SUBROUTINE (DISKIO) PARAMETERS
      DIMENSION TEXT(20)
      COMMON N,MB,MN,NPCF,NPC,ISTAND,IMASK,IDATA,ICOEF,ITYPE,KMAX
      COMMON LNDATA,LBDATA,LMEANS,LVARS,LCORS,LEIGS,LEIGVS,LSCORS,LENGS,
     1LFREQS,LNMASK,LBMASK,LMAT,LKLIST,LNEXT,TEXT
C  LIMIT ON MN = 200 ( NUMBER OF NUMERIC VARIABLES )
```

```
C   LIMIT ON MB = 401 (NUMBER OF BINARY VARIABLES)
C   LIMIT ON N = 250 ( NUMBER OF POINTS/CASES )
C   LIMIT ON KMAX = 1400/N
C   COMMUNICATION AREA SPECIFICALLY RESERVED FOR RESULTS ROUTINES
C   ARRAY COMMON
      DIMENSION ICLA(250),XM(200),XSD(200),IFREQ(401),LENG(250)
      COMMON ICLA,XM,XSD,IFREQ,LENG
C   SINGLE VARIABLE COMMON - OUTPUT CONTROL INTEGERS
      COMMON INRAW,INSTAN,INMEAN,INCOR,INEIG,INEIGV,INSCOR,
     1IBRAW,IBFREQ,IBPERC,IMCOEF,IMKLIS,
     2ICARAY,ICLIST,ICMEAN,ICFREQ,ICPERC,ICRAT
C   DIMENSIONS AND COMMON AREA FOR ALL (RESULT) ROUTINES ***************
C   SPECIFIC ARRAYS FOR THIS SUBROUTINE *********************************
      DIMENSION X(200),Y(200),COR(630),EIG(200),EIGV(1225),LIST(250)
     1,RMIN(200),RMAX(200)
      COMMON EIGV
      EQUIVALENCE (X,EIGV),(Y,EIGV(201)),(EIG,COR,EIGV(401),RMIN),(RMAX,
     1EIGV(601)),(LIST,ICLA)
C   SPECIFIC ARRAYS FOR THIS SUBROUTINE *********************************
C   IS ANY NUMERIC OUTPUT SELECTED
      IF (INRAW+INSTAN+INMEAN+INCOR+INEIG+INEIGV+INSCOR)5,5,10
C   NO NUMERIC OUTPUT SELECTED - ARE CLUSTER NUMERIC MEANS SELECTED
    5 IF (ICMEAN)330,330,10
C   NUMERIC OUTPUT SELECTED - IS FILE PRESENT
   10 IF (LNDATA)15,15,25
C   ERROR - NUMERIC DATA NOT STORED
   15 PRINT 20
   20 FORMAT (25H NUMERIC DATA FILE ABSENT)
      RETURN
C   NUMERIC FILE PRESENT - GET MEANS AND VARIANCES
   25 IR=LMEANS
      CALL DISKIO(6,IR,IDUMY,LBDUMY,XM,MN)
      IR=LVARS
      CALL DISKIO(6,IR,IDUMY,LBDUMY,XSD,MN)
C   COMPUTE STANDARD DEVIATIONS
      DO 30 I=1,MN
   30 XSD(I)=SQRT(XSD(I))
C   IS PRINTOUT OF MEANS SELECTED
      IF (INMEAN)50,50,35
C   PRINTOUT OF MEANS SELECTED
   35 PRINT 40
   40 FORMAT (//38H NUMERIC MEANS AND STANDARD DEVIATIONS/)
C******   PRINT MEANS AND STANDARD DEVIATIONS
      PRINT 45,(I,XM(I),XSD(I),I=1,MN)
   45 FORMAT (I4,2F14.4)
C******   PRINT MEANS AND STANDARD DEVIATIONS
C   IS RAW DATA LISTING SELECTED
   50 IF (INRAW)95,95,55
C   RAW DATA LISTING SELECTED
   55 PRINT 60
   60 FORMAT (//17H RAW NUMERIC DATA/)
C   INITIALISE RANGES
      DO 61 J=1,MN
      RMIN(J)=XM(J)
   61 RMAX(J)=XM(J)
C   BRANCH ON STANDARD SCORES INDICATOR
      GO TO (65,80),ISTAND
C   RAW NUMERIC DATA IN FILE - NOW PRINT UP
   65 IR=LNDATA
      DO 75 I=1,N
```

```
      CALL DISKIO(6,IR,IBDUMY,LBDUMY,X,MN)
C  REVISE RANGES
      DO 69 J=1,MN
      IF (X(J)-RMAX(J))67,67,66
   66 RMAX(J)=X(J)
      GO TO 69
   67 IF (X(J)-RMIN(J))68,69,69
   68 RMIN(J)=X(J)
   69 CONTINUE
C******    PRINT NUMERIC DATA
      PRINT 70,I,(X(J),J=1,MN)
   70 FORMAT (2H S,I3,3X,8F14.4/25(8X,8F14.4/))
C******    PRINT NUMERIC DATA
   75 CONTINUE
      GO TO 91
C  STANDARD SCORES STORED
   80 IR=LNDATA
      DO 90 I=1,N
      CALL DISKIO(6,IR,IBDUMY,LBDUMY,X,MN)
      DO 85 J=1,MN
   85 X(J)=XSD(J)*X(J)+XM(J)+.00005
C******    PRINT NUMERIC DATA
      PRINT 70,I,(X(J),J=1,MN)
C******    PRINT NUMERIC DATA
C  REVISE RANGES
      DO 89 J=1,MN
      IF (X(J)-RMAX(J))87,87,86
   86 RMAX(J)=X(J)
      GO TO 89
   87 IF (X(J)-RMIN(J))88,89,89
   88 RMIN(J)=X(J)
   89 CONTINUE
   90 CONTINUE
   91 PRINT 92
   92 FORMAT (//9H VARIABLE,14H MINIMUM VALUE,4X,14H MAXIMUM VALUE/)
C******    PRINT RANGES
      PRINT 93,(J,RMIN(J),RMAX(J),J=1,MN)
   93 FORMAT (I5,2F18.4)
C******    PRINT RANGES
C  IS LISTING OF STANDARD SCORES SELECTED
   95 IF (INSTAN)145,145,100
C  PRINTUP OF STANDARD SCORES REQUIRED
  100 IR=LNDATA
      PRINT 105
  105 FORMAT (//24H NUMERIC STANDARD SCORES/)
      GO TO (110,135),ISTAND
C  RAW DATA STORED
  110 DO 130 I=1,N
      CALL DISKIO(6,IR,IDUMY,LBDUMY,X,MN)
      DO 115 J=1,MN
  115 X(J)=(X(J)-XM(J))/XSD(J)
C******    PRINT STANDARD SCORES
      PRINT 120,I,(X(J),J=1,MN)
  120 FORMAT (2H S,I3,3X,14F8.4/15(8X,14F8.4/))
C******    PRINT STANDARD SCORES
  130 CONTINUE
      GO TO 145
  135 DO 140 I=1,N
      CALL DISKIO(6,IR,IDUMY,LBDUMY,X,MN)
C******    PRINT STANDARD SCORES
```

```
      PRINT 120,I,(X(J),J=1,MN)
C******   PRINT STANDARD SCORES
  140 CONTINUE
C  TEST IF PRINTOUT OF NUMERIC VARIABLE CORRELATIONS IS SELECTED
  145 IF (INCOR)195,195,150
C  CORRELATIONS SELECTED - ARE THEY FILED
  150 IF (LCORS)155,155,165
C  ERROR - CORRELATIONS NOT FILED
  155 PRINT 160
  160 FORMAT (23H CORRELATIONS NOT FILED)
      GO TO 195
C  FETCH CORRELATIONS
  165 IR=LCORS
      MCOR=(MN*MN+MN)/2
      CALL DISKIO(6,IR,IBDUMY,LBDUMY,COR,MCOR)
      PRINT 170
  170 FORMAT (//30H NUMERIC VARIABLE CORRELATIONS/)
C******   PRINT NUMERIC CORRELATIONS
      DO 175 I=1,MN
  175 LIST(I)=I
      PRINT 180,(LIST(I),I=1,MN)
  180 FORMAT (1X,19I6)
      DO 185 J=1,MN
      KA=(J*J-J)/2+1
      KB=KA+J-1
  185 PRINT 190,J,(COR(I),I=KA,KB)
  190 FORMAT (I4,19F6.3/10(4X,19F6.3/))
C******   PRINT NUMERIC CORRELATIONS
C  ARE EIGENVALUES SELECTED
  195 IF (INEIG)245,245,200
C  EIGENVALUES SELECTED - ARE THEY STORED
  200 IF (LEIGS)205,205,215
C  EIGENVALUES NOT STORED - PRINT ERROR
  205 PRINT 210
  210 FORMAT (23H EIGENVALUES NOT STORED)
      GO TO 245
C  FETCH EIGENVALUES
  215 IR=LEIGS
      CALL DISKIO(6,IR,IBDUMY,LBDUMY,EIG,MN)
C  COMPUTE PERCENTAGE AND CUMULATIVE VARIANCE
      P=MN
      P=100./P
      Q=0.
      DO 220 I=1,MN
      XX=EIG(I)*P
      Q=Q+XX
      X(I)=XX+.005
  220 Y(I)=Q+.005
C******   PRINT EIGENVALUES, PERCENTAGE AND CUMULATIVE VARIANCE
      PRINT 225
  225 FORMAT (//12H EIGENVALUES/)
      PRINT 230,(EIG(I),I=1,MN)
  230 FORMAT (16F7.2)
      PRINT 235
  235 FORMAT (//20H PERCENTAGE VARIANCE/)
      PRINT 230,(X(I),I=1,MN)
      PRINT 240
  240 FORMAT (//20H CUMULATIVE VARIANCE/)
      PRINT 230,(Y(I),I=1,MN)
C******   PRINT EIGENVALUES, PERCENTAGE AND CUMULATIVE VARIANCE
```

```
C   IS PRINTUP OF EIGENVECTORS SELECTED
  245 IF (INEIGV)285,285,250
C   EIGENVECTORS SELECTED - ARE THEY FILED
  250 IF (LEIGS)255,255,265
C   EIGENVECTORS NOT FILED - ERROR
  255 PRINT 260
  260 FORMAT (23H EIGENVECTORS NOT FILED)
      GO TO 285
C   FETCH EIGENVECTORS
  265 IR=LEIGVS
      L=MN*MN
      CALL DISKIO(6,IR,IBDUMY,LBDUMY,EIGV,L)
      PRINT 270
  270 FORMAT (//23H EIGENVECTORS - BY ROWS/)
C******    PRINTUP OF EIGENVECTORS
      DO 275 J=1,MN
      JA=(J-1)*MN+1
      JB=JA+MN-1
  275 PRINT 280,J,(EIGV(I),I=JA,JB)
  280 FORMAT (7H VECTOR,I3,2X,18F6.3/10(12X,18F6.3/))
C******    PRINTUP OF EIGENVECTORS
C   IS PRINTUP OF COMPONENT SCORES SELECTED
  285 IF (INSCOR)330,330,290
C   SCORES SELECTED - ARE THEY FILED
  290 IF (LSCORS)295,295,305
C   SCORES NOT FILED - ERROR
  295 PRINT 300
  300 FORMAT (27H COMPONENT SCORES NOT FILED)
      GO TO 330
C   NOW PRINTUP SCORES - DOES INSCOR EXCEED NPCF
  305 IF (INSCOR-NPCF)315,315,310
C   INSCOR EXCEEDS NPCF SO MODIFY
  310 INSCOR=NPCF
  315 IR=LSCORS
      PRINT 320,INSCOR
  320 FORMAT (//6H FIRST,I4,14H FACTOR SCORES/)
C******    PRINTUP OF FACTOR SCORES
      DO 325 I=1,N
      ID=IR
      CALL DISKIO(6,ID,IBDUMY,LBDUMY,X,INSCOR)
      IR=IR+(NPCF+9)/10
  325 PRINT 328,I,(X(J),J=1,INSCOR)
  328 FORMAT (2H S,I3,3X,16F7.3/10(8X,16F7.3/))
C******    PRINTUP OF FACTOR SCORES
  330 RETURN
      END


      SUBROUTINE IOBIN
C   CALLED BY (RESULT)
C   FUNCTION - TO PRINT THE FOLLOWING DATA FILE VALUES FOR BINARY
C   VARIABLES
C               (A)  RAW DATA LISTING
C               (B)  VARIABLE FREQUENCIES
C               (C)  VARIABLE PERCENTAGE OCCURRENCES
C   BINARY SAMPLE LIST LENGTHS AND VARIABLE FREQUENCIES ARE ALSO READ
C   INTO CORE IF CLUSTER DIAGNOSIS OF BINARY VARIABLES IS SELECTED
C   DIMENSIONS AND COMMON AREA FOR ALL (RESULT) ROUTINES ***************
C   END OF COMMON RESERVED FOR DISK I/O SUBROUTINE (DISKIO) PARAMETERS
      DIMENSION TEXT(20)
```

```
      COMMON N,MB,MN,NPCF,NPC,ISTAND,IMASK,IDATA,ICOEF,ITYPE,KMAX
      COMMON LNDATA,LBDATA,LMEANS,LVARS,LCORS,LEIGS,LEIGVS,LSCORS,LENGS,
     1LFREQS,LNMASK,LBMASK,LMAT,LKLIST,LNEXT,TEXT
C  LIMIT ON MN = 200 ( NUMBER OF NUMERIC VARIABLES )
C  LIMIT ON MB = 401 (NUMBER OF BINARY VARIABLES)
C  LIMIT ON N = 250 ( NUMBER OF POINTS/CASES )
C  LIMIT ON KMAX = 1400/N
C  COMMUNICATION AREA SPECIFICALLY RESERVED FOR RESULTS ROUTINES
C  ARRAY COMMON
      DIMENSION ICLA(250),XM(200),XSD(200),IFREQ(401),LENG(250)
      COMMON ICLA,XM,XSD,IFREQ,LENG
C  SINGLE VARIABLE COMMON - OUTPUT CONTROL INTEGERS
      COMMON INRAW,INSTAN,INMEAN,INCOR,INEIG,INEIGV,INSCOR,
     1IBRAW,IBFREQ,IBPERC,IMCOEF,IMKLIS,
     2ICARAY,ICLIST,ICMEAN,ICFREQ,ICPERC,ICRAT
C  DIMENSIONS AND COMMON AREA FOR ALL (RESULT) ROUTINES ***************
C  SPECIFIC ARRAYS FOR THIS SUBROUTINE *********************************
      DIMENSION LIST(401),X(401)
      COMMON LIST,X
C  SPECIFIC ARRAYS FOR THIS SUBROUTINE *********************************
      IF (IBRAW+IBFREQ+IBPERC+ICFREQ+ICPERC+ICRAT)25,25,10
C  BINARY INFORMATION REQUIRED - IS BINARY FILE PRESENT
   10 IF (LBDATA)15,15,30
C  BINARY FILE NOT PRESENT - PRINT ERROR
   15 PRINT 20
   20 FORMAT (22H BINARY DATA NOT FILED)
   25 RETURN
C  FETCH BINARY SAMPLE LIST LENGTHS
   30 IR=LENGS
      CALL DISKIO(4,IR,LENG,N,XDUMY,LNDUMY)
C  TEST IF LISTING OF RAW BINARY DATA SELECTED
      IF (IBRAW)55,55,35
C  LISTING SELECTED - PRINT HEADING
   35 PRINT 40
   40 FORMAT (//16H RAW BINARY DATA/)
      IR=LBDATA
      DO 50 I=1,N
      L=LENG(I)
      CALL DISKIO(4,IR,LIST,L,XDUMY,LNDUMY)
C****** PRINT RAW BINARY DATA
       PRINT 45,I,L,(LIST(J),J=1,L)
   45 FORMAT (7H SAMPLE,I4,7H LENGTH,I3,9X,18I5/3(30X,18I5/))
C****** PRINT RAW BINARY DATA
   50 CONTINUE
C  TEST IF FREQUENCIES OR PERCENTAGES REQUIRED
   55 IF (IBFREQ+IBPERC+ICRAT)80,80,60
C  FETCH FREQUENCIES
   60 IR=LFREQS
      CALL DISKIO(4,IR,IFREQ,MB,XDUMY,LNDUMY)
C  TEST IF PRINTOUT OF FREQUENCIES SELECTED
      IF (IBFREQ)80,80,65
C  FREQUENCIES PRINTUP SELECTED
   65 PRINT 70
   70 FORMAT (//28H BINARY VARIABLE FREQUENCIES/)
C******  PRINT BINARY FREQUENCIES
      PRINT 75,(IFREQ(I),I=1,MB)
   75 FORMAT (20I5)
C******  PRINT BINARY FREQUENCIES
C  TEST IF PRINTUP OF PERCENTAGES SELECTED
   80 IF (IBPERC)105,105,85
```

```
C  PRINTUP SELECTED - COMPUTE PERCENTAGES AND SET UP MARKER ARRAY LIST
   85 P=N
      P=100./P
      DO 90 I=1,MB
      Q=IFREQ(I)
      X(I)=Q*P+.05
   90 LIST(I)=I
C  ORDER PERCENTAGES WITH HIGHEST FIRST
      CALL ORDER(1,X,LIST,MB)
      PRINT 95
   95 FORMAT (//43H PERCENTAGE OCCURRENCE FOR BINARY VARIABLES/)
C******  PRINT BINARY PERCENTAGES
      PRINT 100,(LIST(I),X(I),I=1,MB)
  100 FORMAT (10(I6,F6.1))
C******  PRINT BINARY PERCENTAGES
  105 RETURN
      END


      SUBROUTINE IOCOEF
C  CALLED BY (RESULT)
C  FUNCTION - TO PRINT THE FOLLOWING DATA FILE COEFFICIENT VALUES
C              (A)  COEFFICIENT MATRIX
C              (B)  K-LINKAGE LISTS
C  DIMENSIONS AND COMMON AREA FOR ALL (RESULT) ROUTINES ****************
C  END OF COMMON RESERVED FOR DISK I/O SUBROUTINE (DISKIO) PARAMETERS
      DIMENSION TEXT(20)
      COMMON N,MB,MN,NPCF,NPC,ISTAND,IMASK,IDATA,ICOEF,ITYPE,KMAX
      COMMON LNDATA,LBDATA,LMEANS,LVARS,LCORS,LEIGS,LEIGVS,LSCORS,LENGS,
     1LFREQS,LNMASK,LBMASK,LMAT,LKLIST,LNEXT,TEXT
C  LIMIT ON MN = 200 ( NUMBER OF NUMERIC VARIABLES )
C  LIMIT ON MB = 401 (NUMBER OF BINARY VARIABLES)
C  LIMIT ON N = 250 ( NUMBER OF POINTS/CASES )
C  LIMIT ON KMAX = 1400/N
C  COMMUNICATION AREA SPECIFICALLY RESERVED FOR RESULTS ROUTINES
C  ARRAY COMMON
      DIMENSION ICLA(250),XM(200),XSD(200),IFREQ(401),LENG(250)
      COMMON ICLA,XM,XSD,IFREQ,LENG
C  SINGLE VARIABLE COMMON - OUTPUT CONTROL INTEGERS
      COMMON INRAW,INSTAN,INMEAN,INCOR,INEIG,INEIGV,INSCOR,
     1IBRAW,IBFREQ,IBPERC,IMCOEF,IMKLIS,
     2ICARAY,ICLIST,ICMEAN,ICFREQ,ICPERC,ICRAT
C  DIMENSIONS AND COMMON AREA FOR ALL (RESULT) ROUTINES ****************
C  SPECIFIC ARRAYS FOR THIS SUBROUTINE *********************************
      DIMENSION DAR(1400),KLINK(1400),X(40),LIST(40)
      COMMON DAR,KLINK,X,LIST
C  SPECIFIC ARRAYS FOR THIS SUBROUTINE *********************************
C  IS COEFFICIENT MATRIX SELECTED
      IF (IMCOEF)50,50,10
C  COEFFICIENT MATRIX IS SELECTED - IS IT STORED
   10 IF (LMAT)15,15,25
C  MATRIX NOT STORED - PRINT ERROR
   15 PRINT 20
   20 FORMAT (33H MATRIX OF COEFFICIENTS NOT FILED)
      GO TO 50
C  PRINT UP COEFFICIENT MATRIX
   25 IR=LMAT
      PRINT 30
   30 FORMAT (//20H COEFFICIENTS MATRIX/)
      DO 45 I=2,N
```

```
C  FETCH ITH. ROW OF MATRIX
      L=I-1
      CALL DISKIO(6,IR,LDUMY,IBDUMY,DAR,L)
      DO 35 J=1,L
   35 DAR(J)=DAR(J)+.0005
C****** PRINT COEFFICIENTS ROW
      PRINT 40,I,(DAR(J),J=1,L)
   40 FORMAT (2H S,I3,3X,14F8.3/15(8X,14F8.3/))
C****** PRINT COEFFICIENTS ROW
   45 CONTINUE
C  TEST IF PRINTUP OF K-LINKAGE LISTS IS SELECTED
   50 IF (IMKLIS)105,105,55
C  PRINTUP IS SELECTED - ARE LISTS STORED
   55 IF (LKLIST)60,60,70
C  LISTS NOT STORED - PRINT ERROR
   60 PRINT 65
   65 FORMAT (26H K-LINKAGE LISTS NOT FILED)
      GO TO 105
C  TEST IF IMKLIS EXCEEDS KMAX
   70 IF (IMKLIS-KMAX)80,80,75
C  IMKLIS IS GREATER THAN KMAX, SO MODIFY IMKLIS
   75 IMKLIS=KMAX
   80 IR=LKLIST
      PRINT 85,IMKLIS
   85 FORMAT (//I4,39H K-LINKAGE LISTS - (NEAREST NEIGHBOURS)/)
C  READ DAR,KLINK FROM FILE
      DO 90 I=1,IMKLIS
      IZ=(I-1)*N+1
C  READ DAR
      CALL DISKIO(6,IR,IDUMY,LBDUMY,DAR(IZ),N)
C  READ KLINK
      CALL DISKIO(4,IR,KLINK(IZ),N,XDUMY,LNDUMY)
   90 CONTINUE
C******  PRINTUP OF DAR AND KLINK
      DO 100 I=1,N
      DO 92 J=1,IMKLIS
      L=(J-1)*N+I
      X(J)=DAR(L)+.0005
   92 LIST(J)=KLINK(L)
      PRINT 95,I,(X(J),LIST(J),J=1,IMKLIS)
   95 FORMAT (2H S,I3,5X, 9(F8.3,I4)/3(10X, 9(F8.3,I4)/))
  100 CONTINUE
C******  PRINTUP OF DAR AND KLINK
  105 RETURN
      END


      SUBROUTINE IOCLUS
C  CALLED BY (RESULT)
C  FUNCTION - TO PRINT THE FOLLOWING VALUES FOR EACH CLUSTER IN A
C  CLASSIFICATION ARRAY ICLA
C        (A)   THE CLASSIFICATION ARRAY
C        (B)   CASE NUMBER LISTS FOR EACH CLUSTER
C        (C)   NUMERIC MEANS, STANDARD DEVIATIONS AND F-TEST COMPARISON
C              WITH OVERALL POPULATION VARIABLE VARIANCE, T INDICATOR
C        (D)   BINARY VARIABLE FREQUENCIES
C        (E)   BINARY VARIABLE PERCENTAGES
C        (F)   BINARY VARIABLE RATIOS (CLUSTER PERCENTAGE)/(OVERALL
C              PERCENTAGE)
C  DIMENSIONS AND COMMON AREA FOR ALL (RESULT) ROUTINES ***************
```

```
C  END OF COMMON RESERVED FOR DISK I/O SUBROUTINE (DISKIO) PARAMETERS
      DIMENSION TEXT(20)
      COMMON N,MB,MN,NPCF,NPC,ISTAND,IMASK,IDATA,ICOEF,ITYPE,KMAX
      COMMON LNDATA,LBDATA,LMEANS,LVARS,LCORS,LEIGS,LEIGVS,LSCORS,LENGS,
     1LFREQS,LNMASK,LBMASK,LMAT,LKLIST,LNEXT,TEXT
C  LIMIT ON MN = 200 ( NUMBER OF NUMERIC VARIABLES )
C  LIMIT ON MB = 401 (NUMBER OF BINARY VARIABLES)
C  LIMIT ON N = 250 ( NUMBER OF POINTS/CASES )
C  LIMIT ON KMAX = 1400/N
C  COMMUNICATION AREA SPECIFICALLY RESERVED FOR RESULTS ROUTINES
C  ARRAY COMMON
      DIMENSION ICLA(250),XM(200),XSD(200),IFREQ(401),LENG(250)
      COMMON ICLA,XM,XSD,IFREQ,LENG
C  SINGLE VARIABLE COMMON - OUTPUT CONTROL INTEGERS
      COMMON INRAW,INSTAN,INMEAN,INCOR,INEIG,INEIGV,INSCOR,
     1IBRAW,IBFREQ,IBPERC,IMCOEF,IMKLIS,
     2ICARAY,ICLIST,ICMEAN,ICFREQ,ICPERC,ICRAT
C  DIMENSIONS AND COMMON AREA FOR ALL (RESULT) ROUTINES ****************
C  SPECIFIC ARRAYS FOR THIS SUBROUTINE **********************************
      DIMENSION LIST(401),SM(200),SSQ(200),ICO(1),OM(200),OSD(200),
     1OSQ(200),XR(1),OX(201),SX(200),X(1)
      COMMON LIST,SM,SSQ,ICO,OM,OSD,OSQ,OX,SX,X
      EQUIVALENCE (XR,OSQ)
C  SPECIFIC ARRAYS FOR THIS SUBROUTINE **********************************
C  NORMALISE ICLA
      DO 5 I=1,N
    5 LIST(I)=0
      DO 15 I=1,N
      J=ICLA(I)
      IF (J)15,15,10
   10 LIST(J)=1
   15 CONTINUE
      NCLUS=0
      DO 25 I=1,N
      IF (LIST(I))25,25,20
   20 NCLUS=NCLUS+1
      LIST(I)=NCLUS
   25 CONTINUE
      DO 35 I=1,N
      IF (ICLA(I))35,35,30
   30 J=ICLA(I)
      ICLA(I)=LIST(J)
   35 CONTINUE
C  END OF NORMALISATION OF ICLA - IS PRINTOUT OF ICLA SELECTED
      IF (ICARAY)55,55,40
C  PRINTOUT SELECTED
   40 PRINT 45
   45 FORMAT (//21H CLASSIFICATION ARRAY/)
C******   PRINTUP OF CLASSIFICATION ARRAY
      PRINT 50,(ICLA(I),I=1,N)
   50 FORMAT (30I4)
C******   PRINTUP OF CLASSIFICATION ARRAY
C  LOOP FOR EACH CLUSTER
   55 DO 295 IC=1,NCLUS
C  COUNT CLUSTER SIZE AND LIST SAMPLES
      NC=0
      DO 65 I=1,N
      IF (ICLA(I)-IC)65,60,65
   60 NC=NC+1
      LIST(NC)=I
```

```
   65 CONTINUE
C  SET FLOATING RECIPROCAL CLUSTER SIZE
      P=NC
      P=1./P
C  PRINT CLUSTER NUMBER AND SIZE
      PRINT 70,IC,NC
   70 FORMAT(///9H CLUSTER ,I3,19H NUMBER OF CASES = ,I4/)
C  IS LISTING OF SAMPLE NUMBERS SELECTED
      IF (ICLIST)85,85,75
C  LISTING IS SELECTED - NOW PRINTUP
C******   CASE NUMBER LISTING
   75 PRINT 80,(LIST(I),I=1,NC)
   80 FORMAT (13H CASE NUMBERS/9(30I4/))
C******   CASE NUMBER LISTING
C  ARE NUMERIC MEANS SELECTED
   85 IF (ICMEAN)185,185,90
C  NUMERIC MEANS SELECTED - IS NUMERIC FILE PRESENT
   90 IF (LNDATA)185,185,95
C  NUMERIC DATA FILED - SET UP MEAN AND STANDARD DEVIATION SUMMATIONS
   95 DO 100 I=1,MN
      XR(I)=0.
  100 SSQ(I)=0.
C  NOW READ CLUSTER SAMPLES
      IB=(MN+9)/10
      IR=LNDATA-IB
      DO 115 I=1,N
      IR=IR+IB
      IF (ICLA(I)-IC)115,105,115
C  SAMPLE I BELONGS TO CLUSTER IC SO READ
  105 J=IR
      CALL DISKIO(6,J,IBDUMY,LBDUMY,X,MN)
C  ADD TO MEAN AND STANDARD DEVIATION COUNTS
      DO 110 J=1,MN
      XR(J)=XR(J)+X(J)
  110 SSQ(J)=SSQ(J)+X(J)*X(J)
  115 CONTINUE
C  BRANCH ON STANDARD SCORES INDICATOR
      GO TO (120,145),ISTAND
C  RAW DATA STORED - COMPUTE CLUSTER MEANS AND STANDARD DEVIATIONS
  120 DO 140 J=1,MN
      OM(J)=XR(J)*P
      Q=SSQ(J)*P-OM(J)*OM(J)
      IF (Q)125,125,130
  125 OSD(J)=0.
      GO TO 135
  130 OSD(J)=SQRT(Q)
  135 SM(J)=(OM(J)-XM(J))/XSD(J)
      Q=OSD(J)/XSD(J)
      SSQ(J)=Q*Q+.00005
  140 LIST(J)=J
      GO TO 168
C  STANDARD SCORES STORED - COMPUTE MEANS AND STANDARD DEVIATIONS
  145 DO 165 J=1,MN
      SM(J)=XR(J)*P
      Q=SSQ(J)*P-SM(J)*SM(J)
      SSQ(J)=Q+.00005
      IF (Q)150,150,155
  150 Q=0.
      GO TO 160
  155 Q=SQRT(Q)
```

```
  160 OM(J)=SM(J)*XSD(J)+XM(J)
      OSD(J)=Q*XSD(J)
  165 LIST(J)=J
C  ORDER SSQ SO THAT LEAST IS FIRST MAINTAINING LIST AS A POINTER
  168 CALL ORDER(2,SSQ,LIST,MN)
C  PRINT OUT HEADING
      PRINT 170
  170 FORMAT (//60H CLUSTER DIAGNOSIS OF MEANS, STANDARD DEVIATIONS AND
     1F-RATIO,/2(7X,3HVAR,3X,7HF-RATIO,5X,1HT,    10X,7HMN-ORIG,6X,8HSTD-
     2ORIG,1X)/)
C  ORGANISE REMAINING VALUES OSQ,OX AND SX
      DO 175 J=1,MN
      L=LIST(J)
      OSQ(J)=OSD(L)+.00005
      OX(J)=OM(L)+.00005
  175 SX(J)=SM(L)+.00005
C******    PRINTUP DIAGNOSTICS
      PRINT 180,(LIST(I),SSQ(I),SX(I),OX(I),OSQ(I),I=1,MN)
  180 FORMAT (2(I10,2F10.4,2F14.4))
C******    PRINTUP DIAGNOSTICS
C  TEST WHETHER BINARY DATA OUTPUT IS SELECTED
  185 IF (ICFREQ+ICPERC+ICRAT)295,295,190
C  TEST WHETHER BINARY DATA FILED
  190 IF (LBDATA)295,295,195
C  BINARY DATA FILED - SET UP VARIABLE FREQUENCY COUNTER
  195 DO 200 I=1,MB
  200 ICO(I)=0
      IR=LBDATA
C  COUNT BINARY FREQUENCIES FOR CLUSTER
      DO 215 I=1,N
      IF (ICLA(I)-IC)215,205,215
C  SAMPLE I IS IN CLUSTER
  205 J=IR
      LN=LENG(I)
C  FETCH SAMPLE LIST
      CALL DISKIO(4,J,LIST,LN,XDUMY,LNDUMY)
C  ADD TO FREQUENCY COUNTS
      DO 210 J=1,LN
      L=LIST(J)
  210 ICO(L)=ICO(L)+1
  215 IR=IR+(LENG(I)+9)/10
C  ARE BINARY FREQUENCIES SELECTED
      IF (ICFREQ)235,235,220
C  FREQUENCIES SELECTED - PRINTUP LIST
  220 PRINT 225
  225 FORMAT (//28H BINARY VARIABLE FREQUENCIES/)
C******    PRINT BINARY FREQUENCIES
      PRINT 230,(ICO(I),I=1,MB)
  230 FORMAT (20I5)
C******    PRINT BINARY FREQUENCIES
C  TEST IF BINARY PERCENTAGE OCCURRENCES REQUIRED
  235 IF (ICPERC)260,260,240
C  COMPUTE PERCENTAGES
  240 DO 245 I=1,MB
      X(I)=ICO(I)
      X(I)=X(I)*100.*P+.05
  245 LIST(I)=I
C  ORDER PERCENTAGES SO THAT HIGHEST IS FIRST WITH LIST AS POINTER
      CALL ORDER(1,X,LIST,MB)
      PRINT 250
```

```
  250 FORMAT (//43H PERCENTAGE OCCURRENCE FOR BINARY VARIABLES/)
C******   PRINTUP PERCENTAGES
      PRINT255,(LIST(I),X(I),I=1,MB)
  255 FORMAT (10(I6,F6.1))
C******   PRINTUP PERCENTAGES
C  BRANCH IF BINARY FREQUENCY RATIOS SELECTED
  260 IF (ICRAT)295,295,265
C  RATIOS SELECTED - COMPUTE RATIOS
  265 C=N
      C=C*P
      DO 280 I=1,MB
      R=IFREQ(I)
      IF (R)270,270,275
  270 X(I)=0.
      GO TO 280
  275 Q=ICO(I)
      X(I)=Q*C/R+.005
  280 LIST(I)=I
C  ORDER RATIOS SO THAT HIGHEST IS FIRST MAINTAINING LIST AS A POINTER
      CALL ORDER(1,X,LIST,MB)
      PRINT 285
  285 FORMAT (//92H BINARY FREQUENCIES RATIO - (PERCENTAGE OCCURRENCE IN
     1 CLUSTER/PERCENTAGE OCCURRENCE OVERALL)/)
C******   PRINTUP FREQUENCIES RATIO
      PRINT 290,(LIST(I),X(I),I=1,MB)
  290 FORMAT (10(I5,F7.2))
C******   PRINTUP FREQUENCIES RATIO
  295 CONTINUE
      RETURN
      END


      SUBROUTINE READ(NUM,X,MN,LIST,MB)
C  STANDARD READ SUBROUTINE FOR INPUT IN NUMERIC MODE ONLY. DATA SHOULD
C  CONFORM TO THE INPUT SPECIFICATIONS GIVEN BELOW.
      DIMENSION X(1)
      READ 5,(X(I),I=1,MN)
      PRINT 5,(X(I),I=1,MN)
    5 FORMAT (16F5.2)
C  DATA SHOULD HAVE VALUE RANGE X.XX - PUNCHED CARDS CONTAIN ONE VALUE
C  PER 5 COLS. (16 VALUES PER CARD) WITH VALUES PUNCHED ANYWHERE IN THE
C  5 COL. FIELD PROVIDED A DECIMAL POINT IS INCLUDED
      RETURN
      END


C  ROUTINE - HIERAR
C  AUTHOR - DAVID WISHART, MATHEMATICAL INSTITUTE, UNIVERSITY OF ST.
C  ANDREWS, FIFE, GREAT BRITAIN.  (PROGRAMME WRITTEN - AUGUST 1968)
C  REFER TO LANCE, G.N. AND WILLIAMS, W.T. (1967), COMPUTER J.(9) P.373
C  PROGRAMME HIERAR - TO PERFORM THE GENERAL HIERARCHICAL CLASSIFICATION
C  SYSTEM FOR THE SIX TRADITIONAL METHODS NEAREST NEIGHBOUR, FURTHEST
C  NEIGHBOUR, GROUP AVERAGE, CENTROID, MEDIAN AND WARDS OPTIMISATION OF
C  THE ERROR SUM OF SQUARES OBJECTIVE FUNCTION AND LANCE/WILLIAMS
C  FLEXIBLE BETA FORM.
C  CALLS SUBROUTINES START, TRANS, CLASS AND DISKIO
C  SEE SUBROUTINE (START) FOR INPUT PARAMETER SPECIFICATIONS
      DEFINE DISK (10,4300)
C  DIMENSIONS AND COMMON AREAS FOR ALL HIERAR ROUTINES ****************
C  END OF COMMON RESERVED FOR DISK I/O SUBROUTINE (DISKIO) PARAMETERS
```

```
      DIMENSION TEXT(20)
      COMMON N,MB,MN,NPCF,NPC,ISTAND,IMASK,IDATA,ICOEF,ITYPE,KMAX
      COMMON LNDATA,LBDATA,LMEANS,LVARS,LCORS,LEIGS,LEIGVS,LSCORS,LENGS,
     1LFREQS,LNMASK,LBMASK,LMAT,LKLIST,LNEXT,TEXT
C   LIMIT ON MN = 200 ( NUMBER OF NUMERIC VARIABLES )
C   LIMIT ON MB = 401 (NUMBER OF BINARY VARIABLES)
C   LIMIT ON N = 250 ( NUMBER OF POINTS/CASES )
      DIMENSION K(250),KD(250),D(250),X(250),Y(250),ICLA(250)
     1,I1(250),I2(250),COEF(250)
C   ARRAY COMMON
      COMMON K,KD,D,X,Y,ICLA,I1,I2,COEF
C   SINGLE VARIABLE COMMON
      COMMON KTRAN,IMIN,JMIN,PMIN,KG,BETA,KA,KB
C   DIMENSIONS AND COMMON AREAS FOR ALL HIERAR ROUTINES *****************
C   READ FILE PARAMETERS, INPUT PARAMETERS AND INITIALISE K,KD,D,ICLA
      CALL START
C   COMPUTE NUMBER OF CYCLES REQUIRED
      NX=N-1
      DO 40 KG=1,NX
C   TRACE FEATURE - REMOVE THIS SECTION IF NOT DESIRED ******************
      IF (SENSE SWITCH 1)98,96
   98 PRINT 99,(D(I),I=2,N)
   99 FORMAT (1X,12E9.3)
      PRINT 97,(K(I),I=1,N)
      PRINT 97,(KD(I),I=2,N)
      PRINT 97,(ICLA(I),I=1,N)
   97 FORMAT (20I5)
   96 CONTINUE
C   TRACE FEATURE - REMOVE THIS SECTION IF NOT DESIRED ******************
C   SEARCH FOR LEAST BETWEEN CLUSTER LINK
C   BRANCH ON COEFFICIENT TYPE INDICATOR
      GO TO (5,10),ITYPE
C   SIMILARITY COEFFICIENT
    5 PMIN=-10.E+90
      GO TO 15
C   DISSIMILARITY COEFFICIENT
   10 PMIN=10.E+90
   15 JMIN=0
      DO 35 J=2,N
C   TEST IF CLUSTER J ACTIVE
      IF (K(J))35,35,20
C   CLUSTER J ACTIVE - BRANCH ON COEFFICIENT TYPE INDICATOR
   20 GO TO (21,25),ITYPE
C   SIMILARITY COEFFICIENT
   21 IF (D(J)-PMIN)35,35,30
C   DISSIMILARITY COEFFICIENT
   25 IF (D(J)-PMIN)30,35,35
   30 PMIN=D(J)
      JMIN=J
   35 CONTINUE
C   SET IMIN - THE OTHER LINK FOR JMIN
      IMIN=KD(JMIN)
C   TRANSFORM MATRIX AT FUSION IMIN,JMIN
      CALL TRANS
C   REVISE ICLA AND PRINT RESULTS
      CALL CLASS
   40 CONTINUE
C   END OF JOB - DESTROY OBSOLETE MATRIX
   45 LNEXT=LMAT
      LKLIST=0
```

```
      LMAT=0
      LNMASK=0
      LBMASK=0
      NPC=0
      IMASK=0
      IDATA=0
      ICOEF=0
      ITYPE=0
      KMAX=0
C  WRITE DISK FILE PARAMETERS TO ABOLISH MATRIX
      CALL DISKIO(2,LSDUMY,IBDUMY,LBDUMY,XDUMY,LNDUMY)
C  EVALUATE DENDROGRAM TABLE
      CALL DENDRO(I1,I2,COEF,N)
      PRINT 50
   50 FORMAT (9H JOB ENDS/1H1)
      CALL EXIT
      END


      SUBROUTINE START
C  READS DISK FILE PARAMETERS, INPUT CARD PARAMETERS KTRAN,KA,KB,BETA,
C  CHECKS FOR ERRORS, PRINTS FILE IDENTIFICATION AND INITIALISES ARRAYS
C  K,KD,D AND ICLA
C     CALLS SUBROUTINES DISKIO AND IOFILE
C  DIMENSIONS AND COMMON AREAS FOR ALL HIERAR ROUTINES ****************
C  END OF COMMON RESERVED FOR DISK I/O SUBROUTINE (DISKIO) PARAMETERS
      DIMENSION TEXT(20)
      COMMON N,MB,MN,NPCF,NPC,ISTAND,IMASK,IDATA,ICOEF,ITYPE,KMAX
      COMMON LNDATA,LBDATA,LMEANS,LVARS,LCORS,LEIGS,LEIGVS,LSCORS,LENGS,
     1LFREQS,LNMASK,LBMASK,LMAT,LKLIST,LNEXT,TEXT
C  LIMIT ON MN = 200 ( NUMBER OF NUMERIC VARIABLES )
C  LIMIT ON MB = 401 (NUMBER OF BINARY VARIABLES)
C  LIMIT ON N = 250 ( NUMBER OF POINTS/CASES )
      DIMENSION K(250),KD(250),D(250),X(250),Y(250),ICLA(250)
     1,I1(250),I2(250),COEF(250)
C  ARRAY COMMON
      COMMON K,KD,D,X,Y,ICLA,I1,I2,COEF
C  SINGLE VARIABLE COMMON
      COMMON KTRAN,IMIN,JMIN,PMIN,KG,BETA,KA,KB
C  DIMENSIONS AND COMMON AREAS FOR ALL HIERAR ROUTINES ****************
      IFN(I)=10*(1+(I-2)/10)*((I-2)/10)/2+(I-2-((I-2)/10)*10)*((I-2)/10+
     11)+LMAT
C  ARITHMETIC STATEMENT FUNCTION WHICH COMPUTES THE DISK RECORD NUMBER
C  CONTAINING THE FIRST SEGMENT OF THE (I)TH. ROW IN A TRIANGULAR MATRIX
C  GIVEN THAT LMAT IS THE DISK RECORD NUMBER OF THE FIRST MATRIX ROW
C  (THE SINGLE COEFFICIENT C(2,1)) AND EACH ROW IS STORED SEQUENTIALLY
C  WITH 10 COEFFICIENTS PER RECORD)
C  READ DISK FILE PARAMETERS
      CALL DISKIO(1,LSDUMY,IBDUMY,LBDUMY,XDUMY,LNDUMY)
C  PRINT DATA FILE IDENTIFICATION HEADING
      CALL IOFILE
C  READ INPUT DATA CARD
      READ 5,KTRAN,KA,KB,BETA
    5 FORMAT (I1,2I3,F5.0)
C  KTRAN = METHOD CODE
C        = 1 - NEAREST NEIGHBOUR
C        = 2 - FURTHEST NEIGHBOUR
C        = 3 - GROUP AVERAGE
C        = 4 - CENTROID
C        = 5 - MEDIAN
```

```
C         = 6 - WARDS METHOD (SEE WARD,J.H.,1963,J.AMER.STAT.ASS.,(58))
C         = 7 - LANCE/WILLIAMS FLEXIBLE FORM
C   KA = MINIMUM NUMBER OF CLUSTERS FOR WHICH GROUPINGS ARE TO BE OUTPUT
C   KB = MAXIMUM NUMBER OF CLUSTERS FOR WHICH GROUPINGS ARE TO BE OUTPUT
C   BETA = DILATATION COEFFICIENT FOR FLEXIBLE FORM ONLY (LANCE AND
C          WILLIAMS SUGGEST BETA = -.25)
C   TEST KA,KB FOR CORRECT LIMITS
      IF (KA-KB)10,10,20
   10 IF (KA)20,20,15
   15 IF (KB-N+1)40,40,20
C   ERROR IN KA OR KB
   20 PRINT 25,KA,KB
   25 FORMAT (/34H ERROR IN CLUSTER OUTPUT SELECTORS,2I4/)
C   GENERAL ABORT ROUTINE
   30 PRINT 35
   35 FORMAT (17H PHASE TERMINATED/9H JOB ENDS/1H1)
      CALL EXIT
C   CHECK KTRAN FOR ERRORS
   40 IF(KTRAN)50,50,45
   45 IF (KTRAN-7)60,56,50
C   ERROR IN KTRAN
   50 PRINT 55,KTRAN
   55 FORMAT (16H ERROR - METHOD ,I1,14H NOT AVAILABLE/)
      GO TO 30
C   LANCE/WILLIAMS FLEXIBLE FORM - CHECK BETA LESS THAN 1
   56 IF (BETA-1.)60,57,57
   57 PRINT 58,BETA
   58 FORMAT (25H ERROR - VALUE OF BETA = ,F5.2,57H AND MUST BE LESS THA
     1N 1 FOR LANCE/WILLIAMS FLEXIBLE FORM/)
      GO TO 30
C   TEST IF COEFFICIENTS FILED
   60 IF (ICOEF)65,65,75
C   ERROR - MATRIX NOT FILED
   65 PRINT 70
   70 FORMAT (30H COEFFICIENTS MATRIX NOT FILED/)
      GO TO 30
C   BRANCH ON METHOD SELECTOR TO TEST COMPATIBILITY
   75 GO TO (95,95,95,80,80,80,80),KTRAN
C   DSQ METHODS ONLY - TEST ICOEF FOR RIGHT COEFFICIENT
   80 IF (ICOEF-2)95,95,85
C   ERROR - INCOMPATIBLE COEFFICIENT
   85 PRINT 90,KTRAN,ICOEF
   90 FORMAT (16H ERROR - METHOD ,I1,30H INCOMPATIBLE WITH COEFFICIENT,I
     13)
      GO TO 30
C   BRANCH TO PRINT METHOD TITLE
   95 GO TO (100,105,110,115,120,125,126),KTRAN
C   NEAREST NEIGHBOUR
  100 PRINT 101
  101 FORMAT (18H1NEAREST NEIGHBOUR//)
      GO TO 130
  105 PRINT 102
  102 FORMAT (19H1FURTHEST NEIGHBOUR//)
      GO TO 130
  110 PRINT 103
  103 FORMAT (14H1GROUP AVERAGE//)
      GO TO 130
  115 PRINT 104
  104 FORMAT (9H1CENTROID//)
      GO TO 130
```

```
  120 PRINT 106
  106 FORMAT (7H1MEDIAN//)
      GO TO 130
  125 PRINT 107
  107 FORMAT (13H1WARDS METHOD//)
      GO TO 130
  126 PRINT 108,BETA
  108 FORMAT (42H1LANCE/WILLIAMS FLEXIBLE FORM WITH BETA = ,F6.3)
  130 PRINT 135,KA,KB
  135 FORMAT (28H OUTPUT CLASSIFICATIONS FOR ,I4,3H TO,I4,9H CLUSTERS//)
C  SET UP ARRAYS D AND KD
      DO 175 J=2,N
C  BRANCH ON COEFFICIENT TYPE INDICATOR
      GO TO (140,145),ITYPE
C  SIMILARITY COEFFICIENT
  140 DMIN=-10.E+90
      GO TO 150
C  DISSIMILARITY COEFFICIENT
  145 DMIN=10.E+90
  150 IM=0
C  GET JTH. COEFFICIENT ROW X
      IREC=IFN(J)
      L=J-1
      CALL DISKIO(6,IREC,IBDUMY,LBDUMY,X,L)
C  EVALUATE D(J)
      DO 170 I=1,L
C  BRANCH ON COEFFICIENT TYPE INDICATOR
      GO TO (155,160),ITYPE
C  SIMILARITY COEFFICIENT
  155 IF (X(I)-DMIN)170,170,165
C  DISSIMILARITY COEFFICIENT
  160 IF (X(I)-DMIN)165,170,170
  165 DMIN=X(I)
      IM=I
  170 CONTINUE
C  INSERT D VALUE
      D(J)=DMIN
  175 KD(J)=IM
C  INITIALISE ICLA AND K
      DO 180 I=1,N
      ICLA(I)=I
  180 K(I)=1
      RETURN
      END


      SUBROUTINE TRANS
C  AT THE FUSION IMIN,JMIN - TRANS REVISES K,KD, AND D VALUES AND
C  MODIFIES THE COEFFICIENTS MATRIX BY A SUITABLE TRANSFORMATION (KTRAN)
C  CALLS FUNCTION TFUN AND SUBROUTINE DISKIO
C  DIMENSIONS AND COMMON AREAS FOR ALL HIERAR ROUTINES ****************
C  END OF COMMON RESERVED FOR DISK I/O SUBROUTINE (DISKIO) PARAMETERS
      DIMENSION TEXT(20)
      COMMON N,MB,MN,NPCF,NPC,ISTAND,IMASK,IDATA,ICOEF,ITYPE,KMAX
      COMMON LNDATA,LBDATA,LMEANS,LVARS,LCORS,LEIGS,LEIGVS,LSCORS,LENGS,
     1LFREQS,LNMASK,LBMASK,LMAT,LKLIST,LNEXT,TEXT
C  LIMIT ON MN = 200 ( NUMBER OF NUMERIC VARIABLES )
C  LIMIT ON MB = 401 (NUMBER OF BINARY VARIABLES)
C  LIMIT ON N = 250 ( NUMBER OF POINTS/CASES )
-     DIMENSION K(250),KD(250),D(250),X(250),Y(250),ICLA(250)
```

```
     1,I1(250),I2(250),COEF(250)
C   ARRAY COMMON
       COMMON K,KD,D,X,Y,ICLA,I1,I2,COEF
C   SINGLE VARIABLE COMMON
       COMMON KTRAN,IMIN,JMIN,PMIN,KG,BETA,KA,KB
C   DIMENSIONS AND COMMON AREAS FOR ALL HIERAR ROUTINES ****************
       IFN(I)=10*(1+(I-2)/10)*((I-2)/10)/2+(I-2-((I-2)/10)*10)*((I-2)/10+
     11)+LMAT
C   ARITHMETIC STATEMENT FUNCTION WHICH COMPUTES THE DISK RECORD NUMBER
C   CONTAINING THE FIRST SEGMENT OF THE (I)TH. ROW IN A TRIANGULAR MATRIX
C   GIVEN THAT LMAT IS THE DISK RECORD NUMBER OF THE FIRST MATRIX ROW
C   (THE SINGLE COEFFICIENT C(2,1)) AND EACH ROW IS STORED SEQUENTIALLY
C   WITH 10 COEFFICIENTS PER RECORD)
C   UPDATE CLUSTER SIZES AT FUSION IMIN,JMIN
       KX=K(IMIN)
       KY=K(JMIN)
       K(IMIN)=KX+KY
       K(JMIN)=0
C   GET THE COEFFICIENTS ROW FOR CLUSTER JMIN - (Y(J),J=1,(JMIN-1))
       LJMIN=JMIN-1
       IREC=IFN(JMIN)
       CALL DISKIO(6,IREC,IBDUMY,LBDUMY,Y,LJMIN)
C   SET DISK CONTROLS FOR SECTION (B) EVALUATION
       IMD=(IMIN-1)/10
       IA=IMD*10+1
C   SET DISK CONTROLS FOR SECTION (C) EVALUATION
       JMD=(JMIN-1)/10
       ICON=1
       IF (IMD-JMD)10,5,10
     5 ICON=2
    10 JA=JMD*10+1
       LJS=JMIN-JA+1
C   TEST IF IMIN EXCEEDS 1
       IF (IMIN-1)55,55,18
C   TEST OK - ENTER SECTION (A) EVALUATION
C   GET THE COEFFICIENTS ROW FOR CLUSTER IMIN - (X(J),J=1,(IMIN-1))
    18 LIMIN=IMIN-1
       IREC=IFN(IMIN)
       JREC=IREC
       CALL DISKIO(6,IREC,IBDUMY,LBDUMY,X,LIMIN)
C   BRANCH ON COEFFICIENT TYPE INDICATOR
       GO TO (15,20),ITYPE
C   SIMILARITY COEFFICIENT
    15 DMIN=-10.E+90
       GO TO 25
C   DISSIMILARITY COEFFICIENT
    20 DMIN=10.E+90
    25 IM=0
       DO 50 J=1,LIMIN
C   IS CLUSTER J ACTIVE
       IF (K(J))50,50,30
C   CLUSTER J IS ACTIVE SO PERFORM TRANSFORMATION AND MODIFY DMIN,IM
    30 X(J)=TFUN(KX,KY,K(J),X(J),Y(J))
C   BRANCH ON COEFFICIENT TYPE INDICATOR
       GO TO (35,40),ITYPE
C   SIMILARITY COEFFICIENT
    35 IF (X(J)-DMIN)50,50,45
C   DISSIMILARITY COEFFICIENT
    40 IF (X(J)-DMIN)45,50,50
    45 DMIN=X(J)
```

```
      IM=J
   50 CONTINUE
C     UPDATE D(IMIN),KD(IMIN)
      D(IMIN)=DMIN
      KD(IMIN)=IM
C     WRITE TRANSFORMED COEFFICIENTS ROW FOR NEW CLUSTER IMIN
      CALL DISKIO(5,JREC,IBDUMY,LBDUMY,X,LIMIN)
C     END OF SECTION (A) EVALUATION - TEST IF IMIN,JMIN ARE CONSECUTIVE
   55 IF (JMIN-IMIN-1)140,140,60
C     NOT CONSECUTIVE SO ENTER SECTION (B) EVALUATION
   60 IR=IMIN+1
      DO 135 J=IR,LJMIN
C     TEST IF CLUSTER J ACTIVE
      IF (K(J))135,135,65
C     SET UP THE NECESSARY DISK MARKERS AND LENGTHS FOR SECTION (B)
   65 LJ=J-1
      IREC=IFN(J)
      JREC=IREC+IMD
      IB=IA+9
      LA=10
      IF (IB-LJ)67,67,66
   66 IB=LJ
      LA=IB-IA+1
   67 KREC=JREC
C     CLUSTER J ACTIVE - WAS PREVIOUS LEAST LINK FOR J WITH IMIN
      IF (KD(J)-IMIN)112,70,112
C     PREVIOUS LEAST LINK WAS WITH IMIN - GET THE COEFFICIENTS ROW FOR J
   70 IREC=IFN(J)
      CALL DISKIO(6,IREC,IBDUMY,LBDUMY,X,LJ)
C     TRANSFORM X(IMIN)
      X(IMIN)=TFUN(KX,KY,K(J),X(IMIN),Y(J))
C     UPDATE D,KD FOR CLUSTER J
C     BRANCH ON COEFFICIENT TYPE INDICATOR
      GO TO (75,80),ITYPE
C     SIMILARITY COEFFICIENT
   75 DMIN=-10.E+90
      GO TO 85
C     DISSIMILARITY COEFFFICIENT
   80 DMIN=10.E+90
   85 IM=0
      DO 110 I=1,LJ
C     TEST IF CLUSTER I IS ACTIVE
      IF (K(I))110,110,90
C     I ACTIVE - BRANCH ON COEFFICIENT TYPE INDICATOR
   90 GO TO (95,100),ITYPE
C     SIMILARITY COEFFICIENT
   95 IF (X(I)-DMIN)110,110,105
C     DISSIMILARITY COEFFICIENT
  100 IF (X(I)-DMIN)105,110,110
  105 DMIN=X(I)
      IM=I
  110 CONTINUE
      D(J)=DMIN
      KD(J)=IM
C     GO TO WRITE X(IMIN) TO DISK
      GO TO 130
C     PREVIOUS LEAST LINK WAS NOT WITH IMIN - SO GET COEFFICIENT X(IMIN)
  112 CALL DISKIO(6,JREC,IBDUMY,LBDUMY,X(IA),LA)
C     EVALUATE TRANSFORMATION
      X(IMIN)=TFUN(KX,KY,K(J),X(IMIN),Y(J))
```

```
C  TEST IF NEW LEAST LINK IS WITH IMIN - BRANCH ON COEFFICIENT TYPE
      GO TO (115,120),ITYPE
C  SIMILARITY COEFFICIENT
  115 IF (X(IMIN)-D(J))130,130,125
C  DISSIMILARITY COEFFICIENT
  120 IF (X(IMIN)-D(J))125,130,130
  125 D(J)=X(IMIN)
      KD(J)=IMIN
C  WRITE COEFFICIENT X(IMIN) TO JTH. COEFFICIENTS ROW
  130 CALL DISKIO(5,KREC,IBDUMY,LBDUMY,X(IA),LA)
  135 CONTINUE
C  TEST IF JMIN = N
  140 IF (JMIN-N)145,255,255
C  JMIN LESS THAN N - EVALUATE SECTION (C)
  145 IR=JMIN+1
      DO 250 J=IR,N
C  TEST IF CLUSTER J IS ACTIVE
      IF (K(J))250,250,150
C  CLUSTER J ACTIVE - SET UP DISK MARKERS AND LENGTHS FOR SECTION (C)
  150 LJ=J-1
      IREC=IFN(J)
      JREC=IREC+IMD
      IB=IA+9
      LA=10
      IF (IB-LJ)160,160,155
  155 IB=LJ
      LA=IB-IA+1
C  TEST IF PREVIOUS LEAST LINK FOR J WAS WITH EITHER IMIN OR JMIN
  160 IF (KD(J)-IMIN)170,200,165
  165 IF (KD(J)-JMIN)170,200,170
C  PREVIOUS LEAST LINK NOT WITH IMIN OR JMIN, SO GET THE SINGLE
C  COEFFICIENTS X(IMIN),X(JMIN)
C  BRANCH ON SEPARATION INDICATOR
  170 GO TO (171,175),ICON
C  COEFFICIENTS ON SEPARATE RECORDS SO FETCH DOWN INDEPENDENTLY
  171 KREC=JREC
      CALL DISKIO(6,KREC,IBDUMY,LBDUMY,X(IA),LA)
      KREC=IREC+JMD
      CALL DISKIO(6,KREC,IBDUMY,LBDUMY,X(JA),LJS)
      GO TO 180
C  COEFFICIENTS ON SAME 10 WORD SEGMENT SO FETCH DOWN TOGETHER
  175 KREC=JREC
      CALL DISKIO(6,KREC,IBDUMY,LBDUMY,X(IA),LA)
C  NOW TRANSFORM TO X(IMIN)
  180 X(IMIN)=TFUN(KX,KY,K(J),X(IMIN),X(JMIN))
C  UPDATE D,KD - BRANCH ON COEFFICIENT TYPE INDICATOR
      GO TO (185,190),ITYPE
C  SIMILARITY COEFFICIENT
  185 IF (X(IMIN)-D(J))245,245,195
C  DISSIMILARITY COEFFICIENT
  190 IF (X(IMIN)-D(J))195,245,245
  195 D(J)=X(IMIN)
      KD(J)=IMIN
      GO TO 245
C  PREVIOUS LEAST LINK WAS WITH IMIN OR JMIN - GET THE ENTIRE
C  COEFFICIENTS ROW FOR CLUSTER J
  200 CALL DISKIO(6,IREC,IBDUMY,LBDUMY,X,LJ)
C  TRANSFORM X(IMIN)
      X(IMIN)=TFUN(KX,KY,K(J),X(IMIN),X(JMIN))
C  UPDATE D,KD - BRANCH ON COEFFICIENT TYPE INDICATOR
```

```
      GO TO (205,210),ITYPE
C  SIMILARITY COEFFICIENT
  205 DMIN=-10.E+90
      GO TO 215
C  DISSIMILARITY COEFFICIENT
  210 DMIN=10.E+90
  215 IM=0
      DO 240 I=1,LJ
C  IS CLUSTER I ACTIVE
      IF (K(I))240,240,220
C  BRANCH ON COEFFICIENT TYPE INDICATOR
  220 GO TO (225,230),ITYPE
C  SIMILARITY COEFFICIENT
  225 IF (X(I)-DMIN)240,240,235
C  DISSIMILARITY COEFFICIENT
  230 IF (X(I)-DMIN)235,240,240
  235 DMIN=X(I)
      IM=I
  240 CONTINUE
      D(J)=DMIN
      KD(J)=IM
C  WRITE THE SINGLE COEFFICIENT X(IMIN) TO THE JTH. MATRIX ROW
  245 CALL DISKIO(5,JREC,IBDUMY,LBDUMY,X(IA),LA)
  250 CONTINUE
  255 RETURN
      END


      FUNCTION TFUN(KP,KQ,KS,DIP,DIQ)
C  COMPUTES THE COEFFICIENTS TRANSFORMATION FOR SIX TRADITIONAL
C  HIERARCHICAL TECHNIQUES, AND THE LANCE/WILLIAMS FLEXIBLE FORM
C  DIMENSIONS AND COMMON AREAS FOR ALL HIERAR ROUTINES ****************
C  END OF COMMON RESERVED FOR DISK I/O SUBROUTINE (DISKIO) PARAMETERS
      DIMENSION TEXT(20)
      COMMON N,MB,MN,NPCF,NPC,ISTAND,IMASK,IDATA,ICOEF,ITYPE,KMAX
      COMMON LNDATA,LBDATA,LMEANS,LVARS,LCORS,LEIGS,LEIGVS,LSCORS,LENGS,
     1LFREQS,LNMASK,LBMASK,LMAT,LKLIST,LNEXT,TEXT
C  LIMIT ON MN = 200 ( NUMBER OF NUMERIC VARIABLES )
C  LIMIT ON MB = 401 (NUMBER OF BINARY VARIABLES)
C  LIMIT ON N = 250 ( NUMBER OF POINTS/CASES )
      DIMENSION K(250),KD(250),D(250),X(250),Y(250),ICLA(250)
     1,I1(250),I2(250),COEF(250)
C  ARRAY COMMON
      COMMON K,KD,D,X,Y,ICLA,I1,I2,COEF
C  SINGLE VARIABLE COMMON
      COMMON KTRAN,IMIN,JMIN,PMIN,KG,BETA,KA,KB
C  DIMENSIONS AND COMMON AREAS FOR ALL HIERAR ROUTINES ****************
C  SET STANDARD TRANSFORMATION COEFFICIENTS
      AP=.5
      AQ=.5
      G=0.
      RK=KP+KQ
      PK=KP
      QK=KQ
      SK=KS
C  BRANCH ON METHOD CODE TO COMPUTE PARTICULAR TRANSFORMATION COEFFS.
      GO TO (1,2,3,4,5,6,7),KTRAN
C  NEAREST NEIGHBOUR
    1 G=-.5
      BETA=0.
```

```
      GO TO (11,10),ITYPE                                               35762
   11 G=.5                                                              35764
      GO TO 10                                                          35770
C  FURTHEST NEIGHBOUR                                                   35780
    2 G=.5                                                              35790
      BETA=0.                                                           35800
      GO TO (12,10),ITYPE                                               35802
   12 G=-.5                                                             35804
      GO TO 10                                                          35810
C  GROUP AVERAGE                                                        35820
    3 AP=PK/RK                                                          35830
      AQ=QK/RK                                                          35840
      BETA=0.0                                                          35850
      GO TO 10                                                          35860
C  CENTROID                                                             35870
    4 AP=PK/RK                                                          35880
      AQ=QK/RK                                                          35890
      BETA=-AP*AQ                                                       35900
      GO TO 10                                                          35910
C  MEDIAN                                                               35920
    5 BETA=-.25                                                         35930
      GO TO 10                                                          35940
C  WARDS METHOD FOR OPTIMISING THE ERROR SUM OF SQUARES OBJECTIVE FUNCN.35950
    6 AP=(SK+PK)/(SK+RK)                                                35960
      AQ=(SK+QK)/(SK+RK)                                                35970
      BETA=-SK/(SK+RK)                                                  35980
      GO TO 10                                                          35990
C  LANCE/WILLIAMS FLEXIBLE FORM                                         36000
    7 AP=(1.-BETA)/2.                                                   36010
      AQ=AP                                                             36020
      G=0.                                                              36030
C  COMPUTE TRANSFORMED COEFFICIENT                                      36040
   10 TFUN=AP*DIP+AQ*DIQ+BETA*PMIN+G*ABSF(DIP-DIQ)                      36050
      RETURN                                                            36060
      END                                                               36070


      SUBROUTINE CLASS                                                  36080
C  TO REVISE THE CLASSIFICATION ARRAY ICLA AT FUSION IMIN,JMIN, PRINT   36090
C  AND PUNCH ICLA IF CYCLE KG IS AN OUTPUT REQUEST CYCLE                36100
C  DIMENSIONS AND COMMON AREAS FOR ALL HIERAR ROUTINES *****************36110
C  END OF COMMON RESERVED FOR DISK I/O SUBROUTINE (DISKIO) PARAMETERS   36120
      DIMENSION TEXT(20)                                                36130
      COMMON N,MB,MN,NPCF,NPC,ISTAND,IMASK,IDATA,ICOEF,ITYPE,KMAX       36140
      COMMON LNDATA,LBDATA,LMEANS,LVARS,LCORS,LEIGS,LEIGVS,LSCORS,LENGS,36150
     1LFREQS,LNMASK,LBMASK,LMAT,LKLIST,LNEXT,TEXT                       36160
C  LIMIT ON MN = 200 ( NUMBER OF NUMERIC VARIABLES )                    36170
C  LIMIT ON MB = 401 (NUMBER OF BINARY VARIABLES)                       36180
C  LIMIT ON N = 250 ( NUMBER OF POINTS/CASES )                          36190
      DIMENSION K(250),KD(250),D(250),X(250),Y(250),ICLA(250)           36200
     1,I1(250),I2(250),COEF(250)                                        36205
C  ARRAY COMMON                                                         36210
      COMMON K,KD,D,X,Y,ICLA,I1,I2,COEF                                 36220
C  SINGLE VARIABLE COMMON                                               36230
      COMMON KTRAN,IMIN,JMIN,PMIN,KG,BETA,KA,KB                         36240
C  DIMENSIONS AND COMMON AREAS FOR ALL HIERAR ROUTINES *****************36250
C  REVISE ICLA AT FUSION IMIN,JMIN                                      36260
      DO 10 I=1,N                                                       36270
      IF (ICLA(I)-JMIN)10,8,10                                          36280
    8 ICLA(I)=IMIN                                                      36290
```

```
   10 CONTINUE
      R=PMIN+.0005
      NC=N-KG
C  NOTE FUSED CLUSTERS FOR DENDROGRAM LIST
      I1(KG)=IMIN
      I2(KG)=JMIN
      COEF(KG)=R
C  IS THIS AN OUTPUT CYCLE
      IF (NC-KB)9,20,18
    9 IF (NC-KA)18,20,20
C  NOT AN OUTPUT CYCLE SO PRINT FUSION MESSAGE
   18 PRINT 19,KG,IMIN,JMIN,R,NC,IMIN
   19 FORMAT (6H CYCLE,I4,16H NOW FUSE POINTS,2I4,15H AT COEFFICIENT,F8.
     13,3H - ,I3,33H CLUSTERS AND NEW CLUSTER CODE IS,I4)
      RETURN
C  THIS IS AN OUTPUT CYCLE, SO BRANCH ON METHOD INDICATOR
   20 GO TO (1,2,3,4,5,6,7),KTRAN
    1 PRINT 11,KG,IMIN,JMIN,R,NC
      PUNCH 11,KG,IMIN,JMIN,R,NC
   11 FORMAT (24H NEAREST NEIGHBOUR GROUP, I4,12H FUSE POINTS,2I4,8H AT
     1COEF,F8.3,I4,9H CLUSTERS)
      GO TO 25
    2 PRINT 12,KG,IMIN,JMIN,R,NC
      PUNCH 12,KG,IMIN,JMIN,R,NC
   12 FORMAT (25H FURTHEST NEIGHBOUR GROUP,I4,12H FUSE POINTS,2I4,8H AT
     1COEF,F8.3,I4,9H CLUSTERS)
      GO TO 25
    3 PRINT 13,KG,IMIN,JMIN,R,NC
      PUNCH 13,KG,IMIN,JMIN,R,NC
   13 FORMAT (20H GROUP AVERAGE GROUP,      I4,12H FUSE POINTS,2I4,8H AT
     1COEF,F8.3,I4,9H CLUSTERS)
      GO TO 25
    4 PRINT 14,KG,IMIN,JMIN,R,NC
      PUNCH 14,KG,IMIN,JMIN,R,NC
   14 FORMAT (15H CENTROID GROUP,           I4,12H FUSE POINTS,2I4,8H AT
     1COEF,F8.3,I4,9H CLUSTERS)
      GO TO 25
    5 PRINT 15,KG,IMIN,JMIN,R,NC
      PUNCH 15,KG,IMIN,JMIN,R,NC
   15 FORMAT (13H MEDIAN GROUP,             I4,12H FUSE POINTS,2I4,8H AT
     1COEF,F8.3,I4,9H CLUSTERS)
      GO TO 25
    6 PRINT 16,KG,IMIN,JMIN,R,NC
      PUNCH 16,KG,IMIN,JMIN,R,NC
   16 FORMAT (19H WARDS METHOD GROUP,       I4,12H FUSE POINTS,2I4,8H AT
     1COEF,F8.3,I4,9H CLUSTERS)
      GO TO 25
    7 PRINT 17,KG,IMIN,JMIN,R,NC
      PUNCH 17,KG,IMIN,JMIN,R,NC
   17 FORMAT (20H FLEXIBLE FORM GROUP,      I4,12H FUSE POINTS,2I4,8H AT
     1COEF,F8.3,I4,9H CLUSTERS)
C  PRINT AND PUNCH ICLA
   25 PRINT 30,(ICLA(I),I=1,N)
      PUNCH 35,(ICLA(I),I=1,N)
   30 FORMAT (20I5)
   35 FORMAT (26I3)
      RETURN
      END
```

```
      SUBROUTINE DENDRO(I1,I2,COEF,N)
C   FUNCTION - TO ARRANGE AN ORDERING OF THE SAMPLE CODES IN LIST SO THAT
C   A DENDROGRAM CAN BE CONSRUCTED WITHOUT INTERSECTING STEMS. PROVISION
C   IS ALSO MADE TO PUNCH THE HIERARCHICAL FUSION DATA ON TO CARDS.
      DIMENSION I1(1),I2(1),LIST(250),COEF(250)
      N1=N-1
      N2=N-2
      LIST(1)=I1(N1)
      LIST(2)=I2(N1)
      DO 25 I=1,N2
      J=N1-I
      IA=I1(J)
      IB=I2(J)
      IX=I+1
      DO 5 J=1,IX
      IF (LIST(J)-IA)5,10,5
    5 CONTINUE
   10 JX=J+1
      IF (J-IX)15,25,25
   15 LX=I+J+3
      DO 20 K=JX,IX
      L=LX-K
   20 LIST(L)=LIST(L-1)
   25 LIST(JX)=IB
      PRINT 30
   30 FORMAT (17H1DENDROGRAM TABLE,80X,15H FUSION SUMMARY/)
      PRINT 35,(LIST(I),I1(I),I2(I),COEF(I),I=1,N1),LIST(N)
   35 FORMAT (I4,2H .,9X,1H.,9X,1H.,9X,1H.,9X,1H.,9X,1H.,9X,1H.,9X,1H.,9
     1X,1H.,9X,1H.,2I5,F10.3)
C   USE THIS SECTION TO PUNCH FUSION DATA ON TO CARDS SUITABLE FOR OFF-
C   LINE DENDROGRAM PLOTTING OR HIERARCHY RECONSTRUCTION.  REMOVE THE C,S
C   IN COLUMN 1 OF THE NEXT 6 CARDS
C     PUNCH 40,N
C     PUNCH 40,(LIST(I),I=1,N)
C     PUNCH 40,(I1(I),I2(I),I=1,N1)
C     PUNCH 45,(COEF(I),I=1,N1)
C  40 FORMAT (26I3)
C  45 FORMAT (10F8.3)
      RETURN
      END


C   ROUTINE - ALLK
C   AUTHOR - DAVID WISHART, MATHEMATICAL INSTITUTE, UNIVERSITY OF ST.
C   ANDREWS, FIFE, GREAT BRITAIN.  (PROGRAMME WRITTEN - AUGUST 1968)
C   THIS PROGRAMME SELECTS THE (N-1) SIMILARITY COEFFICIENTS, FROM THE
C   STORED TRIANGULAR MATRIX, FOR EACH OF THE CASES NUMBERED ISTART TO
C   IEND, ORDERS THE COEFFICIENTS ACCORDING TO SIMILARITY, AND PRINTS THE
C   FIRST NX FOR EACH CASE
C   CONTROL CARDS CAN BE INPUT SEQUENTIALLY FOR SEPARATE INVESTIGATION
C   OF CASE BLOCKS - AND PROGRAMME ENDS ON A TRAP ERROR.
      DEFINE DISK (10,4300)
C   DIMENSIONS AND COMMON AREA FOR ALL (MODE) ROUTINES *****************
C
C
C   END OF COMMON RESERVED FOR THE GENERAL DISK INPUT/OUTPUT SUBROUTINE
C   (DISKIO) PARAMETERS
C
      DIMENSION TEXT(20)
      COMMON N,MB,MN,NPCF,NPC,ISTAND,IMASK,IDATA,ICOEF,ITYPE,KMAX
```

```
      COMMON LNDATA,LBDATA,LMEANS,LVARS,LCORS,LEIGS,LEIGVS,LSCORS,LENGS,
     1LFREQS,LNMASK,LBMASK,LMAT,LKLIST,LNEXT,TEXT
C
C  LIMIT ON MN = 200 ( NUMBER OF NUMERIC VARIABLES )
C  LIMIT ON MB = 401 (NUMBER OF BINARY VARIABLES)
C  LIMIT ON N = 250 ( NUMBER OF POINTS/CASES )
C  LIMIT ON KMAX = 1400/N
C
C  COMMUNICATION AREA SPECIFICALLY RESERVED FOR (MODE) ROUTINES
C
C  ARRAY COMMON
C
      DIMENSION K(250),KP(251),KD(250),KCOP(250),KDCOP(250),IUN(10),
     1PD(251),D(250),C(250),DCOP(250)
C
      COMMON K,KP,KD,KCOP,KDCOP,IUN,PD,D,C,DCOP
C
C  SINGLE VARIABLE COMMON
C
      COMMON IP,IFUSE,LIM,LINK,MINC,KL,KMIN,ISTAGE,DMIN,PMIN,PERC
     1,NUMOUT,COEF,ION,MINFUS
C
C  DIMENSIONS AND COMMON AREA FOR ALL (MODE) ROUTINES *****************
C  READ DISK PARAMETERS
      CALL DISKIO(1,A,B,C,D,E)
C  READ CONTROL CARD
C       ISTART = FIRST CASE FOR WHICH K-LINKAGE LISTS ARE REQUIRED
C       IEND =   LAST  CASE FOR WHICH K-LINKAGE LISTS ARE REQUIRED
C       NX = NUMBER OF SIMILARITY COEFFICIENTS TO BE PRINTED FOR EACH CAS
 5    READ 10,ISTART,IEND,NX
 10   FORMAT (3I4)
      DO 50 I=ISTART,IEND
      CALL GET(1,I)
      GO TO (15,20),ITYPE
 15   C(I)=-10.E+90
      GO TO 25
 20   C(I)=+10.E+90
 25   DO 30 J=1,N
 30   K(J)=J
      CALL ORDER (ITYPE,C,K,N)
      PRINT 35,I
 35   FORMAT(/19H LINKAGES FOR CASE ,I4/)
      PRINT 40,(C(J), K(J),J=1,NX)
 40   FORMAT (10(F8.3,I4))
      PRINT 45
 45   FORMAT (/)
 50   CONTINUE
C  RETURN TO READ A NEW CONTROL CARD
      GO TO 5
      END


      SUBROUTINE ORDER (IC,X,KA,N)
C  VERSION ORDER(2)
C  SORTS THE N ELEMENTS OF X INTO -
C  -- DESCENDING ORDER WHEN IC=1
C  -- ASCENDING  ORDER WHEN IC=2
C  AND PERMUTES THE INDEX ARRAY KA ACCORDINGLY
C  -------------------------------------------------------------------
C  THIS IS A MODIFIED VERSION OF THE ALGOL (QUICKERSORT) ALGORITHM
```

```
C  NO. 271, BY R.S. SCOWEN, GIVEN IN - COLLECTED ALGORITHMS FROM THE
C  COMMUNICATIONS OF THE A.C.M. (CACM)
C  ----------------------------------------------------------------------
C  THIS SUBROUTINE SUCCEEDS THE VERSION OF (ORDER) GIVEN IN (A FORTRAN
C  II PROGRAMME FOR NUMERICAL CLASSIFICATION) BY D. WISHART, ST. ANDREWS
C  AUGUST 1968.
C  ----------------------------------------------------------------------
C  TWO VERSIONS OF (ORDER) ARE PROVIDED - THIS VERSION IS CONSIDERABLY
C  FASTER THAN ORDER(1) GIVEN IN STATEMENTS 23100 TO 23380 FOR ALL
C  VALUES OF N, BUT REQUIRES MORE STORAGE.
      DIMENSION X(1),KA(1),LT(20),UT(20)
      J=N
      I=1
      M=1
    5 IF (J-I-1)85,85,10
   10 IP=(I+J)/2
      T=X(IP)
      X(IP)=X(I)
      KT=KA(IP)
      KA(IP)=KA(I)
      IQ=J
      K=I+1
   15 IF (K-IQ)20,20,65
   20 GO TO (30,25),IC
   25 IF (X(K)-T)55,55,35
   30 IF (X(K)-T)35,55,55
   35 GO TO (45,40),IC
   40 IF (X(IQ)-T)50,60,60
   45 IF (X(IQ)-T)60,60,50
   50 P=X(K)
      X(K)=X(IQ)
      X(IQ)=P
      KP=KA(K)
      KA(K)=KA(IQ)
      KA(IQ)=KP
      IQ=IQ-1
   55 K=K+1
      GO TO 15
   60 IQ=IQ-1
      IF (IQ-K)65,35,35
   65 X(I)=X(IQ)
      X(IQ)=T
      KA(I)=KA(IQ)
      KA(IQ)=KT
      IF (2*IQ-I-J)75,75,70
   70 LT(M)=I
      UT(M)=IQ-1
      I=IQ+1
      GO TO 80
   75 LT(M)=IQ+1
      UT(M)=J
      J=IQ-1
   80 M=M+1
      GO TO 5
   85 IF (I-J)90,110,110
   90 GO TO (100,95),IC
   95 IF (X(I)-X(J))110,110,105
  100 IF (X(I)-X(J))105,110,110
  105 P=X(I)
      X(I)=X(J)
```

```
      X(J)=P                                                            38590
      KP=KA(I)                                                          38600
      KA(I)=KA(J)                                                       38610
      KA(J)=KP                                                          38620
  110 IF (M-1)120,120,115                                               38630
  115 M=M-1                                                             38640
      I=LT(M)                                                           38650
      J=UT(M)                                                           38660
      GO TO 5                                                           38670
  120 RETURN                                                            38680
      END                                                               38690


*   SPS VERSION OF SUBROUTINE DISKIO FOR FAST DISK ACCESS ON THE        38700
*   IBM 1620 II. THIS ROUTINE REPLACES THE PUBLISHED FORTRAN IID        38710
*   VERSION AND ROUGHLY REDUCES EXECUTION TIME TO ONE-THIRD             38720
*   TAKE CARE TO ENSURE THAN DISK CONTROL FIELD COM CORRECTLY           38730
*   SPECIFIES THE START OF THE DISK WORK SPACE                          38740
*   AUTHOR - DAVID WISHART, MATHEMATICAL INSTITUTE, UNIVERSITY OF       38750
*   ST. ANDREWS, FIFE, GREAT BRITAIN.  (PROGRAMME WRITTEN - AUG 68)     38760
S       DS   ,*+101                                                     38770
        DC   6,987898,5-S                                               38780
        DAC  6,DISKIO,7-S                                               38790
        DVLC22-S,5,LEN,2,8,2,5,5,SUB-6,5,0,30,0                         38800
        DSC  17,0,0                                                     38810
        DORGS-100                                                       38820
*   CAUTION - BEFORE TESTING THIS ROUTINE READ THE COMMENTS ON          38830
*   CARDS 39290, 39310, AND 39970 TO 40130                              38840
*   DANGER - IF YOU DONT TAKE CARE ALL HELL WILL BE LET LOOSE           38850
ISEL    DSA  0,0,0,0,0,0                                                38860
LSEC    DS   ,ISEL+5                                                    38870
IBIN    DS   ,ISEL+10                                                   38880
LB      DS   ,ISEL+15                                                   38890
XAR     DS   ,ISEL+20                                                   38900
LN      DS   ,ISEL+25                                                   38910
        DS   1,@                                                        38920
        DC   5,0                                                        38930
SUB     TFM  TF+6,ISEL-4                                                38940
        AM   TF+6,4,10                                                  38950
        AM   SUB-1,5,10                                                 38960
        TF   CF+11,SUB-1,11                                             38970
        BNF  TF,CF+11                                                   38980
CF      CF   CF+11                                                      38990
        TF   CF+11,CF+11,11                                             39000
        B7   CF-12                                                      39010
TF      TF   ISEL,CF+11                                                 39020
        AM   TF+6,1,10                                                  39030
        BNR  SUB+12,TF+6,11                                             39040
        AM   SUB-1,1,10                                                 39050
*   BRANCH FOR FETCH/RECORD DISK PARAMETERS                             39060
        CM   ISEL,2,6                                                   39070
        BNH  COMMON                                                     39080
        TFM  CORE,AREA                                                  39090
*   SET UP FETCH/RECORD DIGIT INDICATOR AT IO                           39100
        MM   ISEL,5,610                                                 39110
        TD   IO,99                                                      39120
*   SET UP 10 WORD LENGTH FOR FLOATING F=8                              39130
        TFM  WORD,10,10                                                 39140
        TFM  MODE,100,9                                                 39150
*   SET FLOATING ADDRESS AND ARRAY LENGTH                               39160
```

```
       TF   AD,XAR
*  BRANCH IF NOT FIXED
       CM   ISEL,4,6
       BNH FIX
       TF   LENGTH,LN,11
       B7   SEC
*  SET UP 5 WORD LENGTH FOR FIXED I=5
FIX    TFM WORD,5,10
       TFM MODE,50,9
*SET UP FIXED ADDRESS AND ARRAY LENGTH
       TF   AD,IBIN
       TF   LENGTH,LB,11
*  SET SECTOR ADDRESS IN DISK CONTROL FIELD
SEC    TF   SECT,LSEC,11
*  WHEN MODIFYING THIS ROUTINE CHECK THIS INSTRUCTION
       AM   SECT,218,9
*  COMPUTE STARTING POSITION OF ARRAY
       S    AD,WORD
       AM   AD,1,10
*  SET END OF RECORD WHEN READING A FULL SECTOR
       TFM LAST,AREA
       A    LAST,MODE
*  BRANCH TO RECORD
IN     BD   RECORD,IO
*  THIS SECTION FETCHES FROM DISK - NOW READ
A      GET DEF
*  TRANSMIT RECORD MARK TO END OF FILE
       TD   LAST,REC,6
*  TEST FOR FULL SECTOR FILE
       CM   LENGTH,10
       BNL FULL
*  PARTIAL FILE - REVISE POSITION OF RECORD MARK
       M    LENGTH,WORD
       SF   95
       AM   99,AREA
*  TRANSMIT RECORD MARK TO END OF PARTIAL FILE
       TD   99,REC,6
*      NOW SEND FILE TO ARRAY AREA
FULL   TRNMAD,AREA,6
       B7   OUT
*  THIS SECTION RECORDS TO DISK - SET UP END OF ARRAY ADDRESS
RECORDTF   END,AD
       A    END,MODE
*  SAVE DIGIT IN ARRAY
       TD   SAVE,END,11
*  SEND RECORD MARK TO END OF FILE
       TD   END,REC,6
*  TRANSMIT FILE TO I/O AREA
       TRNMAREA,AD,11
*  REPLACE SAVED DIGIT
       TD   END,SAVE,6
*  NOW WRITE SECTOR
B      PUT DEF
*  END OF DISK OPERATION - INCREMENT SECTOR COUNT
OUT    AM   LSEC,1,610
       SM   LENGTH,10,10
*  BRANCH IF OPERATION COMPLETE
       BNH SUB-1,,6
*  INCREMENT CORE AND SECTOR ADDRESS
       A    AD,MODE
```

```
       AM   SECT,1,10
       B7   IN
COMMONCM    ISEL,2,6
       BL   BRING
*  THIS SECTION RECORDS DISK PARAMETERS TO FIRST 4 SECTORS
C      PUT BCOM
       B7   SUB-1,,6
*  THIS SECTION FETCHES DISK PARAMETERS TO LAST 400 CORE POSITIONS
BRING GET BCOM
       B7   SUB-1,,6
IO     DS   1
MODE   DS   3
WORD   DS   2
AD     DS   5
LENGTHDS    5
END    DS   5
LAST   DS   5
X      DAS 51
AREA   DS   ,X-1
SAVE   DS   1
*  WHEN MODIFYING THIS ROUTINE ENSURE CORRECT DRIVE CODE 3
DISK   DC   2,3
SECT   DS   5
       DC   3,1
CORE   DS   5
REC    DC   1,@
DEF    DD   ,DISK,,N,A
*  THIS DDA DEFINES THE START OF THE WORKING CYLINDERS. ENSURE
*  THAT THE SECTOR ADDRESS (219) SPECIFIES THE CORRECT STARTING
*  SECTOR OF THE NORMAL FORTRAN WORK SPACE, AND THAT THE DRIVE
*  CODE (3) REFERS TO THE APPROPRIATE DRIVE WHERE THE WORK AREA
*  IS LOCATED.
COM    DDA ,3,00219,004,59600
       DC   1,@
*  ALSO CHANGE DISK TO DC 2,D WHERE D IS THE APPROPRIATE DRIVE
*  CODE, AND AM SECT,218,9 WHERE 218 IS THE START OF THE WORK
*  SPACE MINUS 1
BCOM   DD   ,COM,,N,A
LEN    DC   2,@
       DEND
```

```
$$  ----------------------------------------------------------------------
$$  LISTING OF INPUT CARDS FOR THE PENNSYLVANIA AREAL GEOLOGY STUDY
$$  ALL CARDS WHICH COMMENCE WITH $$ OR *LOCAL ARE 1620 CONTROL CARDS OR
$$  COMMENTS - EVERYTHING ELSE IS DATA INPUT
$$  ----------------------------------------------------------------------
```

```
      SUBROUTINE READ (NUM,X,MN,LIST,LENG)
C   VERSION OF SUBROUTINE READ USED FOR INPUT OF THE PENNSYLVANIA
C   GEOLOGIC FORMATION DATA
C   TWO DATA BLOCKS ARE FILED - THE NUMERIC MATRIX CONTAINS THE
C   PROPORTIONS OUT OF 10 OF 48 ROCKS ESTIMATED FOR EACH UNIT.  THE
C   BINARY MATRIX NOTES THE PRESENCE OR ABSENCE OF THE ROCKS IN EACH
```

```
C   SAMPLE
C
      DIMENSION X(1),LIST(1)
C
      DIMENSION P(33)
      IF (NUM-99)15,5,15
    5 READ 10,(P(I),LIST(I),I=1,11)
      READ 10,(P(I),LIST(I),I=12,22)
      READ 10,(P(I),LIST(I),I=23,33)
      NX=33
 10   FORMAT (3X,11(F5.0,I2))
      GO TO 20
   15 READ 10,(P(I),LIST(I),I=1,11)
      READ 10,(P(I),LIST(I),I=12,22)
      NX=22
 20   LENG=0
      DO 21 I=1,MN
 21   X(I)=0.0
      DO 30 I=1,NX
      IF (LIST(I))35,35,25
 25   LENG=LENG+1
      K=LIST(I)
      X(K)=P(I)
 30   CONTINUE
 35   PRINT 40,NUM,(P(I),LIST(I),I=1,LENG)
 40   FORMAT(I4,6X,12(F6.1,I3)/10X,12(F6.1,I3))
C
      RETURN
      END
$$$$
$$JOB
$$XEQSFILE                   1
*LOCALFILE,FILEIN,LOAD,EIGEN
PENNSYLVANIA GEOLOGIC FORMATION DATA
 176  48  48
  1 5.5  8 2.0  7 2.0 13 0.5  6  .      .      .      .      .      .      .
  1  .      .      .      .      .      .      .      .      .      .      .
  2 5.0 13 4.0 14 1.0  1 0.2 10  .      .      .      .      .      .      .
  2  .      .      .      .      .      .      .      .      .      .      .
  3 2.3  1 1.0 16 0.2 14 0.8 19 0.8 20 0.7 21 1.8 22 0.7 23 0.1 24 0.3 25 0.1 27
  3 1.0 17 0.1 15 0.1 47 0.1 18  .      .      .      .      .      .      .
  4 0.3 23 0.1 24 0.2 25 0.5  1 6.5 27 2.0 29  .      .      .      .      .
  4  .      .      .      .      .      .      .      .      .      .      .
  5 7.0 29 1.5 27 1.0 30  .      .      .      .      .      .      .      .
  5  .      .      .      .      .      .      .      .      .      .      .
  6 0.1 18 0.1 30 0.8 29 3.0 27 1.0 25 0.7 24 1.6 23 1.0 22 0.5 21 0.5 20 0.5 19
  6 0.1 17 0.1 01  .      .      .      .      .      .      .      .      .
  7 0.2 25 0.3 24 1.3 23 1.0 22 0.8 21 1.0 19 1.0 20 0.2 18 0.8 17 2.0 16 1.5  1
  7  .      .      .      .      .      .      .      .      .      .      .
  8 0.8 21 1.0 20 0.8 19 0.4 18 0.7 17 1.0 16 0.6  1 1.1 22 1.0 23 0.1 24 0.3 25
  8 2.0 27 0.1 30 0.2 29  .      .      .      .      .      .      .      .
  9 2.3 27 1.0 29 1.5 30 1.3 31 1.3 32 0.2 24 0.2 25 0.1  1 0.3 33 0.1 34 0.1 35
  9 0.1 36 0.2 37 0.1 43 0.1 41 0.1 40 0.4 39 0.1 38  .      .      .      .
 10 0.5 40 0.5 41 6.0 43 0.4 39 0.5 44 0.6 45 0.3 46 0.3 37 0.1 36 0.1 35 0.1 34
 10 0.1 33 0.1 32  .      .      .      .      .      .      .      .      .
 11 0.5 43 0.6 41 0.8 40 2.5 39 3.0 37 0.3 01 0.3 36 0.3 35 0.2 34 0.3 33 1.0 32
 11 0.2 31  .      .      .      .      .      .      .      .      .      .
 12 0.2 32 4.0 31 4.5 30 1.3 29  .      .      .      .      .      .      .
 12  .      .      .      .      .      .      .      .      .      .      .
 13 0.8 30 3.0 29 0.1 28 2.8 27 0.1 26 1.8 25 0.8 24 0.5 23  .      .      .
```

```
13  .       .       .       .       .       .       .       .       .       .       .
14 3.0 23 0.6 25 0.8 24 0.6  1 5.0 22 0.1 21 0.1 20  .       .       .       .
14  .       .       .       .       .       .       .       .       .       .       .
15 0.7 22 0.8 21 1.0 20 6.3 19 0.7 18 0.5 17 0.1 16  .       .       .       .
15  .       .       .       .       .       .       .       .       .       .       .
16 1.0 19 1.0 18 2.3 17 3.0 16 0.1 15 2.5  1  .       .       .       .       .
16  .       .       .       .       .       .       .       .       .       .       .
17 0.1  7 6.0  8 4.0 13 0.2 10  .       .       .       .       .       .       .
17  .       .       .       .       .       .       .       .       .       .       .
18 0.2 10 2.0 13 4.5 14 0.2 47 1.0 16 1.5  1 0.1 17 0.1 18 0.1 19 0.1 20 0.1 15
18  .       .       .       .       .       .       .       .       .       .       .
19 0.1 18 2.5 01 0.1 16 0.1 17 0.3 19 1.0 20 0.7 21 1.5 22 1.0 23 0.5 24 0.8 25
19 1.5 27  .       .       .       .       .       .       .       .       .       .
20 0.1 01 9.5 27 0.5 29  .       .       .       .       .       .       .       .
20  .       .       .       .       .       .       .       .       .       .       .
21 1.5 29 3.0 27 0.8 25 0.8 24 1.0 23 2.0 22 0.1 01 0.2 21 0.2 20 0.1 19 0.1 17
21 0.2 16 0.1 18  .       .       .       .       .       .       .       .       .
22 0.1 25 0.1 24 0.5 23 1.0 22 0.3 21 0.5 20 0.1 18 0.1 17 0.3  1 2.0 16 0.2 47
22 4.5 14  .       .       .       .       .       .       .       .       .       .
23 0.3 47 5.5 14 0.3 15 1.5 16 1.5  1 0.2 21 0.3 22 0.2 23  .       .       .
23  .       .       .       .       .       .       .       .       .       .       .
24 0.1 37 0.9  1 0.1 21 0.7 22 1.0 23 0.4 24 0.4 25 2.0 27 1.0 29 1.3 30 1.0 31
24 0.8 32 0.2 33 0.1 34 0.1 35 0.1 36  .       .       .       .       .       .
25 0.1 29 0.3 30 0.4 31 0.8 32 0.5 33 0.2 34 0.2 35 0.2 36 0.6  1 6.3 37 0.4 45
25  .       .       .       .       .       .       .       .       .       .       .
26 0.5 37 0.2 46 2.0 45 0.7 44 2.7 43 0.1  1 0.5 41 1.0 40 2.0 39  .       .
26  .       .       .       .       .       .       .       .       .       .       .
27 0.7  1 0.9 39 4.0 37 0.2 36 0.2 35 1.3 33 2.5 32 0.2 34  .       .       .
27  .       .       .       .       .       .       .       .       .       .       .
28 1.7 32 3.3 31 2.5 30 2.3 29 0.1 28 0.2 27  .       .       .       .       .
28  .       .       .       .       .       .       .       .       .       .       .
29 0.6 29 0.1 28 2.5 27 0.4 26 1.3 24 5.0 25  .       .       .       .       .
29  .       .       .       .       .       .       .       .       .       .       .
30 4.5 23 0.2 24 0.7  1 2.5 22 0.6 21 0.7 20 0.7 19 0.1 18 0.1 17  .       .
30  .       .       .       .       .       .       .       .       .       .       .
31 2.3 19 1.0 18 1.8 17 0.7  1 4.0 16 0.1 15 0.1 14  .       .       .       .
31  .       .       .       .       .       .       .       .       .       .       .
32 2.0 16 0.6 15 0.4  1 6.5 14 0.1 12 0.2 13  .       .       .       .       .
32  .       .       .       .       .       .       .       .       .       .       .
33 2.5  8 1.0 14 6.5 13 0.2 10  .       .       .       .       .       .       .
33  .       .       .       .       .       .       .       .       .       .       .
34 5.0 14 0.2 15 0.2 47 1.5 16 1.5  1 0.3 17 0.1 18 0.1 19 1.3 20  .       .
34  .       .       .       .       .       .       .       .       .       .       .
35 4.0  1 0.3 20 0.2 21 0.8 22 0.6 23 0.2 24 0.6 25 3.0 27  .       .       .
35  .       .       .       .       .       .       .       .       .       .       .
36 1.5  1 5.0 27 1.0 25 1.0 24 0.5 23 0.5 22 0.1 21 0.1 20 0.1 19  .       .
36  .       .       .       .       .       .       .       .       .       .       .
37 2.8  1 0.1 27 0.1 25 0.2 24 0.3 23 1.5 22 0.5 21 1.0 20 1.0 19 0.3 18 0.3 17
37 1.2 16 0.1 47 0.1 15  .       .       .       .       .       .       .       .
38 1.5 16 3.8  1 4.5 14 0.1 15 0.1 47  .       .       .       .       .       .
38  .       .       .       .       .       .       .       .       .       .       .
39 2.5 14 1.3  1 0.1 47 0.1 15 0.4 16 0.1 17 0.1 18 0.1 19 0.2 20 0.2 21 0.8 22
39 1.0 23 0.2 24 0.2 25 1.3 27 0.6 30 0.1 31 0.6 29  .       .       .       .
40 3.0 37 0.7  1 0.3 36 0.5 35 0.2 34 0.7 33 1.2 32 1.2 31 1.0 30 0.6 29 0.4 27
40 0.1 25 0.1 24  .       .       .       .       .       .       .       .       .
41 5.0 37 1.8 45 0.8 44 2.0 43 0.2 41 0.1 40  .       .       .       .       .
41  .       .       .       .       .       .       .       .       .       .       .
42 4.0 43 0.8 41 1.5 40 2.8 39 0.1 44 0.6 37 0.1  1  .       .       .       .
42  .       .       .       .       .       .       .       .       .       .       .
43 0.1 39 3.0 37 0.2  1 0.7 36 0.1 34 0.7 35 1.8 33 2.5 32 0.6 31 0.3 30  .
```

```
43
44 0.2 32 0.7 31 2.3 30 2.5 29 0.1 28 2.8 27 0.2 26 1.0 25 0.2 28
44
45 0.3 26 2.5 27 0.1 28 2.0 25 1.0 24 0.3  1 1.0 22 2.3 23 0.1 21 0.1 20 0.1 19
45
46 0.6 23 1.0  1 1.0 22 0.3 21 0.6 20 1.2 19 0.4 18 1.0 17 3.0 16 0.3 15 0.6 14
46
47 1.0 16 0.6  1 0.5 15 7.5 14 0.1 12 0.2 13
47
48 2.5 14 0.1  1 0.1 12 0.1  9 0.1 11 0.1 10 7.0 13
48
49 0.2 10 6.5 13 0.9  1 2.5 14
49
50 0.8 22 2.0 14 2.5  1 0.2 15 0.2 47 1.5 16 1.0 19 0.2 17 0.1 18 0.8 20 0.5 21
50
51 1.5 01 0.3 21 0.2 20 0.2 19 0.1 18 0.1 17 3.0 22 1.5 23 0.4 24 1.5 25 0.7 27
51
52 3.5  1 0.2 27 0.2 25 0.1 24 0.6 23 1.8 22 0.5 21 0.5 20 0.3 19 0.2 18 0.2 17
52 1.5 16
53 1.3  1 0.2 21 2.3 22 2.5 23 0.2 20 0.1 19 0.2 24 0.2 25 3.0 27
53
54 1.5 22 0.4 21 0.7 20 0.4 19 0.2 18 0.8 17 3.8 16 0.7  1 0.2 47 0.2 15 1.2 14
54
55 0.1 47 0.2 16 0.1 17 0.1 18 0.1 19 0.2 20 0.2 21 1.0 22 1.0 23 0.2 24 0.2 25
55 1.0 27 1.0 29 1.0 30 1.0 31 0.8 33 0.1 34 0.2 35 0.2 36 0.1 37 0.2  1 1.0 32
56 0.1 31 0.4 32 0.2 33 0.1 34 0.2 35 0.2 36 0.7  1 6.5 37 0.1 38 0.2 44 0.1 45
56 1.0 43 0.1 42
57 0.1 37 0.4 45 2.5 44 6.0 43 0.5 41 0.1 42
57
58 1.5 43 0.1 42 0.5 41 1.0 40 1.0  1 0.1 38 3.0 37 0.1 36 0.1 34 0.1 35 0.1 32
58 2.0 39 0.1 33
59 0.2 36 0.2 35 0.5 37 0.8 33 0.1 34 1.8 32 1.6 31 1.8 30 1.8 29 0.1 28 0.7 27
59 0.1 26 0.2 25
60 0.2  1 0.4 29 0.2 28 3.8 27 0.5 26 0.5 24 3.8 25 0.1 22 0.5 23
60
61 0.2 27 0.1 26 0.7  1 0.4 25 0.5 24 1.0 23 2.3 22 0.5 21 0.7 20 0.8 19 0.2 18
61 0.7 17 1.5 16 0.2 15 0.3 14
62 0.2 15 0.1 18 0.2 17 0.3  1 1.0 16 4.5 14 0.1 12 3.5 13
62
63 0.1 10 0.2 11 0.1  9 0.5  1 0.1 12 0.4 14 8.5 13
63
64 0.1  9 1.0 01 0.2 11 0.3 10 1.5  8 7.0 13
64
65 6.8 13 2.5 14 0.6  1 0.2 10
65
66 2.8 14 0.2 47 0.2 15 1.0  1 2.8 16 0.6 17 0.2 18 1.5 19 0.5 20
66
67 3.0  1 1.0 22 0.6 21 1.2 20 0.8 19 0.3 18 0.8 17 1.8 16
67
68 0.8  1 1.0 16 0.1 17 0.1 18 0.2 20 0.2 21 1.0 22 1.0 23 0.1 24 0.1 25 1.5 27
68 1.5 29 1.5 30 0.5 31 0.1 19
69 2.2 29 0.4 30 5.0 27 0.4 25 0.4 24 0.3  1 0.1 22 0.8 23
69
70 0.1 16 0.4 17 0.2 18 0.3 19 0.3 20 0.4 21 4.0 22 2.0 23 0.5  1 0.2 24 0.2 25
70 0.7 27 0.5 29 0.2 30
71 0.1 38 1.0 39 2.2 37 0.5  1 0.2 36 0.4 35 0.1 34 0.5 33 1.2 32 1.0 30 1.0 31
71 0.5 29 0.6 27 0.1 25 0.1 24 0.1 23
72 0.9 37 0.1 38 0.1  1 3.5 39 3.0 43 0.2 42 0.8 41 0.7 40 0.2 44 0.1 45
72
73 0.6 41 0.9 40 2.5 44 4.3 43 0.2 42 0.7 39 0.2  1 0.1 38 0.1 37
```

73	.		.		.		.		.		.		.		.		.		.		.	
74	0.5	29	0.1	40	0.7	39	0.2	1	0.2	38	3.5	37	0.2	36	0.3	35	0.1	34	0.6	33	1.6	32
74	1.0	31	1.0	30	.		.		.		.		.		.		.		.		.	
75	0.1	32	0.3	31	0.6	30	2.3	29	0.1	28	2.3	27	0.1	26	1.0	24	0.9	25	2.3	23	0.1	22
75	.		.		.		.		.		.		.		.		.		.		.	
76	0.8	25	0.6	1	0.1	26	1.2	24	1.8	23	2.5	22	0.5	21	0.5	20	0.3	19	0.2	18	0.8	17
76	0.8	16	.		.		.		.		.		.		.		.		.		.	
77	0.1	22	0.1	21	0.1	20	0.1	19	0.1	18	0.2	17	0.1	1	1.0	16	3.7	14	3.5	13	0.1	12
77	0.1	11	0.3	15	.		.		.		.		.		.		.		.		.	
78	0.1	14	0.1	12	9.0	13	0.3	11	0.4	10	0.1	9	.		.		.		.		.	
78	.		.		.		.		.		.		.		.		.		.		.	
79	0.2	11	0.2	10	7.5	13	0.6	1	1.3	8	0.1	9	.		.		.		.		.	
79	.		.		.		.		.		.		.		.		.		.		.	
80	0.6	13	7.3	8	0.8	7	0.9	6	0.5	1	.		.		.		.		.		.	
80	.		.		.		.		.		.		.		.		.		.		.	
81	9.2	13	0.2	10	0.1	1	0.5	14	.		.		.		.		.		.		.	
81	.		.		.		.		.		.		.		.		.		.		.	
82	6.0	14	0.2	15	0.2	47	1.3	1	2.0	16	.		.		.		.		.		.	
82	.		.		.		.		.		.		.		.		.		.		.	
83	3.2	1	1.6	16	0.1	17	0.1	18	0.1	19	0.2	20	0.2	21	1.2	22	1.0	23	0.1	24	0.1	25
83	0.8	27	0.4	29	0.5	30	0.2	31	.		.		.		.		.		.		.	
84	3.0	32	3.0	31	0.1	33	1.8	30	0.6	29	0.8	27	0.1	24	0.1	25	0.2	23	.		.	
84	.		.		.		.		.		.		.		.		.		.		.	
85	0.6	31	2.7	30	2.3	29	2.7	27	0.7	25	0.6	24	0.4	23	.		.		.		.	
85	.		.		.		.		.		.		.		.		.		.		.	
86	0.4	1	2.0	23	0.6	24	0.4	25	2.0	27	1.0	29	1.0	30	0.9	31	0.9	32	0.3	33	0.1	34
86	0.1	35	.		.		.		.		.		.		.		.		.		.	
87	0.2	31	0.7	32	0.5	33	0.3	35	0.1	34	0.2	36	0.7	1	1.3	37	5.5	39	0.1	38	.	
87	.		.		.		.		.		.		.		.		.		.		.	
88	2.3	39	4.5	43	0.3	42	0.7	44	0.8	45	0.6	41	0.3	40	.		.		.		.	
88	.		.		.		.		.		.		.		.		.		.		.	
89	0.1	45	0.2	44	1.7	43	0.1	42	0.4	41	0.9	40	0.1	38	0.3	1	2.0	39	2.7	37	0.2	36
89	0.2	35	0.1	34	0.3	33	0.1	31	0.3	32	.		.		.		.		.		.	
90	0.2	37	0.1	36	0.1	35	0.1	34	0.2	33	1.0	32	1.2	31	1.8	30	2.5	29	0.1	28	1.6	27
90	0.1	26	0.2	25	0.2	24	0.1	23	.		.		.		.		.		.		.	
91	0.5	27	0.1	26	0.3	25	0.5	24	2.5	23	2.0	22	0.6	21	0.6	20	0.5	19	0.2	18	0.8	17
91	0.2	1	0.6	16	0.1	15	0.1	14	.		.		.		.		.		.		.	
92	0.1	21	0.1	20	0.1	19	0.1	18	0.5	17	2.5	16	0.2	15	2.7	14	1.0	1	2.5	13	0.1	12
92	0.1	11	.		.		.		.		.		.		.		.		.		.	
93	0.1	14	0.1	12	0.1	11	0.2	10	0.5	1	1.2	08	7.7	13	.		.		.		.	
93	.		.		.		.		.		.		.		.		.		.		.	
94	0.5	1	3.0	13	6.5	8	.		.		.		.		.		.		.		.	
94	.		.		.		.		.		.		.		.		.		.		.	
95	0.1	13	0.8	1	1.1	7	1.0	6	7.0	8	.		.		.		.		.		.	
95	.		.		.		.		.		.		.		.		.		.		.	
96	0.2	1	0.4	8	0.5	7	7.8	6	0.1	5	1.0	4	.		.		.		.		.	
96	.		.		.		.		.		.		.		.		.		.		.	
97	0.3	1	0.3	10	9.1	13	0.3	14	.		.		.		.		.		.		.	
97	.		.		.		.		.		.		.		.		.		.		.	
98	0.1	13	0.7	15	0.1	47	1.5	1	6.5	14	0.2	16	0.1	18	0.1	17	0.1	19	0.1	20	.	
98	.		.		.		.		.		.		.		.		.		.		.	
99	0.8	1	0.5	16	0.1	17	0.1	18	0.1	19	0.3	20	0.3	21	0.8	22	0.6	23	0.1	24	0.1	25
99	0.8	27	0.6	29	0.9	30	0.8	31	1.0	32	0.2	33	0.1	34	0.2	35	0.1	36	1.0	37	0.2	38
99	0.1	39	.		.		.		.		.		.		.		.		.		.	
100	3.5	37	0.5	1	0.2	38	0.7	35	0.1	34	3.0	32	0.6	36	0.8	33	0.3	31	.		.	
100	.		.		.		.		.		.		.		.		.		.		.	
101	0.5	32	2.3	31	2.2	30	3.0	29	1.8	27	.		.		.		.		.		.	
101	.		.		.		.		.		.		.		.		.		.		.	
102	1.0	27	1.8	29	0.9	30	0.9	31	2.5	32	0.6	33	0.1	34	0.4	35	0.3	36	0.1	1	0.8	37
102	0.1	38	0.1	39	.		.		.		.		.		.		.		.		.	

103	0.1	36	0.2	1	0.8	37	0.2	38	4.5	39	0.7	40	0.6	41	2.5	43	0.1	42	.		.	
103	.		.		.		.		.		.		.		.		.		.		.	
104	0.1	36	0.1	34	0.2	33	5.0	43	0.2	42	0.5	41	0.5	40	1.8	39	0.1	38	0.7	37	0.5	1
104	0.2	35	.		.		.		.		.		.		.		.		.		.	
105	0.1	40	0.3	39	0.1	38	0.7	37	0.3	1	0.2	36	0.2	35	0.1	34	0.7	33	2.5	32	1.8	31
105	1.8	30	0.6	29	0.1	28	0.1	27	.		.		.		.		.		.		.	
106	0.1	31	0.5	30	1.1	29	0.1	28	2.8	27	0.1	26	1.1	25	0.6	24	1.5	23	0.8	22	0.3	21
106	0.3	20	0.1	19	0.1	18	0.2	17	0.1	16	.		.		.		.		.		.	
107	0.1	22	0.1	21	0.1	20	0.1	19	0.1	18	0.5	17	0.1	1	2.3	16	0.5	15	2.9	14	0.1	12
107	0.1	11	3.0	13	.		.		.		.		.		.		.		.		.	
108	0.1	14	0.1	12	0.2	11	0.2	10	1.0	1	0.9	8	7.5	13	.		.		.		.	
108	.		.		.		.		.		.		.		.		.		.		.	
109	1.0	1	0.1	10	0.2	13	6.0	8	0.9	7	1.8	6	.		.		.		.		.	
109	.		.		.		.		.		.		.		.		.		.		.	
110	0.1	1	1.0	7	5.3	6	3.5	8	.		.		.		.		.		.		.	
110	.		.		.		.		.		.		.		.		.		.		.	
111	0.4	7	0.1	8	9.5	6	.		.		.		.		.		.		.		.	
111	.		.		.		.		.		.		.		.		.		.		.	
112	0.2	5	4.0	6	5.0	4	0.8	1	.		.		.		.		.		.		.	
112	.		.		.		.		.		.		.		.		.		.		.	
113	0.2	10	0.3	1	8.0	13	1.5	14	.		.		.		.		.		.		.	
113	.		.		.		.		.		.		.		.		.		.		.	
114	5.0	14	0.2	15	0.1	47	0.3	16	1.3	1	0.1	17	0.1	18	0.1	19	0.2	20	0.2	21	0.8	22
114	0.7	23	0.1	24	0.1	25	0.3	27	0.2	29	0.1	30	.		.		.		.		.	
115	0.2	22	0.3	23	0.1	24	0.1	25	0.4	27	0.5	29	0.7	30	0.7	31	0.8	32	0.2	33	0.1	34
115	0.2	35	0.2	36	1.6	37	0.2	38	3.0	39	0.5	40	0.2	41	.		.		.		.	
116	0.1	32	1.9	39	0.6	38	4.8	37	0.5	36	0.7	1	0.5	35	0.1	34	0.4	33	.		.	
116	.		.		.		.		.		.		.		.		.		.		.	
117	2.0	31	0.1	30	3.8	32	0.1	34	0.7	35	1.0	1	1.5	33	0.3	36	0.1	37	.		.	
117	.		.		.		.		.		.		.		.		.		.		.	
118	0.5	31	3.8	32	0.7	33	0.1	34	0.5	35	0.4	36	2.0	37	0.2	38	1.3	39	0.2	41	0.2	40
118	0.2	43	.		.		.		.		.		.		.		.		.		.	
119	2.0	39	5.5	43	1.2	41	0.1	42	1.2	40	0.1	38	0.1	37	.		.		.		.	
119	.		.		.		.		.		.		.		.		.		.		.	
120	0.3	43	0.2	41	0.3	40	1.5	39	0.1	38	2.3	37	0.5	1	0.1	36	0.1	34	0.3	35	0.8	33
120	2.3	32	0.6	31	0.2	30	.		.		.		.		.		.		.		.	
121	0.3	32	0.8	31	2.0	30	2.7	29	0.1	28	2.7	27	0.1	26	0.5	25	0.2	24	0.2	23	0.1	22
121	.		.		.		.		.		.		.		.		.		.		.	
122	0.2	27	0.1	26	0.5	25	0.3	24	1.1	23	1.0	22	0.3	21	0.3	20	0.2	19	0.2	18	0.3	17
122	0.4	1	2.0	16	0.3	15	1.6	14	0.6	13	0.1	12	.		.		.		.		.	
123	0.1	15	0.4	14	0.5	1	0.1	12	0.2	11	0.2	10	8.0	13	0.6	8	.		.		.	
123	.		.		.		.		.		.		.		.		.		.		.	
124	0.1	10	0.5	13	4.5	8	0.5	1	1.3	7	3.0	6	.		.		.		.		.	
124	.		.		.		.		.		.		.		.		.		.		.	
125	0.4	7	0.3	1	0.1	8	9.2	6	.		.		.		.		.		.		.	
125	.		.		.		.		.		.		.		.		.		.		.	
126	10.	6	.		.		.		.		.		.		.		.		.		.	
126	.		.		.		.		.		.		.		.		.		.		.	
127	8.6	6	0.2	5	0.1	1	1.1	4	.		.		.		.		.		.		.	
127	.		.		.		.		.		.		.		.		.		.		.	
128	8.0	4	0.1	6	2.0	1	.		.		.		.		.		.		.		.	
128	.		.		.		.		.		.		.		.		.		.		.	
129	0.1	10	1.5	13	7.5	14	0.5	1	0.1	15	0.1	47	0.1	16	.		.		.		.	
129	.		.		.		.		.		.		.		.		.		.		.	
130	0.7	30	1.5	14	0.6	16	1.5	1	0.1	47	0.1	15	0.2	17	0.2	18	0.1	19	0.2	20	1.0	22
130	1.0	23	0.1	25	0.1	24	0.7	27	0.7	29	0.7	31	0.5	32	0.2	21	.		.		.	
131	0.1	30	0.4	31	1.5	32	0.2	33	0.1	34	0.2	35	0.2	36	0.8	37	1.5	39	1.0	40	0.9	41
131	0.1	42	2.5	43	.		.		.		.		.		.		.		.		.	
132	4.5	39	5.0	37	0.1	40	0.3	38	.		.		.		.		.		.		.	
132	.		.		.		.		.		.		.		.		.		.		.	

```
133 5.5 37 1.3 36 1.3 35 0.1 34 0.8  1 0.7 33 0.3 32  .    .    .    .
133  .    .    .    .    .    .    .    .    .    .    .
134 0.8 32 0.1  1 0.4 33 0.1 34 0.3 35 0.2 36 2.3 37 3.0 39 0.2 38 0.8 40 0.5 43
134 0.8 41  .    .    .    .    .    .    .    .    .    .
135 0.1 43 0.3 41 0.6 40 0.2 38 3.5 39 3.8 37 0.3  1 0.8 48  .    .    .
135  .    .    .    .    .    .    .    .    .    .    .
136 0.7 37 0.5  1 0.2 36 0.3 35 0.1 34 0.5 33 2.5 32 0.9 31 2.5 30 1.0 29 0.3 27
136  .    .    .    .    .    .    .    .    .    .    .
137 0.8 29 0.1 30 2.5 27 0.1 26 1.3 25 0.3 24 1.5 23 1.3 22 0.3 21 0.3 20 0.2 19
137 0.1 18 0.4 17 0.3  1 0.3 16 0.1 28  .    .    .    .    .    .
138 0.1 21 0.1 20 0.1 19 0.1 18 0.3 17 1.3 16 0.2 15 1.8 14 0.2 12 5.0 13 0.1 10
138 0.2  1  .    .    .    .    .    .    .    .    .    .
139 0.1 11 0.1 10 2.0 13 2.3  1 4.0  8 0.7  7 0.4  6  .    .    .    .
139  .    .    .    .    .    .    .    .    .    .    .
140 0.1  8 0.4  7 3.0  6 0.2  5 0.2  1 5.5  4 0.1  3  .    .    .    .
140  .    .    .    .    .    .    .    .    .    .    .
141 0.1  5 7.5  6 0.4  1 2.0  4  .    .    .    .    .    .    .
141  .    .    .    .    .    .    .    .    .    .    .
142 10.  6  .    .    .    .    .    .    .    .    .    .
142  .    .    .    .    .    .    .    .    .    .    .
143 0.2  5 0.5  1 4.3  4 5.0  6  .    .    .    .    .    .    .
143  .    .    .    .    .    .    .    .    .    .    .
144 9.0  4 0.7  6 0.2  7  .1  8  .    .    .    .    .    .    .
144  .    .    .    .    .    .    .    .    .    .    .
145 4.3 14 3.0 16 0.2 47 0.2 15 1.5  1 0.1 17 0.1 18 0.1 19 0.3 20 0.2 21  .
145  .    .    .    .    .    .    .    .    .    .    .
146 0.1 17 0.1 18 1.3 32 1.1 31 0.5 29 0.6  1 0.2 16 0.2 20 0.2 21 1.6 22 1.2 23
146 0.1 24 0.1 25 0.6 27 0.8 30 0.4 37 0.3 39 0.1 40 0.1 41 0.1 43  .    .
147 0.1 36 0.2 37 0.6 39 0.3 40 1.3 41 6.5 43 0.3 42 0.5 44 0.1 45  .    .
147  .    .    .    .    .    .    .    .    .    .    .
148 1.2 41 1.2 40 6.5 39 0.5 38 0.5 37  .    .    .    .    .    .
148  .    .    .    .    .    .    .    .    .    .    .
149 0.2 39 0.4 38 8.5 37 0.5 36 0.2 35 0.1 34  .    .    .    .    .
149  .    .    .    .    .    .    .    .    .    .    .
150 0.2 36 0.2 38 2.0 37 0.1 35 7.5 39  .    .    .    .    .    .
150  .    .    .    .    .    .    .    .    .    .    .
151 0.2 39 0.1 38 5.0 37 0.5 36 0.3 35 1.3 48 0.3 46 0.2 45 0.5  1 0.1 34 0.5 33
151 0.6 32  .    .    .    .    .    .    .    .    .    .
152 0.3 22 0.5 23 0.1 20 0.1 35 0.1 21 0.1 34 0.2 33 2.2 32 0.8 31 1.0 30 1.3 29
152 0.1 28 2.5 27 0.1 26 0.4 25 0.2 24  .    .    .    .    .    .
153 0.1 27 0.1 26 0.2 25 0.2 24 0.6 23 0.8 22 0.3 21 0.3 20 0.2 19 0.2 18 0.4  1
153 0.7 17 1.8 16 0.2 15 2.3 14 1.2 13 0.1 12  .    .    .    .    .
154 0.2 14 0.1 12 6.5 13 0.2 11 0.2 10 0.4  1 2.5  8  .    .    .    .
154  .    .    .    .    .    .    .    .    .    .    .
155 2.3  8 0.3  1 1.2  7 3.5  6 0.2  5 2.5  4  .    .    .    .    .
155  .    .    .    .    .    .    .    .    .    .    .
156 0.3  3 0.2  1 0.1  5 9.0  4 0.1  6  .    .    .    .    .    .
156  .    .    .    .    .    .    .    .    .    .    .
157 4.0  4 0.2  5 0.3  1 5.5  6  .    .    .    .    .    .    .
157  .    .    .    .    .    .    .    .    .    .    .
158 9.9  6 0.1  1  .    .    .    .    .    .    .    .    .
158  .    .    .    .    .    .    .    .    .    .    .
159 1.8  6 1.0  1 0.1  3 0.2  5 7.0  4  .    .    .    .    .    .
159  .    .    .    .    .    .    .    .    .    .    .
160 5.8  4 0.8 03 0.2  1 0.9  6 0.9  7 1.5  8  .    .    .    .    .
160  .    .    .    .    .    .    .    .    .    .    .
161 1.2  1 5.5 22 0.5 21 1.2 20 0.4 19 0.1 18 0.4 17 0.5 16  .    .    .
161  .    .    .    .    .    .    .    .    .    .    .
162 0.5 45 0.5 22 1.5 23 0.1 24 0.1 25 1.0 27 0.7 29 2.0 30 0.7 31 2.0 32 0.1 41
162 0.3 43 0.1 42 0.1 44  .    .    .    .    .    .    .    .
```

```
163 0.5 42 4.5 43 2.0 44 3.0 45  .     .     .     .     .     .     .
163  .     .     .     .     .     .     .     .     .     .     .
164 2.5 43 3.0 41 1.5 40 3.0 39  .     .     .     .     .     .     .
164  .     .     .     .     .     .     .     .     .     .     .
165 6.0 39 1.3 38 3.0 37  .     .     .     .     .     .     .     .
165  .     .     .     .     .     .     .     .     .     .     .
166 0.4 43 2.0 39 0.1 38 4.5 37 0.5 36 0.5 35 0.1 34 0.6 33 1.0 32 0.1 45  .
166  .     .     .     .     .     .     .     .     .     .     .
167 0.1 46 0.5 45 1.0 43 0.1 37 0.1 41 0.5  1 0.1 36 0.2 40 0.4 39 0.5 33 0.2 35
167 3.3 32 1.2 31 0.8 30 0.5 29 0.1 28 0.1 27 0.1 34  .     .     .     .
168 0.1 31 0.3 30 0.6 29 0.1 28 2.2 27 0.1 26 0.4 25 0.2 24 1.2 23 1.2 22 0.5 21
168 0.3 20 0.1 19 0.1 18 0.4 17 1.5 16 0.1 15 0.3 14  .     .     .     .
169 0.1 17 0.5 16 0.1 15 1.8 14 7.0 13 0.1 10 0.1 12 0.1 11  .     .     .
169  .     .     .     .     .     .     .     .     .     .     .
170 0.1 10 1.0 13 4.3  8 0.6  1 1.2  7 2.0  6 0.1  5 0.2  4  .     .     .
170  .     .     .     .     .     .     .     .     .     .     .
171 0.2  7 1.5  6 0.2  1 0.2  5 6.0  4 1.8  3  .     .     .     .     .
171  .     .     .     .     .     .     .     .     .     .     .
172 7.7  3 2.3  4  .     .     .     .     .     .     .     .     .
172  .     .     .     .     .     .     .     .     .     .     .
173 0.6  3 8.9  4 0.4  6 0.1  5  .     .     .     .     .     .     .
173  .     .     .     .     .     .     .     .     .     .     .
174 5.8  4 0.2  5  .5  1 3.5  6  .     .     .     .     .     .     .
174  .     .     .     .     .     .     .     .     .     .     .
175 0.2  1 6.5  4 2.0  3 1.2  2  .     .     .     .     .     .     .
175  .     .     .     .     .     .     .     .     .     .     .
176 0.1  3 0.1  2 3.0  4 1.2  1 1.0  6 1.0  7 2.3  8 0.1 10 0.1 11 1.2 13  .
176  .     .     .     .     .     .     .     .     .     .     .
$$$$

$$JOB
$$XEQSCORREL                   1
*LOCALCORREL,COREAD,ANALYS,COEF
 1 7   48
$$$$

$$JOB
$$XEQSMODE                     1
*LOCALMODE,ORDER,IOFILE,OUTPUT,INTRO
 3 1 2 0.8 N
$$$$

$$JOB
$$XEQSHIERAR                   1
*LOCALHIERAR,CLASS,DENDRO
2  2 10
$$$$

$$  FINAL GROUPINGS, PUNCHED BY MODE AND HIERAR, CORRESPONDING TO
$$  FIGURES 2 - 5 OF THE KANSAS CONTRIBUTION
$$JOB
$$XEQSRESULT                   1
*LOCALRESULT,IOINP,IONUM,IOBIN,IOCLUS
  X        XX                XXX XX
 MODE COMPLETE GROUP  5 KL= 3  74 DENSE 12 CLUSTERS ENC RATIO  .64 COEF     .087
  5 12  8 10  7 10  8 10  7  9  4  7  7  8  8  8  3 12  8 10 10  6  6  7 11  9
  7  7 10  8  8  6  3  6  8 10  8  6  6  7 11  9  7  7 10  8  6  3  3  8  8  8
 10  8  7 11  9  4  7 10  8 12  3  3  3  6  8  7 10  8  7  9  9  7 10  8 12  3
  3  5  3  6  8  7  7  7  4  9  4  7  8 12  3  5  5  2  3  6  7  7  7  7  9  9
  7 10 12  3  5  2  2  1  3  6  4  4  7  7  9  7  7  8  3  5  2  2  2  1  6  7
```

```
  4  4 11  4  4  7 10  3  5  1  2  2  1  1  6  7  9  4 11  4 11  7  6  3  1  1
  1  2  1  1  8  7  9  9  4  4  7 10  3  5  1  7  1  1  1  1
 FURTHEST NEIGHBOUR GROUP 166 FUSE POINTS   3   5 AT COEF   2.022  10 CLUSTERS
  5  6  7 10  7  7  7  7  7  7  4  7  7  7  7  7  5  6  7 10  7  6  6  7  4  7
  4  7  7  7  7  6  3  6  7 10  7  6  7  4  4  7  4  7  7  7  6  3  3  7  7  7
  7  7  7  4  7  4  7  7  7  6  3  3  3  7  7  7 10  7  4  7  7  4  7  7  6  3
  3  5  3  6  7  7  7  7  9  7  4  7  7  6  3  5  5  2  3  6  7  4  7  7  7  7
  7  7  6  3  5  2  2  1  3  6  4  4  7  4  7  4  7  7  3  5  2  2  2  1  6  7
  7  4  4  4  4  7  7  3  5  1  2  2  1  1  6  7  7  9  4  9  4  7  7  3  1  1
  1  2  1  1  7  7  7  7  9  4  7  7  3  5  1  8  1  1  1  1
 WARDS METHOD GROUP 166 FUSE POINTS   4  12 AT COEF   4.512  10 CLUSTERS
  5  6  8  7  7  7  8  7  7  9  4  7  7  8  8  8  5  6  8  7  7  6  6  7  4  9
  4  7  7  8  8  6  3  6  8  7  8  6  8  4  4  9  4  7  7  8  6  3  3  8  8  8
  7  8  7  4  9  4  7  7  8  6  3  3  3  8  8  7  7  8  4  9  9  4  7  8  6  3
  3  5  3  6  8  7  7  7 10  9  4  7  8  6  3  5  5  2  3  6  7  4  7  7  9  9
  7  7  6  3  5  5  2  1  3  6  4  4  7  4  9  4  7  8  3  5  2  2  2  1  6  8
  9  4  4  4  4  7  7  6  5  1  2  2  1  1  6  7  9 10  4 10  4  7  8  3  5  1
  1  2  1  1  8  7  9  9 10  4  7  7  3  5  1  1  1  1  1  5
 FLEXIBLE FORM GROUP 166 FUSE POINTS   4   5 AT COEF   3.123  10 CLUSTERS
  5  6  8  7  7  7  8  7  7  9  4  7  7  8  8  8  5  6  8  7  7  6  6  7  4  9
  4  7  7  8  8  6  3  6  7  7  8  6  8  4  4  9  4  7  7  8  6  3  3  8  8  8
  7  8  7  4  9  4  7  7  8  6  3  3  3  8  8  7  7  8  4  9  9  4  7  8  6  3
  3  5  3  6  8  7  7  7  4  9  4  7  8  6  3  5  5  2  3  6  7  4  7  7  9  9
  7  7  6  3  5  5  2  1  3  6  4  4  7  4  9  4  7  8  3  5  2  2  2  1  6  8
  9  4  4  4  4  7  7  3  5  1  2  2  1  1  6  7  9  4  4  4  4  7  8  3  5  1
  1  2  1  1  8  7  9  9  4  4  7  7  3  5  1 10  1  1  1  5
$$$$
$$  ---------------------------------------------------------------------------
```

```
$$  ---------------------------------------------------------------------------
$$  LISTING OF INPUT DATA CARDS FOR THE TRIAL DATA SCHEDULE
$$  ---------------------------------------------------------------------------


$$   RECOMPILE NEW VERSION OF SUBROUTINE READ  ********************************
      SUBROUTINE READ (NUM,X,MN,LIST,LENG)
C   USER READ SUBROUTINE - REWRITTEN FOR EACH DATA DECK
C   NUM - SAMPLE NUMBER SET BY THE MAIN PROGRAMME AND AVAILABLE IN THE
C         SUBROUTINE FOR CHECKING PURPOSES
C   X(I),I=1,MN - THE ARRAY OF MN NUMERIC VARIABLE VALUES
C   LIST(I),I=1,LENG -THE LIST OF BINARY ATTRIBUTES POSSESSED BY THE CASE
C
      DIMENSION X(1),LIST(1)
C
C   READ THE CASE VALUES
      READ 5,(X(I),I=1,MN),(LIST(I),I=1,10)
    5 FORMAT (6F5.2,10I3)
C
C   COMPUTE THE BINARY SAMPLE LIST LENGTH
C
      DO 10 I=1,10
      IF (LIST(I)) 15,15,10
   10 CONTINUE
   15 LENG=I-1
```

```
C  PRINT DATA VALUES
      PRINT 5,(X(I),I=1,MN),(LIST(I),I=1,LENG)
C
C  THE ONUS LIES WITH THE USER TO
C           (A)  WRITE SUITABLE READ AND FORMAT STATEMENTS
C           (B)  IN THE CASE OF BINARY INPUT - CALCULATE THE LIST LENGTH
C           (C)  PRINT THE INPUT DATA FOR CHECKING PURPOSES
C
      RETURN
      END

$$      START OF SEQUENCE 1 ***************************************************

$$JOB
$$XEQSFILE                    1
*LOCALFILE,FILEIN,LOAD,EIGEN
TRIAL DATA - TESTS MIXED-MODE OPERATIONS AND PROVIDES PROGRAM DIAGNOSTICS
   9   6   9SC 6E
 2.56  .57  .53  .69  .43  .46  2  3  4  5  6
 2.70  .72  .54  .72  .41  .25  1  3  4  5  6
 2.10  .50  .52  .68  .46  .30  1  2  3  5  6
 2.11  .47  .71  .84  .56  .53  1  2  3  4  6
 1.74  .38  .49  .53  .42  .42  5  7  8  9
 1.38  .25  .38  .41  .33  .22  5  6  8  9
 2.04  .45  .68  .80  .45  .40  5  6  7  9
 1.92  .57  .70  .78  .55 1.24  5  6  7  8
 2.37  .87  .82  .87  .51  .63  6  7  8  9
$$$$
$$JOB
$$XEQSCORREL                  1
*LOCALCORREL,COREAD,ANALYS,COEF
13 5       6
$$$$
$$JOB
$$XEQSMODE                    1
*LOCALMODE,ORDER,IOFILE,OUTPUT,INTRO
     1      N                              MODE PARAMETER CARD
$$$$
$$JOB
$$XEQSRESULT                  1
*LOCALRESULT,IOINP,IONUM,IOBIN,IOCLUS
XXXXXX 6  XXX       X 5       XXXXXX   RESULT PARAMETER CARD
   MODE NUCLEI GROUP  1 KL= 3   9 DENSE  2 CLUSTERS ENC RATIO 1.00 COEF     .333
  1  1  1  1  5  5  5  5  5
$$$$

$$      START OF SEQUENCE 2 ***************************************************

$$JOB
$$XEQSCORREL                  1
*LOCALCORREL,COREAD,ANALYS,COEF
 1 5 4
$$$$
$$JOB
$$XEQSMODE                    1
*LOCALMODE,ORDER,IOFILE,OUTPUT,INTRO
 3
$$$$
$$JOB
$$XEQSHIERAR                  1
```

```
*LOCALHIERAR,IOFILE,CLASS,DENDRO
7  2  4 -.25
$$$$
$$JOB
$$XEQSRESULT                    1
*LOCALRESULT,IOINP,IONUM,IOBIN,IOCLUS
                                XXX XX
 FLEXIBLE FORM GROUP   6 FUSE POINTS   1   3 AT COEF   2.606   3 CLUSTERS
  1  1  1  1  5  5  1  8  8
$$$$

$$      START OF SEQUENCE 3 ***********************************************

$$JOB
$$XEQSCORREL                    1
*LOCALCORREL,COREAD,ANALYS,COEF
 3 5   9  2
   4   6
$$$$
$$JOB
$$XEQSRESULT                    1
*LOCALRESULT,IOINP,IONUM,IOBIN,IOCLUS
                       X 4
$$$$
$$  ----------------------------------------------------------------------
```

TRIAL DATA SCHEDULE - START OF SEQUENCE 1

EXECUTE FILE

```
TRIAL DATA - TESTS MIXED-MODE OPERATIONS AND PROVIDES PROGRAM DIAGNOSTICS

NUMBER OF CASES =     9
NUMBER OF BINARY VARIABLES =     9
NUMBER OF NUMERIC VARIABLES =     6

FILE DATA MATRIX


READ AND FILE MIXED-MODE DATA
REPLACE NUMERIC DATA BY STANDARD SCORES
PRINCIPAL COMPONENTS SOLUTION SELECTED FOR NUMERIC DATA
CORRELATIONS TO BE FILED
2.56   .57   .53   .69   .43   .46   2   3   4   5   6
2.70   .72   .54   .72   .41   .25   1   3   4   5   6
2.10   .50   .52   .68   .46   .30   1   2   3   5   6
2.11   .47   .71   .84   .56   .53   1   2   3   4   6
1.74   .38   .49   .53   .42   .42   5   7   8   9
1.38   .25   .38   .41   .33   .22   5   6   8   9
2.04   .45   .68   .80   .45   .40   5   6   7   9
1.92   .57   .70   .78   .55  1.24   5   6   7   8
2.37   .87   .82   .87   .51   .63   6   7   8   9
RAW DATA FILED
BINARY SAMPLE LIST LENGTHS FILED
BINARY VARIABLE FREQUENCIES FILED
NUMERIC MEANS FILED
NUMERIC VARIANCES FILED
NUMERIC CORRELATIONS FILED
FILE EIGENVALUES
ALL EIGENVECTORS FILED
STANDARD SCORES FILED
  6 COMPONENT SCORES FILED
FILE COMPLETE
JOB ENDS
```

EXECUTE CORREL (SEQUENCE 1)

```
TRIAL DATA - TESTS MIXED-MODE OPERATIONS AND PROVIDES PROGRAM DIAGNOSTICS

NUMBER OF CASES =     9
NUMBER OF BINARY VARIABLES =     9
NUMBER OF NUMERIC VARIABLES =     6

CALCULATION OF COEFFICIENT MATRIX AND K-LINKAGE LISTS


MASK ALL NUMERIC DATA


USE BINARY DATA
NO BINARY MASK APPLIED
SELECT COEFFICIENT NUMBER 13
COEFFICIENT IS SIMILARITY TYPE
FILE    5 K-LINKAGE LISTS
DISK FILE USES     65 RECORDS

COEFFICIENTS AND K-LINKAGE LISTS CALCULATED AND FILED
JOB ENDS
```

EXECUTE MODE (SEQUENCE 1)

```
TRIAL DATA - TESTS MIXED-MODE OPERATIONS AND PROVIDES PROGRAM DIAGNOSTICS


NUMBER OF CASES =     9
NUMBER OF BINARY VARIABLES =     9
NUMBER OF NUMERIC VARIABLES =     6


STANDARD SCORES FILED
BINARY DATA FILED
  6 COMPONENT SCORES FILED
BINARY COEFFICIENT CALCULATED
NO MASK USED
COEFFICIENT NUMBER 13 CALCULATED
  5 K-LINKAGE LISTS FILED




MODE ANALYSIS

DENSITY LEVEL =  3
MINIMUM NUMBER OF CLUSTERS CONTROL =  1
MINIMUM ENCLOSURE RATIO =   .80
FUSION OUTPUT OF CLUSTERS SIZE - MUST EXCEED  1 DENSE POINTS


NOW INTRODUCE DENSE POINT   3 AT COEFFICIENT      .444 AND FORM NEW CLUSTER NUCLEUS   3
NOW INTRODUCE DENSE POINT   2 AT COEFFICIENT      .444 AND FORM NEW CLUSTER NUCLEUS   2
NOW FUSE CLUSTERS    2    3 AT COEFFICIENT      .444. NEW CLUSTER CODE IS   2
NOW INTRODUCE DENSE POINT   4 AT COEFFICIENT      .444 AND FORM NEW CLUSTER NUCLEUS   4
NOW FUSE CLUSTERS    2    4 AT COEFFICIENT      .444. NEW CLUSTER CODE IS   2
NOW INTRODUCE DENSE POINT   1 AT COEFFICIENT      .444 AND FORM NEW CLUSTER NUCLEUS   1
NOW FUSE CLUSTERS    1    2 AT COEFFICIENT      .444. NEW CLUSTER CODE IS   1
NOW INTRODUCE DENSE POINT   7 AT COEFFICIENT      .333 AND FORM NEW CLUSTER NUCLEUS   7
NOW INTRODUCE DENSE POINT   8 AT COEFFICIENT      .333 AND FORM NEW CLUSTER NUCLEUS   8
NOW FUSE CLUSTERS    7    8 AT COEFFICIENT      .333. NEW CLUSTER CODE IS   7
NOW INTRODUCE DENSE POINT   9 AT COEFFICIENT      .333 AND FORM NEW CLUSTER NUCLEUS   9
NOW FUSE CLUSTERS    7    9 AT COEFFICIENT      .333. NEW CLUSTER CODE IS   7
NOW INTRODUCE DENSE POINT   6 AT COEFFICIENT      .333 AND FORM NEW CLUSTER NUCLEUS   6
NOW FUSE CLUSTERS    6    7 AT COEFFICIENT      .333. NEW CLUSTER CODE IS   6
NOW INTRODUCE DENSE POINT   5 AT COEFFICIENT      .333 AND FORM NEW CLUSTER NUCLEUS   5
NOW FUSE CLUSTERS    5    6 AT COEFFICIENT      .333. NEW CLUSTER CODE IS   5

CLASSIFICATIONS PRIOR TO FUSION
  MODE NUCLEI GROUP  1 KL= 3   9 DENSE  2 CLUSTERS ENC RATIO 1.00 COEF     .333
   1    1    1    1    5    5    5    5    5
 MODE COMPLETE GROUP  1 KL= 3   9 DENSE  2 CLUSTERS ENC RATIO 1.00 COEF     .333
   1    1    1    1    5    5    5    5    5
NOW FUSE CLUSTERS    1    5 AT COEFFICIENT      .222. NEW CLUSTER CODE IS   1
NUMBER OF CLUSTERS =   1
ENCLOSURE RATIO =  1.00
END CONDITIONS SATISFIED
JOB ENDS
```

EXECUTE RESULT (SEQUENCE 1)

TRIAL DATA - TESTS MIXED-MODE OPERATIONS AND PROVIDES PROGRAM DIAGNOSTICS

NUMBER OF CASES = 9
NUMBER OF BINARY VARIABLES = 9
NUMBER OF NUMERIC VARIABLES = 6

STANDARD SCORES FILED
BINARY DATA FILED
6 COMPONENT SCORES FILED
BINARY COEFFICIENT CALCULATED
NO MASK USED
COEFFICIENT NUMBER 13 CALCULATED
5 K-LINKAGE LISTS FILED

NUMERIC MEANS AND STANDARD DEVIATIONS

1	2.1022	.3842
2	.5311	.1721
3	.5966	.1300
4	.7022	.1409
5	.4577	.0687
6	.4944	.2913

RAW NUMERIC DATA

S 1	2.5600	.5700	.5300	.6900	.4300	.4600
S 2	2.7000	.7200	.5400	.7200	.4100	.2500
S 3	2.1000	.5000	.5200	.6800	.4600	.3000
S 4	2.1100	.4700	.7100	.8400	.5600	.5300
S 5	1.7400	.3800	.4900	.5300	.4200	.4200
S 6	1.3800	.2500	.3800	.4100	.3300	.2200
S 7	2.0400	.4500	.6800	.8000	.4500	.4000
S 8	1.9200	.5700	.7000	.7800	.5500	1.2400
S 9	2.3700	.8700	.8200	.8700	.5100	.6300

VARIABLE	MINIMUM VALUE	MAXIMUM VALUE
1	1.3800	2.7000
2	.2500	.8700
3	.3800	.8200
4	.4100	.8700
5	.3300	.5600
6	.2200	1.2400

NUMERIC STANDARD SCORES

S 1	1.1914	.2259	-.5126	-.0867	-.4039	-.1182
S 2	1.5558	1.0972	-.4357	.1261	-.6948	-.8389
S 3	-.0057	-.1807	-.5895	-.1576	.0323	-.6673
S 4	.0202	-.3550	.8715	.9776	1.4865	.1220
S 5	-.9427	-.8778	-.8202	-1.2220	-.5493	-.2555
S 6	-1.8797	-1.6330	-1.6661	-2.0735	-1.8582	-.9419
S 7	-.1619	-.4711	.6408	.6937	-.1131	-.3241

S 8	-.4742	.2259	.7946	.5518	1.3411	2.5588
S 9	.6969	1.9686	1.7173	1.1904	.7594	.4652

NUMERIC VARIABLE CORRELATIONS

	1	2	3	4	5	6
1	1.000					
2	.800	1.000				
3	.372	.655	1.000			
4	.624	.711	.924	1.000		
5	.265	.442	.847	.831	1.000	
6	-.029	.282	.592	.459	.722	1.000

EIGENVALUES

3.93	1.37	.39	.21	.07	.01

PERCENTAGE VARIANCE

65.66	22.85	6.53	3.59	1.20	.17

CUMULATIVE VARIANCE

65.66	88.51	95.04	98.63	99.83	100.00

EIGENVECTORS - BY ROWS

VECTOR 1	.308	.401	.469	.483	.437	.313
VECTOR 2	-.632	-.405	.143	-.056	.333	.548
VECTOR 3	.163	.429	-.313	-.387	-.241	.694
VECTOR 4	.541	-.495	-.506	.086	.419	.143
VECTOR 5	-.214	.433	-.248	-.390	.674	-.311
VECTOR 6	.373	-.238	.584	-.673	.087	-.021

FIRST 6 FACTOR SCORES

S 1	-.038	-1.112	.502	.598	-.231	.117
S 2	.209	-2.190	.399	.117	-.006	-.062
S 3	-.622	-.354	-.303	.289	.361	-.179
S 4	1.434	.762	-1.075	.471	.208	.070
S 5	-1.939	.580	.154	-.032	.211	.158
S 6	-4.127	.593	.112	-.458	-.040	-.032
S 7	.246	.130	-.896	-.213	-.574	-.043
S 8	1.973	2.141	1.008	.207	-.106	-.076
S 9	2.865	-.551	.098	-.980	.178	.048

RAW BINARY DATA

SAMPLE 1 LENGTH	5	2	3	4	5	6
SAMPLE 2 LENGTH	5	1	3	4	5	6
SAMPLE 3 LENGTH	5	1	2	3	5	6
SAMPLE 4 LENGTH	5	1	2	3	4	6

SAMPLE 5	LENGTH	4	5	7	8	9
SAMPLE 6	LENGTH	4	5	6	8	9
SAMPLE 7	LENGTH	4	5	6	7	9
SAMPLE 8	LENGTH	4	5	6	7	8
SAMPLE 9	LENGTH	4	6	7	8	9

BINARY VARIABLE FREQUENCIES

3	3	4	3	7	8	4	4	4

PERCENTAGE OCCURRENCE FOR BINARY VARIABLES

6	88.9	5	77.8	9	44.4	8	44.4	3	44.4	7	44.4	4	33.3	1	33.3	2	33.3

COEFFICIENTS MATRIX

S 2	.444							
S 3	.444	.444						
S 4	.444	.444	.444					
S 5	.111	.111	.111	0.000				
S 6	.222	.222	.222	.111	.333			
S 7	.222	.222	.222	.111	.333	.333		
S 8	.222	.222	.222	.111	.333	.333	.333	
S 9	.111	.111	.111	.111	.333	.333	.333	.333

5 K-LINKAGE LISTS - (NEAREST NEIGHBOURS)

S 1	.444	2	.444	3	.444	4	.222	6	.222	7
S 2	.444	1	.444	3	.444	4	.222	6	.222	7
S 3	.444	1	.444	2	.444	4	.222	6	.222	7
S 4	.444	1	.444	2	.444	3	.111	6	.111	7
S 5	.333	6	.333	7	.333	8	.333	9	.111	1
S 6	.333	5	.333	7	.333	8	.333	9	.222	1
S 7	.333	5	.333	6	.333	8	.333	9	.222	1
S 8	.333	5	.333	6	.333	7	.333	9	.222	1
S 9	.333	5	.333	6	.333	7	.333	8	.111	1

TRIAL DATA - TESTS MIXED-MODE OPERATIONS AND PROVIDES PROGRAM DIAGNOSTICS
MODE NUCLEI GROUP 1 KL= 3 9 DENSE 2 CLUSTERS ENC RATIO 1.00 COEF .333

CLASSIFICATION ARRAY

1	1	1	1	2	2	2	2	2

CLUSTER 1 NUMBER OF CASES = 4

CASE NUMBERS
1 2 3 4

CLUSTER DIAGNOSIS OF MEANS, STANDARD DEVIATIONS AND F-RATIO

VAR	F-RATIO	T	MN-ORIG	STD-ORIG	VAR	F-RATIO	T	MN-ORIG	STD-ORIG
6	.1534	-.3755	.3850	.1141	4	.2049	.2148	.7325	.0638
2	.3147	.1969	.5650	.0966	3	.3622	-.1665	.5750	.0783
1	.4835	.6904	2.3675	.2671	5	.7032	.1050	.4650	.0577

BINARY VARIABLE FREQUENCIES

3 3 4 3 3 4 0 0 0

PERCENTAGE OCCURRENCE FOR BINARY VARIABLES

3 100.0 6 100.0 1 75.0 2 75.0 4 75.0 5 75.0 9 0.0 7 0.0 8 0.0

BINARY FREQUENCIES RATIO - (PERCENTAGE OCCURRENCE IN CLUSTER/PERCENTAGE OCCURRENCE OVERALL)

2 2.25 3 2.25 4 2.25 1 2.25 6 1.13 5 .96 9 0.00 7 0.00 8 0.00

CLUSTER 2 NUMBER OF CASES = 5

CASE NUMBERS
5 6 7 8 9

CLUSTER DIAGNOSIS OF MEANS, STANDARD DEVIATIONS AND F-RATIO

VAR	F-RATIO	T	MN-ORIG	STD-ORIG	VAR	F-RATIO	T	MN-ORIG	STD-ORIG
1	.7267	-.5523	1.8900	.3275	5	1.2216	-.0839	.4520	.0760
3	1.4703	.1333	.6140	.1577	6	1.4741	.3005	.5820	.3537
2	1.4924	-.1574	.5040	.2103	4	1.5697	-.1718	.6780	.1766

BINARY VARIABLE FREQUENCIES

0 0 0 0 4 4 4 4 4

PERCENTAGE OCCURRENCE FOR BINARY VARIABLES

5 80.0 8 80.0 7 80.0 9 80.0 6 80.0 3 0.0 4 0.0 2 0.0 1 0.0

BINARY FREQUENCIES RATIO - (PERCENTAGE OCCURRENCE IN CLUSTER/PERCENTAGE OCCURRENCE OVERALL)

8 1.80 7 1.80 9 1.80 5 1.03 6 .90 3 0.00 2 0.00 1 0.00 4 0.00

TRIAL DATA SCHEDULE - START OF SEQUENCE 2
EXECUTE CORREL

```
TRIAL DATA - TESTS MIXED-MODE OPERATIONS AND PROVIDES PROGRAM DIAGNOSTICS


NUMBER OF CASES =     9
NUMBER OF BINARY VARIABLES =     9
NUMBER OF NUMERIC VARIABLES =     6

CALCULATION OF COEFFICIENT MATRIX AND K-LINKAGE LISTS


SELECT FIRST   4 COMPONENT SCORES
SELECT COEFFICIENT NUMBER  1
COEFFICIENT IS DISSIMILARITY TYPE
FILE   5 K-LINKAGE LISTS
DISK FILE USES    65 RECORDS

COEFFICIENTS AND K-LINKAGE LISTS CALCULATED AND FILED
JOB ENDS
```

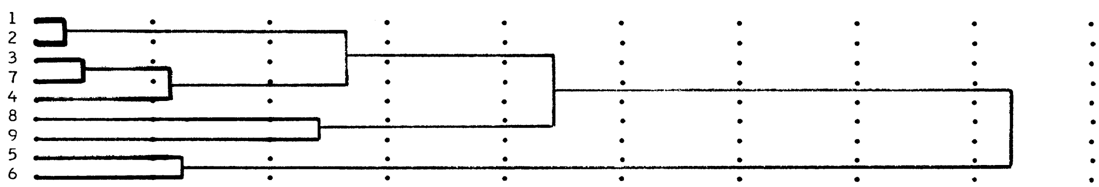
EXECUTE MODE (SEQUENCE 2)

```
TRIAL DATA - TESTS MIXED-MODE OPERATIONS AND PROVIDES PROGRAM DIAGNOSTICS


NUMBER OF CASES =     9
NUMBER OF BINARY VARIABLES =     9
NUMBER OF NUMERIC VARIABLES =     6


STANDARD SCORES FILED
BINARY DATA FILED
  6 COMPONENT SCORES FILED
COEFFICIENT  1 COMPUTED USING FIRST   4 COMPONENT SCORES


MODE ANALYSIS

DENSITY LEVEL =  3
MINIMUM NUMBER OF CLUSTERS CONTROL =  1
MINIMUM ENCLOSURE RATIO =   .80
FUSION OUTPUT OF CLUSTERS SIZE - MUST EXCEED  0 DENSE POINTS


NOW INTRODUCE DENSE POINT   3 AT COEFFICIENT    .730 AND FORM NEW CLUSTER NUCLEUS   3
NOW INTRODUCE DENSE POINT   1 AT COEFFICIENT   1.060 JOINS CLUSTER   3 AND NEW CLUSTER CODE IS   1
NOW INTRODUCE DENSE POINT   7 AT COEFFICIENT   1.060 JOINS CLUSTER   1 AND NEW CLUSTER CODE IS   1
NOW INTRODUCE DENSE POINT   5 AT COEFFICIENT   1.529 JOINS CLUSTER   1 AND NEW CLUSTER CODE IS   1
NOW INTRODUCE DENSE POINT   4 AT COEFFICIENT   1.652 JOINS CLUSTER   1 AND NEW CLUSTER CODE IS   1
NUMBER OF CLUSTERS =   1
ENCLOSURE RATIO =   .89
END CONDITIONS SATISFIED
JOB ENDS
```

EXECUTE HIERAR (SEQUENCE 2)

```
TRIAL DATA - TESTS MIXED-MODE OPERATIONS AND PROVIDES PROGRAM DIAGNOSTICS


NUMBER OF CASES =      9
NUMBER OF BINARY VARIABLES =     9
NUMBER OF NUMERIC VARIABLES =     6


STANDARD SCORES FILED
BINARY DATA FILED
  6 COMPONENT SCORES FILED
COEFFICIENT  1 COMPUTED USING FIRST   4 COMPONENT SCORES

LANCE/WILLIAMS FLEXIBLE FORM WITH BETA =  -.250
OUTPUT CLASSIFICATIONS FOR    2 TO   4 CLUSTERS


CYCLE    1 NOW FUSE POINTS   1    2 AT COEFFICIENT     .366 -   8 CLUSTERS AND NEW CLUSTER CODE IS    1
CYCLE    2 NOW FUSE POINTS   3    7 AT COEFFICIENT     .398 -   7 CLUSTERS AND NEW CLUSTER CODE IS    3
CYCLE    3 NOW FUSE POINTS   3    4 AT COEFFICIENT    1.216 -   6 CLUSTERS AND NEW CLUSTER CODE IS    3
CYCLE    4 NOW FUSE POINTS   5    6 AT COEFFICIENT    1.243 -   5 CLUSTERS AND NEW CLUSTER CODE IS    5
FLEXIBLE FORM GROUP    5 FUSE POINTS    8    9 AT COEF    2.573    4 CLUSTERS
   1    1    3    3    5    5    3    8    8
FLEXIBLE FORM GROUP    6 FUSE POINTS    1    3 AT COEF    2.606    3 CLUSTERS
   1    1    1    1    5    5    1    8    8
FLEXIBLE FORM GROUP    7 FUSE POINTS    1    8 AT COEF    4.527    2 CLUSTERS
   1    1    1    1    5    5    1    1    1
CYCLE    8 NOW FUSE POINTS   1    5 AT COEFFICIENT   10.969 -   1 CLUSTERS AND NEW CLUSTER CODE IS    1
```

DENDROGRAM TABLE



FUSION SUMMARY

1	2	.366
3	7	.398
3	4	1.216
5	6	1.243
8	9	2.573
1	3	2.606
1	8	4.527
1	5	10.969

JOB ENDS

EXECUTE RESULT (SEQUENCE 2)

TRIAL DATA - TESTS MIXED-MODE OPERATIONS AND PROVIDES PROGRAM DIAGNOSTICS

NUMBER OF CASES = 9
NUMBER OF BINARY VARIABLES = 9
NUMBER OF NUMERIC VARIABLES = 6

STANDARD SCORES FILED
BINARY DATA FILED
6 COMPONENT SCORES FILED
COEFFICIENTS MATRIX NOT FILED

TRIAL DATA - TESTS MIXED-MODE OPERATIONS AND PROVIDES PROGRAM DIAGNOSTICS
FLEXIBLE FORM GROUP 6 FUSE POINTS 1 3 AT COEF 2.606 3 CLUSTERS

CLASSIFICATION ARRAY

1 1 1 1 2 2 1 3 3

CLUSTER 1 NUMBER OF CASES = 5

CASE NUMBERS
1 2 3 4 7

CLUSTER DIAGNOSIS OF MEANS, STANDARD DEVIATIONS AND F-RATIO

VAR	F-RATIO	T	MN-ORIG	STD-ORIG	VAR	F-RATIO	T	MN-ORIG	STD-ORIG
6	.1232	-.3652	.3880	.1023	4	.2006	.3106	.7460	.0631
2	.3232	.0633	.5420	.0979	3	.3941	-.0050	.5960	.0816
1	.5030	.5200	2.3020	.2725	5	.5702	.0614	.4620	.0519

PERCENTAGE OCCURRENCE FOR BINARY VARIABLES

6 100.0 3 80.0 5 80.0 4 60.0 1 60.0 2 60.0 9 20.0 7 20.0 8 0.0

BINARY FREQUENCIES RATIO - (PERCENTAGE OCCURRENCE IN CLUSTER/PERCENTAGE OCCURRENCE OVERALL)

2 1.80 3 1.80 4 1.80 1 1.80 6 1.13 5 1.03 9 .45 7 .45 8 0.00

CLUSTER 2 NUMBER OF CASES = 2

CASE NUMBERS
5 6

CLUSTER DIAGNOSIS OF MEANS, STANDARD DEVIATIONS AND F-RATIO

VAR	F-RATIO	T	MN-ORIG	STD-ORIG	VAR	F-RATIO	T	MN-ORIG	STD-ORIG
6	.1178	-.5986	.3200	.1000	2	.1426	-1.2553	.3150	.0650
3	.1789	-1.2431	.4350	.0550	4	.1813	-1.6477	.4700	.0600
1	.2195	-1.4112	1.5600	.1800	5	.4283	-1.2037	.3750	.0450

PERCENTAGE OCCURRENCE FOR BINARY VARIABLES

5 100.0 8 100.0 9 100.0 7 50.0 6 50.0 3 0.0 4 0.0 2 0.0 1 0.0

BINARY FREQUENCIES RATIO - (PERCENTAGE OCCURRENCE IN CLUSTER/PERCENTAGE OCCURRENCE OVERALL)

8 2.25 9 2.25 5 1.29 7 1.13 6 .56 3 0.00 2 0.00 4 0.00 1 0.00

CLUSTER 3 NUMBER OF CASES = 2

CASE NUMBERS
8 9

CLUSTER DIAGNOSIS OF MEANS, STANDARD DEVIATIONS AND F-RATIO

VAR	F-RATIO	T	MN-ORIG	STD-ORIG	VAR	F-RATIO	T	MN-ORIG	STD-ORIG
5	.0846	1.0503	.5300	.0200	4	.1020	.8712	.8250	.0450
3	.2129	1.2560	.7600	.0600	1	.3429	.1113	2.1450	.2250
2	.7593	1.0973	.7200	.1500	6	1.0958	1.5121	.9350	.3050

PERCENTAGE OCCURRENCE FOR BINARY VARIABLES

7 100.0 6 100.0 8 100.0 5 50.0 9 50.C 2 0.0 1 0.0 4 0.0 3 0.0

BINARY FREQUENCIES RATIO - (PERCENTAGE OCCURRENCE IN CLUSTER/PERCENTAGE OCCURRENCE OVERALL)

8 2.25 7 2.25 6 1.13 9 1.13 5 .64 3 0.00 2 0.00 1 0.00 4 0.00

TRIAL DATA SCHEDULE - START OF SEQUENCE 3
EXECUTE CORREL

```
TRIAL DATA - TESTS MIXED-MODE OPERATIONS AND PROVIDES PROGRAM DIAGNOSTICS


NUMBER OF CASES =     9
NUMBER OF BINARY VARIABLES =     9
NUMBER OF NUMERIC VARIABLES =     6

CALCULATION OF COEFFICIENT MATRIX AND K-LINKAGE LISTS


MASK ALL BINARY DATA


USE NUMERIC DATA
THE FOLLOWING NUMERIC VARIABLES ARE MASKED FROM THE COEFFICIENT CALCULATION
  4   6
STANDARD SCORES SELECTED
SELECT COEFFICIENT NUMBER  3
COEFFICIENT IS SIMILARITY TYPE
FILE   5 K-LINKAGE LISTS
DISK FILE USES     66 RECORDS

COEFFICIENTS AND K-LINKAGE LISTS CALCULATED AND FILED
JOB ENDS
```

EXECUTE RESULT (SEQUENCE 3)

```
TRIAL DATA - TESTS MIXED-MODE OPERATIONS AND PROVIDES PROGRAM DIAGNOSTICS


NUMBER OF CASES =     9
NUMBER OF BINARY VARIABLES =     9
NUMBER OF NUMERIC VARIABLES =     6


STANDARD SCORES FILED
BINARY DATA FILED
  6 COMPONENT SCORES FILED
NUMERIC COEFFICIENT CALCULATED
THE FOLLOWING VARIABLES WERE USED TO COMPUTE COEFFICIENTS
  1   2   3   5
COEFFICIENT NUMBER  3 CALCULATED
  5 K-LINKAGE LISTS FILED


COEFFICIENTS MATRIX
```

S 2	.925							
S 3	.501	.326						
S 4	-.695	-.914	-.035					
S 5	-.689	-.825	.249	.869				
S 6	-.409	-.048	-.735	-.354	-.291			
S 7	-.513	-.588	-.780	.516	.077	.171		
S 8	-.930	-.956	-.168	.846	.906	.091	.343	
S 9	-.362	.005	-.702	-.405	-.323	.998	.119	.047

4 K-LINKAGE LISTS - (NEAREST NEIGHBOURS)

S	1	.925	2	.501	3	-.362	9	-.409	6
S	2	.925	1	.326	3	.005	9	-.048	6
S	3	.501	1	.326	2	.249	5	-.035	4
S	4	.869	5	.846	8	.516	7	-.035	3
S	5	.906	8	.869	4	.249	3	.077	7
S	6	.998	9	.171	7	.091	8	-.048	2
S	7	.516	4	.343	8	.171	6	.119	9
S	8	.906	5	.846	4	.343	7	.091	6
S	9	.998	6	.119	7	.047	8	.005	2

NO CLASSIFICATION DIAGNOSIS
JOB ENDS

KANSAS GEOLOGICAL SURVEY COMPUTER PROGRAM
THE UNIVERSITY OF KANSAS, LAWRENCE

PROGRAM ABSTRACT

Title (If subroutine state in title):

FORTRAN II programs for 8 methods of cluster analysis (CLUSTAN I)

Date: 4 January 1969

Author, organization: D. Wishart, Computing Laboratory, University of St. Andrews, St. Andrews, Fife, Scotland.

Direct inquiries to: Author or

Name: D.F. Merriam Address: Kansas Geological Survey, University of Kansas, Lawrence, Kansas

Purpose/description: Numerical classification of N samples according to numeric or binary variable values. This set of programs forms the basis of a cluster analysis program library - CLUSTAN I.

Mathematical method: Mode analysis, and the hierarchical fusion processes - nearest neighbor, farthest neighbor, group average, centroid, median, Ward's error sum method and Lance-Williams flexible

Restrictions, range: Maxima are - number of samples 250, number of numeric variables 200, number of binary variables 400 (for the FORTRAN IV version, the maximum number of samples is increased to 999).

Computer manufacturer: IBM Model: 1620 Model II

Programming language: FORTRAN II D

Memory required: 60 K Approximate running time: Depends on N

Special peripheral equipment required: Disk

Remarks (special compilers or operating systems, required word lengths, number of successful runs, other machine versions, additional information useful for operation or modification of program)

All disk input/output is controlled by one subroutine DISKIO which can be rewritten suitably for other disk systems or magnetic tape simulation. CLUSTAN I is available for IBM System 360/44 and has been tested on numerous occasions during 1968. Versions for KDF9, ICL 1909, ICL 4/50, CDC 6600 and other IBM System 360 models have been developed.

7. Computer applications in the earth sciences: Colloquium on classification procedures, edited by D.F. Merriam, 1966 . . . . $1.00
8. Prediction of the performance of a solution gas drive reservoir by Muskat's Equation, by Apolonio Baca, 1967 . . . . $1.00
9. FORTRAN IV program for mathematical simulation of marine sedimentation with IBM 7040 or 7094 computers, by J.W. Harbaugh and W.J. Wahlstedt, 1967 . . . . . $1.00
10. Three-dimensional response surface program in FORTRAN II for the IBM 1620 computer, by R.J. Sampson and J.C. Davis, 1967 . . . . $0.75
11. FORTRAN IV program for vector trend analyses of directional data, by W.T. Fox, 1967 . $1.00
12. Computer applications in the earth sciences: Colloquium on trend analysis, edited by D.F. Merriam and N.C. Cocke, 1967 . . . . $1.00
13. FORTRAN IV computer programs for Markov chain experiments in geology, by W.C. Krumbein, 1967. . . . . $1.00
14. FORTRAN IV programs to determine surface roughness in topography for the CDC 3400 computer, by R.D. Hobson, 1967 . . . . $1.00
15. FORTRAN II program for progressive linear fit of surfaces on a quadratic base using an IBM 1620 computer, by A.J. Cole, C. Jordan, and D.F. Merriam, 1967 . . . . . $1.00
16. FORTRAN IV program for the GE 625 to compute the power spectrum of geological surfaces, by J.E. Esler and F.W. Preston, 1967 . . . . $0.75
17. FORTRAN IV program for Q-mode cluster analysis of nonquantitative data using IBM 7090/7094 computers, by G.F. Bonham-Carter, 1967 . . . . $1.00
18. Computer applications in the earth sciences: Colloquium on time-series analysis, D.F. Merriam, editor, 1967 . . . . $1.00
19. FORTRAN II time-trend package for the IBM 1620 computer, by J.C. Davis and R.J. Sampson, 1967 . . . . $1.00
20. Computer programs for multivariate analysis in geology, D.F. Merriam, editor, 1968 . . $1.00
21. FORTRAN IV program for computation and display of principal components, by W.J. Wahlstedt and J.C. Davis, 1968 . . . . $1.00
22. Computer applications in the earth sciences: Colloquium on simulation, D.F. Merriam and N.C. Cocke, editors, 1968 . . . . $1.00
23. Computer programs for automatic contouring, by D.B. McIntyre, D.D. Pollard, and R. Smith, 1968 . . . . $1.50
24. Mathematical model and FORTRAN IV program for computer simulation of deltaic sedimentation, by G.F. Bonham-Carter and A.J. Sutherland, 1968 . . . . . . $1.00
25. FORTRAN IV CDC 6400 computer program for analysis of subsurface fold geometry, by E.H.T. Whitten, 1968 . . . . $1.00
26. FORTRAN IV computer program for simulation of transgression and regression with continuous-time Markov models, by W.C. Krumbein, 1968 . . . . $1.00
27. Stepwise regression and nonpolynomial models in trend analysis, by A.T. Miesch and J.J. Connor, 1968 . . . . $1.00
28. KWIKR8 a FORTRAN IV program for multiple regression and geologic trend analysis, by J.E. Esler, P.F. Smith, and J.C. Davis, 1968 . . . . $1.00
29. FORTRAN IV program for harmonic trend analysis using double Fourier series and regularly gridded data for the GE 625 computer, by J.W. Harbaugh and M.J. Sackin, 1968 . $1.00
30. Sampling a geological population (workshop on experiment in sampling), by J.C. Griffiths and C.W. Ondrick, 1968 . . . . $1.00
31. Multivariate procedures and FORTRAN IV program for evaluation and improvement of classifications, by Ferruh Demirmen, 1969 . . . . $1.00
32. FORTRAN IV programs for canonical correlation and canonical trend-surface analysis, by P.J. Lee, 1969 . . . . $1.00
33. FORTRAN IV program for construction of Pi diagrams with the Univac 1108 computer, by Jeffrey Warner, 1969 . . . . $1.00
34. FORTRAN IV program for nonlinear estimation, by R.B. McCammon, 1969 . . . $0.75
35. FORTRAN IV computer program for fitting observed count data to discrete distribution models of binomial, Poisson and negative binomial, by C.W. Ondrick and J.C. Griffiths, 1969 . . . . $0.75
36. GRAFPAC, graphic output subroutines for the GE 635 computer, by F.J. Rohlf, 1969 . $1.00
37. An iterative approach to the fitting of trend surfaces, by A.J. Cole $1.00
38. FORTRAN II programs for 8 methods of cluster analysis (CLUSTAN I), by David Wishart, 1969 . . . . $1.50